Actes de la conférence CAID 2024

(Conference on Artificial Intelligence for Defense)

Organisée par

Traitement d'image par IA

Méthodes auto-supervisées appliquées à l'analyse d'images satellites, enjeux et comparatif
Self-Supervised Learning applied to satellite image analysis, challenges and benchmark
Fabien Merceron, Vincent Partimbene, Gohar Dashyan, Sébastien Saubert, Guillaume Peltier, Kévin Sanchis and Pierre-Antoine Ganaye

Analyse Comparative des Approches de Désapprentissage Automatique pour la Protection des Données
Comparative Analysis of Machine Unlearning Approaches for Data Protection
Maria Salgado Herrera, Vincent Thouvenot, Alice Heliou, Adrien Becue

L'IA pour la cybersécurité

Outils d'Intelligence Artificielle pour la Détection d'Anomalies dans la Surveillance des Aéronefs Basse Altitude
Machine Learning Toolbox for Anomaly Detection in Low-Flying Aircraft Surveillance
Melvyn Pirolley, Raphael Couturier, Aymeric Cretin, Antoine Chevrot, Thomas Dubot

Raffinement du diffing binaire par arbitrage de la similarité et du matching
Improving binary diffing through similarity and matching intricacie
Roxane Cohen, Robin David, Riccardo Mori, Florian Yger, Fabrice Rossi

GABAIN pour les graphes de données et la sélection de variables. Application au clustering en cybersécurité
Graph representation and features selection using GABAIN. Application to clustering in Cybersecurity
Barbara Pilastre, Tristan Bitard-Feildel

IA pour agents autonomes

Transformer-based State Estimation for Multi-Target Tracking: Sensitivity Analysis against Varying Kinematic Parameters and Clutter Density
Transformer-Based State Estimation for Multi-Target Tracking: Sensitivity Analysis against Varying Kinematic Parameters and Clutter Density
Valentin Sonntag, Jean-Marc Le Caillec, Alain Peres, Stephane Devaud

Méthodologie pour un contrôle de drone explicable, cohérent et généralisable par apprentissage par renforcement
Methodology for explainable, consistent, and generalizable Reinforcement Learning drone control
Robinson Denève, Paul Chaudron, Axel Puig, Alexandre Kotenkoff, Mathias Formoso

Guidage de drone pour la triangulation a N-vues basé sur l'apprentissage par renforcement multi-agents
Multi-Agent Reinforcement Learning based Drone Guidance for N-View Triangulation
Timothee Gavin, Murat Bronz, Simon Lacroix

Couplage entre un apprentissage par renforcement profond et une machine à états : approche théorique
Coupling deep reinforcement learning and a state machine : a theoretical approach
Idriss Abdallah, Laurent Ciarletta, Patrick Hénaff, Jonathan Champagne, Matthieu Matthieu Bonavent

Étude des méthodes de distillation de connaissances pour la segmentation sémantique des images sous-marines
An investigation of knowledge distillation methods for underwater image semantic segmentation
Gabriel Gueganno, Ayoub Karine, Thibault Napoleon, Franck Florin,Ayman Alfalou

Traitement automatique des langues

POPCORN : IA d'extraction d'information à partir de sources textuelles pour le renseignement militaire
POPCORN: AI for extracting information from textual sources for military intelligence
Cédric Lopez, Sylvain Verdy, Guillaume Gadek, Maxime Prieur, Didier Schwab, Gilles Sérasset, Nakanyseth Vuth

Pourquoi se limiter à une recherche quand on peut l'étendre ? Amélioration de l'architecture RAG par des stratégies d'expansion de requêtes et d'agrégation de documents
Why just search when you can expand ? Enhancing RAG with Query Expansion Strategies and Document Aggregation
Louis Jourdain, Skander Hellal, Tony Marini

Benchmarking Self-supervised Learning Methods in Remote Sensing

Fabien Merceron, Vincent Partimbene, Gohar Dashyan, Sébastien Saubert,
Guillaume Peltier, Kévin Sanchis and Pierre-Antoine Ganaye
Safran.AI
Paris, France
Email: <name>.<surname>@safran-ai.safrangroup.com

***Abstract*—Self-supervised pretraining has proved to be a competitive tool to improve downstream task performance in the field of remote sensing. Attempts to create geospatial foundation models based on such pretraining techniques are increasing in numbers, and are a promising solution to exploit the vast amount of unannotated remote sensing imagery. Due to the widespread availability of various self-supervised techniques, either generic or specific to remote sensing, it becomes of importance for practitioners to find a way to identify the best performing pretraining method based on the downstream task being tackled. In this paper, we present a systematic benchmark of commonly used self-supervised pretraining methods and provide insights into the most appropriate approach depending on the chosen downstream tasks. Our results indicate that Masked Auto Encoders (MAE), a reconstruction-based method, seems to be the overall winner on most use-cases. We also show that ImageNet remains a powerful pretraining dataset and can produce competitive baselines, while building a tailored pretraining dataset using high-resolution satellite images can effectively improve the downstream performance compared to such baselines. Finally, we study the computational efficiency of pretraining methods and provide recommendations based on the available budget.**

***Index Terms*—deep learning, computer vision, remote sensing, optical imagery, foundation model, self-supervised learning, land use classification, object detection, semantic segmentation, benchmark**

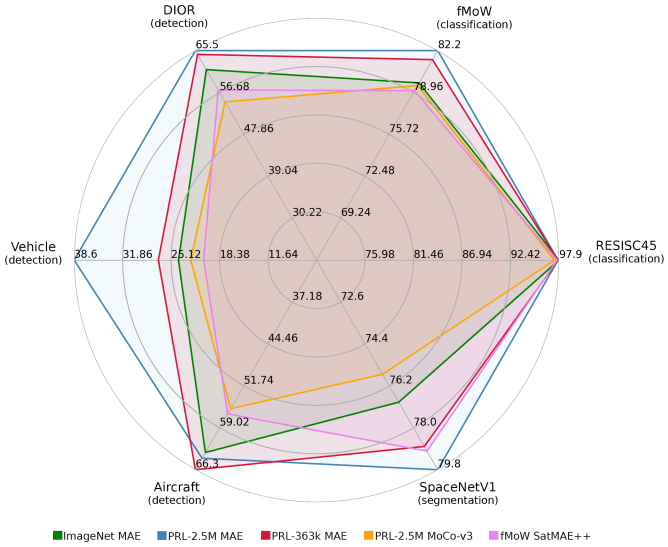


Fig. 1. Pretraining a ViT-Large with MAE on our internal datasets (PRL-2.5M and PRL-363k) improves performance on most downstream tasks compared to publicly available weights.

I. Introduction

The emergence of foundation models, a family of general-purpose models for solving a wide range of computer vision tasks, has overturned the traditional methodology of using a dedicated model for each task. This paradigm shift allows a common backbone to be used for any downstream task, especially classification, segmentation and detection. The training of a foundation model usually consists of two steps: pretraining and finetuning. The pretraining is done with the self-supervised learning (SSL) methodology, it makes it possible to train an encoder on a large quantity of unlabeled data using a pretext task, in order to learn how to extract meaningful visual features. Once complete, the finetuning step is applied to learn the downstream task, reusing the model's encoder while creating a new decoder.

If the pretraining is carried out correctly, the weights of the encoder can be reused in a variety of downstream tasks. Thus, selecting the most adapted SSL method is crucial as it highly impacts the generalization capabilities of the resulting model. Furthermore, the efficiency of the method is another important factor to be considered as pretraining is the main source of computational costs, especially when using large datasets.

Faced with all these technological opportunities, the question logically arises as to which is the optimal pretraining solution to use within the context of remote sensing.

In this article, we provide details on the efficiency of some SSL methods for vision tasks. Specifically, we study the effectiveness of a chosen joint embedding and reconstruction-based method on several downstream tasks commonly performed in remote sensing: land use classification, object detection and semantic segmentation. To do this, we study performance variation by exploring two major criteria: the size of the backbone and the pretraining dataset's size and composition.

Our main contributions are as follows:

- We explore the effectiveness of SSL methods in the context of remote sensing in several downstream tasks, including dense tasks such as object detection and semantic segmentation.
- We use several pretraining datasets of different size and composition, including very high-resolution commercial images and applications in both civilian and military

contexts. Furthermore, we study the benefits of such datasets compared to publicly available ones.
- We conduct our experiments on several backbone sizes to study the scaling capabilities of each method, including a large backbone that isn't commonly used in other works.
- We compare in-domain pretrainings with publicly available weights that serve as practical baselines for limited budgets and question the necessity to perform dedicated pretrainings to achieve high downstream performance.

As shown in Fig. 1, our internal datasets, combined with MAE, are able to consistently outperform other approaches. However, we note that publicly available weights are competitive alternatives when custom pretraining is not an option. We hope that our work proves useful to practitioners in facilitating the selection of an SSL method for remote sensing applications.

II. Related Work

A. Self-supervised Learning

SSL methods can be classified into different families based on how they interact with data. joint embedding methods learn to map similar data points close to each other in a latent space, while performing the opposite for dissimilar data points. MoCo [1] is arguably one of the reference methods of this family, building positive pairs using random data augmentation on a given image, and negative pairs using other images. Additionally, a contrastive loss that incentivizes similar embeddings for positive pairs and dissimilar embeddings for negative pairs is computed. To store negative pairs, MoCo uses a dictionary queue of fixed size (also called memory bank), which is decorrelated from the batch size. SimCLR [2] follows the same strategy, but uses the images contained in the current batch to form negative pairs, which alleviates the need of maintaining a queue, but requires a large batch size to achieve good performance. Later on, MoCo-v2 [3] and MoCo-v3 [4] improve the performance of the original method by using components from SimCLR and by replacing the original ResNet50 encoder with a ViT.

Another family of interest is reconstruction-based methods. Specifically, MAE [5] is arguably one of the most commonly used method of this type, which uses a ViT-based encoder-decoder architecture. The input image is first divided into non-overlapping patches of equal size. Then, most of the patches (typically 75%) are masked and the remaining patches are sent through the encoder as a sequence. Finally, the decoder aims at reconstructing the masked parts of the input as a reconstruction loss is used for learning. MAE significantly improves the efficiency of pretraining by only processing the visible patches.

B. Self-supervised Learning in Remote Sensing

The application of SSL methods in the field of remote sensing has received much attention in recent years [6], which can be explained by the large amount of available data and the high cost of the annotation process. These methods proved to be a competitive alternative to the traditional supervised learning methods [7]. Additionally, remote sensing imagery comes with its own set of characteristics that can be exploited to further improve performance, e.g., varying Ground Sample Distances (GSD), temporal dependencies between images, or the availability of additional bands in the context of multispectral imagery. Following this direction, adaptations of existing SSL methods have been proposed to take into account such specificities. SatMAE [8] leverages temporal and multispectral information during the reconstruction task by concatenating image timestamps to the positional encoding, and by using a curated selection of additional multispectral bands on top of the usual RGB ones for pretraining. Scale-MAE [9] incorporates a form of multi-scale decoding using Laplacian pyramids and proposes an update of the positional encoding to take into account the GSD of satellite images. SatMAE++ [10] extends Scale-MAE to work with multiple image scales by downsampling the input image twice and only sending the most downsized image through the encoder-decoder before upsampling it back to its original resolution. In this paper, we compare generic SSL methods with ones that take advantage of these specificities, and measure the expected gains associated with this additional complexity.

C. Self-supervised Learning Benchmarks in Remote Sensing

In the context of remote sensing, few works are dedicated to benchmarking SSL methods. Wang et al. [11] proposes an exhaustive review of SSL methods, including MoCo-v2, and provides a benchmark of several methods on three datasets. However, evaluation is only performed on a classification task using linear probing. Corley et al. [12] studies the impact of image sizing and normalization during pretraining on downstream task performance, and argues that ImageNet pretraining is a solid competitor. Nonetheless, this work only studies classification as a downstream task and MoCo-v2 as a pretraining method, and doesn't perform any finetuning.

On another note, some works that aim at building geospatial foundation models also provide extensive benchmarks of several SSL methods and downstream tasks. Similar to our work, Cha et al. [13] investigates the impact of increasing the number of model parameters in the context of SSL pretraining applied on object detection and semantic segmentation, primarily using ViTs. However, this work only explores a single pretraining method and a single pretraining dataset. More recently, Guo et al. [14] compares their proposed SkySense foundation model with a handful of SSL methods built for remote sensing, on various downstream tasks including classification, segmentation and detection. However, no attention is given to generic SSL methods nor ImageNet pretraining, as well as pretraining efficiency.

In contrast, our work focuses on the impact of different SSL methods on downstream task performance (classification, detection and segmentation) using two backbone sizes. It also investigates the impact of the pretraining dataset size and composition, while providing a pragmatic look on the necessity to perform a dedicated pretraining by comparing the

achievable performance with more frugal approaches based on publicly available weights, e.g., ImageNet.

III. Experimental Setup

In the following, we thoroughly compare the performance of joint embedding methods and reconstruction-based methods on various downstream tasks, in the context of remote sensing. Specifically, we choose to compare the performance of MoCo-v3 and MAE as two reference approaches, respectively, for these two families, and also as commonly used methods in remote sensing. The objective of these experiments is to get a better understanding of how each family of methods behaves with respect to the chosen downstream task, which in turn is useful to decide which pretraining method to favor when working on a downstream task. One might also wonder if remote sensing specific data and designs are necessary and significantly beneficial for downstream performance. For this reason, we also compare our results with SatMAE and SatMAE++ as the best performing remote sensing specific SSL methods, and with several in-domain pretraining datasets of various sizes. Finally, to measure the impact of backbone size, we systematically use ViT-Base and ViT-Large as backbones for all our experiments.

A. Pretraining

We study the impact of the pretraining data on downstream performance by first performing a self-supervised pretraining step on several remote sensing datasets. One of these is a public reference in the field, while others are internal datasets that we use to measure the expected gain when scaling the pretraining dataset size; our largest dataset being around 2.5 million images, which is approximately seven times larger than the public dataset we are using. In the following, we describe each of these datasets and discuss implementation details for the self-supervised methods we choose to focus on.

1) Datasets:

a) fMoW RGB: Functional Map of the World (fMoW) [15] is a large-scale dataset for functional land use classification. The dataset offers a wide range of ground resolutions from 0.5m to 35m per pixel. Since the original image size of fMoW varies, we pre-process the images identically to [15] and resize the input images to 224×224 pixels.

b) PRL-363k and PRL-2.5M: We build two high-resolution datasets by using mostly commercial data from the *Maxar/DigitalGlobe* satellites WorldView 2, 3, 4 and Pléiades Neo 3, 4. PRL-363k consists of the same number of images as fMoW RGB, i.e. 363,571 images. PRL-2.5M is a larger dataset consisting of 2,552,188 images. Both datasets consist of a curated collection of optical images with native GSDs ranging from 0.3m to 0.7m per pixel, all resampled to 0.3m. The geographic location of the images is not specified. The original image size is 512×512 but a 224×224 random crop of the input image is taken for pretraining.

2) Implementation Details:

a) MAE: The model configuration, optimizer and learning rate scheduler are the same as in [5]. We use 32 NVIDIA A100 to pretrain our model for 800 epochs with a base learning rate of $2.4e{-}4$ and an effective batch size of 16,384 for ViT-Base and 8192 for ViT-Large. We adopt the linear learning rate scaling rule [16]: $lr = base\_lr \times \frac{batchsize}{256}$. We apply data augmentation by performing a random flip with probability 0.5, and images are normalized using the standard ImageNet normalization. We use ImageNet MAE pretrained weights as initialization before pretraining following [17]. For all subsequent finetunings, we use the last epoch to initialize the backbone weights.

b) MoCo-v3: The model configuration, optimizer and learning rate scheduler are the same as in [4]. We use 32 NVIDIA A100 to pretrain our model for 300 epochs with a base learning rate of $2.4{-}4$ and an effective batch size of 4096 for both ViT-Base and ViT-Large. As for MAE, we adopt the linear learning rate scaling rule [16]. We apply the same data augmentation as described in [4]. We use ImageNet MoCo-v3 pretrained weights as initialization before pretraining following [17]. For all subsequent finetunings, we use the last epoch to initialize the backbone weights.

B. Downstream Tasks

Our pretrained models are finetuned on various downstream tasks, including classification (fMoW, RESISC45), segmentation (SpaceNetV1) and object detection (DIOR, PRL-Vehicle and PRL-Aircraft). In the following, we describe the content of each dataset as well as the implementation details for finetunings. Finally, we present the evaluation metrics used to measure downstream performance and the selected baselines.

1) Datasets:

a) fMoW RGB: We also use fMoW as a downstream classification task. We follow the official train and validation splits, which consist of 363,571 images and 50,041 images, respectively, distributed across 62 fine-grained and diverse categories.

b) RESISC45: Northwestern Polytechnical University (NWPU) developed RESISC45 [18], including 31,500 images distributed across 45 different scene categories from over 100 countries extracted from Google Earth. Each category contains 700 labeled images of size 256×256 pixels, resulting in a well-balanced distribution of scenes. The spatial resolutions of the images range from 0.2 to 30 meters per pixel. We use the dataset splits defined in [19] and keep the original input image size of 256×256.

c) SpaceNetV1: A dataset of 6,940 WorldView 2 satellite images at 0.5m per pixel [20]. We convert the original building footprint annotations (i.e. polygons) into segmentation masks and use the same dataset split as [8]. We keep the original image size of 400×400 pixels.

d) DIOR: A large-scale benchmark dataset for object detection in optical remote sensing images, which consists of 23,463 images of resolution varying from 0.5 to 30m per pixel and 192,518 object instances annotated with non-oriented bounding boxes. We follow the official train, validation and

TABLE I
STATISTICS FOR PRL-VEHICLE AND PRL-AIRCRAFT OBJECT DETECTION DATASETS

Observable	Split	Images	Pos. tiles	Neg. tiles	Num. Objects
Vehicle	train	204	49,932	9,963	369,851
	val	63	20,297	43,409	170,864
	test	88	4,712	46,269	32,550
Aircraft	train	3,239	19,110	3962	51,136
	val	202	2,067	5,348	5,386
	test	83	496	17,393	1,235

test splits, which are composed of 5,862 images, 5,863 images, and 11,738 images, respectively.

e) PRL-Vehicle and PRL-Aircraft: Our internal downstream datasets consist of Maxar WorldView-3 satellite images at 0.3m resolution, divided into tiles of 224×224 pixels and 512×512 pixels for vehicle and aircraft, respectively. The statistics of our datasets are reported in Table I. The vehicle dataset covers 8 military and civilian classes, while the aircraft dataset covers 6 military and civilian classes. Note that for PRL-Vehicle, we expect results to be on the lower end compared to other datasets as objects are very small.

2) Implementation Details:

a) Classification: We use a linear classification head on top of the ViT backbone. Augmentations, optimizer and learning rate scheduler are the same as in [5]. We use an effective batch size of 2048 for ViT-Base and 1024 for ViT-Large. Regarding the training we use a base learning rate of $2e{-}3$ and $4e{-}3$ for ViT-Base and ViT-Large, respectively. For the other configurations, we use a base learning rate of $1.5e{-}4$. We apply a layer-wise learning rate decay [21] of 0.75 for ViT-Large and 0.65 for ViT-Base following [22]. We use 4 NVIDIA V100 and finetune for 50 epochs on fMoW-RGB and 100 epochs on RESISC45, as in [9].

b) Segmentation: We use the UPerNet [23] head to perform semantic segmentation, as well as the feature pyramid implementation of ViTDet [24] to exploit multi-scale features. Augmentations, optimizer and learning rate scheduler are the same as in [8]. We use a base learning rate of $1.5e{-}4$ and an effective batch size of 64 for both ViT-Base and ViT-Large. We use 4 NVIDIA V100 and finetune for 100 epochs on SpaceNetV1, as in [8].

c) Detection: We use a RetinaNet [25] head to perform the detection task as well as the feature pyramid implementation of ViTDet [24]. Augmentations, optimizer and learning rate scheduler are the same as in [8]. The effective batch size is 64 for each dataset. For PRL-Aircraft and PRL-Vehicle, we use 4 NVIDIA V100 and finetune for 50 epochs. For DIOR, we use 4 NVIDIA A100 and finetune for 100 epochs. During finetuning we use LoRA [26] to get a better and faster convergence. Note that we do not use LoRA for other downstream tasks as it results in a performance drop.

C. Evaluation Metrics

We use the following metrics for our evaluations:

- **Classification** Top-1 accuracy. The evaluation epoch is selected based on the highest top-1 accuracy achieved on the validation set.
- **Segmentation** Mean Intersection over Union (IoU). The evaluation epoch is selected based on the highest mean IoU achieved on the validation set.
- **Detection** Mean average precision (mAP@0.5) of the PASCAL VOC object challenge [27]. The evaluation epoch is selected based on the highest mAP@0.5 achieved on the validation set.

D. Baselines

We compare our own pretrained backbones to several reference baselines that are identical for all downstream tasks. The first group of selected baselines consists of ImageNet pretrained weights. For that, we select supervised, MAE and MoCo-v3 weights for both ViT-Base and ViT-Large backbones. The second group consists of in-domain baselines, composed of SatMAE and SatMAE++ as they achieve state-of-the-art performance in the field of remote sensing. For compatibility reasons with the input data of our downstream tasks, we only use the RGB weights (not the multi-temporal or multi-spectral versions) of these methods. As weights are only available for ViT-Large, SatMAE and SatMAE++ will only be used as reference for this backbone.

IV. EXPERIMENTAL RESULTS

In this section, we discuss the results of our experiments with the aim of drawing insights about the behavior or MAE and MoCo-v3 when presented with various pretraining and downstream datasets. In some cases, the performance achieved by most methods are very close to each other. We argue that a variance study would have been beneficial to consolidate our conclusions, but we were not able to do so due to the high amount of additional experiments to be run.

A. Classification

Table II shows the Top-1 accuracies for RESISC45 and fMoW RGB. First, comparing the ImageNet baselines with our own built pretrained weights, we can see that the ImageNet baselines are strong. Specifically for RESISC45, the best weights for ViT-Base are ImageNet supervised and MoCo-v3 pretraining on PRL-2.5M. For ViT-Large, the best performance are obtained with ImageNet pretraining with MAE, closely followed by PRL-2.5M pretrained with MAE. Regarding fMoW RGB, our pretrainings outperform baseline approaches but the gap is mainly noticeable on PRL-2.5M pretrained with MAE, which ranks first for ViT-Base and ViT-Large. We argue that the competitive performance of ImageNet baselines can be explained by the fact that ImageNet is a dataset built for classification with centered, well-sized observables. Thus, features generated by supervised or SSL pretraining with ImageNet should be adequate by design for any classification task.

Regarding SatMAE and SatMAE++ with the ViT-Large backbone, we can see that none of them outperform the ImageNet MAE baseline, but, it should be noted that they support

TABLE II
TOP-1 ACCURACY ON THE RESISC45 AND FMOW CLASSIFICATION DATASETS

Backbone	Dataset	Method	RESISC45	fMoW
ViT-Base	-	Random init.	76.7	66.0
ViT-Base	IN	MoCo-v3	97.4	78.2
ViT-Base	IN	MAE	97.5	78.6
ViT-Base	IN	Sup.	**97.6**	79.0
ViT-Base	fMoW	MoCo-v3	97.5	79.2
ViT-Base	fMoW	MAE	97.5	79.5
ViT-Base	PRL-363k	MoCo-v3	97.5	78.9
ViT-Base	PRL-363k	MAE	97.5	**80.1**
ViT-Base	PRL-2.5M	MoCo-v3	**97.6**	79.8
ViT-Base	PRL-2.5M	MAE	97.4	**80.1**
ViT-Large	-	Random init.	70.5	68.6
ViT-Large	IN	MoCo-v3	97.2	78.0
ViT-Large	IN	MAE	**97.9**	79.7
ViT-Large	IN	Sup.	97.4	79.0
ViT-Large	fMoW	SatMAE	97.0	76.1
ViT-Large	fMoW	SatMAE++	97.7	79.1
ViT-Large	fMoW	MoCo-v3	97.7	79.4
ViT-Large	fMoW	MAE	97.5	80.3
ViT-Large	PRL-363k	MoCo-v3	97.1	78.9
ViT-Large	PRL-363k	MAE	97.8	81.5
ViT-Large	PRL-2.5M	MoCo-v3	97.4	79.5
ViT-Large	PRL-2.5M	MAE	97.8	**82.2**

TABLE III
MAP@0.5 ON THE DIOR, PRL-VEHICLE AND PRL-AIRCRAFT DETECTION DATASETS

Backbone	Dataset	Method	DIOR	Vehicle	Aircraft
ViT-Base	-	Random init.	29.2	12.6	29.9
ViT-Base	IN	MoCo-v3	51.8	19.9	59.5
ViT-Base	IN	MAE	55.7	22.2	60.6
ViT-Base	IN	Sup.	56.2	20.6	**65.4**
ViT-Base	fMoW	MoCo-v3	52.2	20.2	57.0
ViT-Base	fMoW	MAE	55.2	23.4	55.6
ViT-Base	PRL-363k	MoCo-v3	51.7	21.9	59.6
ViT-Base	PRL-363k	MAE	55.6	22.9	58.3
ViT-Base	PRL-2.5M	MoCo-v3	53.1	24.3	57.2
ViT-Base	PRL-2.5M	MAE	**60.5**	**25.4**	64.6
ViT-Large	-	Random init.	21.4	4.9	30.0
ViT-Large	IN	MoCo-v3	55.2	19.8	65.3
ViT-Large	IN	MAE	61.5	24.1	63.3
ViT-Large	IN	Sup.	57.4	22.5	64.7
ViT-Large	fMoW	SatMAE	53.1	21.0	54.2
ViT-Large	fMoW	SatMAE++	57.2	20.6	56.6
ViT-Large	fMoW	MoCo-v3	56.6	22.3	58.1
ViT-Large	fMoW	MAE	59.7	26.5	62.5
ViT-Large	PRL-363k	MoCo-v3	54.7	21.0	60.1
ViT-Large	PRL-363k	MAE	64.7	26.9	**66.3**
ViT-Large	PRL-2.5M	MoCo-v3	54.7	22.4	55.7
ViT-Large	PRL-2.5M	MAE	**65.5**	**38.6**	64.3

multi-spectral / multi-temporal inputs that our experimental setup does not.

Looking at the benefit of our internal datasets against the fMoW RGB dataset, we can see a positive performance impact. Indeed, PRL-2.5M always ranks higher than fMoW RGB, especially with the MAE paradigm, which highlights the usefulness of scaling up the amount of pretraining data. PRL-363k shows weaker results than PRL-2.5M, but still manages to improve performance over fMoW RGB, especially with the MAE paradigm, which could be explained by the higher resolution of PRL-363k images compared to fMoW.

Finally, by focusing our attention on pretraining methods, we can observe that the MAE paradigm shows better performance than MoCo-v3. Indeed, except for the RESISC45 dataset with the ViT-Base backbone, MAE consistently outperforms MoCo-v3. On top of that, we can see that the use of PRL-363k or PRL-2.5M over fMoW has a negative performance impact on MoCo-v3, which is not the case with MAE.

B. Detection

Table III shows the mAP@0.5 for DIOR and both PRL-Aircraft and PRL-Vehicle. The ImageNet baselines on all datasets are competitive especially on the PRL-Aircraft dataset where it ranks first for ViT-Base and among the top for ViT-Large. For DIOR and PRL-Vehicle, the best performance is achieved by pretraining with MAE on the biggest dataset (PRL-2.5M). We argue that the particularly high performance of ImageNet supervised pretraining on PRL-Aircraft might be due to the fact that, compared to PRL-Vehicle and DIOR, objects are closer to what can be found in ImageNet dataset, i.e., covering a large portion of the image.

As for in-domain baselines, both SatMAE and SatMAE++ are among the worst performing methods except for DIOR where SatMAE++ ranks in the middle near MoCo-v3 methods.

All other things being equal, using internal datasets such as PRL-363k and PRL-2.5M instead of fMoW seems to be more beneficial. However, the MAE pretraining on fMoW always ranks higher that MoCo-v3 on PRL-363k, and sometimes PRL-2.5M. In light of these results, we argue that the choice of the pretraining method is essential in order to fully exploit the benefits of large-scale pretraining datasets.

Finally, when comparing pretraining methods, we can observe that MAE always outperforms MoCo-v3 at comparable settings. It has also better scaling properties, as going from PRL-363k to PRL-2.5M, results in the highest gain in metrics for MAE, whereas MoCo-v3 only achieves small or nonexistent gains. This is also the case when going from ViT-Base to ViT-Large. From that, we conclude that scaling the backbone seems to have a greater impact than scaling the dataset.

C. Segmentation

Table IV shows the mean IoU for SpaceNetV1. First, comparing the ImageNet baselines with our own pretrained weights, we can see that the ImageNet baselines are competitive, especially the ImageNet supervised one. Indeed, this baseline is only outperformed by the MAE pretraining on PRL-2.5M for the ViT-Base backbone and the MAE pretraining for both PRL-363k and PRL-2.5M for the ViT-Large backbone.

Regarding in-domain baselines, SatMAE is the worst performing method, on the opposite of SatMAE++, which manages to rank second among all models for the ViT-Large backbone, making it a solid baseline.

TABLE IV
MEAN IOU ON THE SPACENETV1 SEGMENTATION DATASET

Backbone	Dataset	Method	SpaceNetV1
ViT-Base	-	Random init.	70.8
ViT-Base	IN	MoCo-v3	74.5
ViT-Base	IN	MAE	76.9
ViT-Base	IN	Sup.	78.4
ViT-Base	fMoW	MoCo-v3	77.0
ViT-Base	fMoW	MAE	76.0
ViT-Base	PRL-363k	MoCo-v3	77.4
ViT-Base	PRL-363k	MAE	75.9
ViT-Base	PRL-2.5M	MoCo-v3	77.7
ViT-Base	PRL-2.5M	MAE	**79.8**
ViT-Large	-	Random init.	72.2
ViT-Large	IN	MoCo-v3	74.4
ViT-Large	IN	MAE	76.9
ViT-Large	IN	Sup.	77.9
ViT-Large	fMoW	SatMAE	75.3
ViT-Large	fMoW	SatMAE++	79.0
ViT-Large	fMoW	MoCo-v3	77.0
ViT-Large	fMoW	MAE	77.1
ViT-Large	PRL-363k	MoCo-v3	76.2
ViT-Large	PRL-363k	MAE	78.8
ViT-Large	PRL-2.5M	MoCo-v3	75.7
ViT-Large	PRL-2.5M	MAE	**79.8**

When comparing our internal datasets with fMoW, we see that pretraining with PRL-363k or PRL-2.5M shows benefits over fMoW but mainly with MAE as pretraining with MoCo-v3 often results in a performance loss.

At last, when looking at pretraining methods, we observe that MAE performs better overall than MoCo-v3. Indeed, except for the ViT-Base backbone pretrained on PRL-363k, the performance of MAE is higher than MoCo-v3 in any other case.

V. DISCUSSION

In this section, we provide general insights with the aim of facilitating the choice of a pretraining methods. First, we question the benefits of a custom in-domain SSL pretraining over existing ImageNet pretrained weights. Table V shows the difference between several aggregations from our different downstream results. If not mentioned otherwise and applicable, all aggregations consider both backbones (ViT-Base and ViT-Large), both paradigms (MAE and MoCo-v3) and exclude results from SatMAE and SatMAE++, as they tend to achieve sub-par performance in our RGB setting.

The first column shows the average performance gap of using in-domain SSL pretraining over ImageNet baselines. It shows that in-domain SSL pretrained weights provide benefits in term of downstream performance for segmentation, but with mitigated results for classification and detection. We argue that the supervised pretraining on ImageNet, a classification task, can yield better representations for a classification downstream task. Considering these results, we recommend using already available ImageNet weights when working on classification, and potentially go for a dedicated pretraining with the MAE paradigm to try to push the performance further.

The second column shows the potential benefit of using PRL-363k over fMoW, knowing that they have the exact same number of images but that PRL-363k is of higher native resolution. We compare the difference of the average metrics for PRL-363k against fMoW, and show that there are no strong positive benefits for the classification task and close to no benefits for the segmentation and detection tasks, except for the aircraft outlier. We argue that the domain gap between PRL-363k, which contains high resolution images, and downstream task datasets, which contain a mix of resolutions closer to the ones in fMoW, can be responsible for this absence of significant benefits. Finally, we hypothesize that pretraining on fMoW should yield better downstream performance on fMoW as the model has already seen the data during the pretraining step.

The third column shows the average performance gap of pretraining on PRL-2.5M over fMoW, e.g. with a bigger and high resolution dataset. Results show that there are clear benefits using the PRL-2.5M, confirming that pretraining on larger datasets can be beneficial. Additional experiments could be dedicated to studying the impact of pretraining resolution on downstream performance by lowering the resolution of PRL-2.5M . Based on the results of this column and the previous one, we would recommend building an internal pretraining dataset only if its expected size is higher that publicly available datasets. Building an aggregate of various public datasets can also be an interesting alternative, as in [28].

The last column shows the performance gap from using MAE over MoCo-v3. Overall, MAE provides better performance than MoCo-v3, and the gap between MAE and MoCo-v3 grows larger using a ViT-Large backbone when using the same pretraining dataset. Thus, we argue that MAE should be favored as it scales positively on all downstream tasks and backbone sizes.

On another note, pretraining requires a significant amount of computing power to converge within few hours or days. During our experiments we have observed a large difference in terms of speed and memory usage, thus we choose to report the efficiency as a major criteria of evaluation, to compare the MoCo-v3 and MAE. Figure 2 shows that the time required to pretrain using one image sample is much higher with MoCo-v3. In addition, we note that the memory consumption significantly increases with MoCo-v3, forcing us to reduce the batch size (4 times smaller than MAE for ViT-Base, as discussed in Experimental Setup) and thus increasing the number of iterations for each epoch. We believe that MAE is the best compromise in terms of efficiency, reaching the best performance with a given compute budget.

In light of these results, we can draw the following conclusions:

- **Dedicated pretraining is beneficial for most downstream tasks** As shown in the results, using weights from a dedicated in-domain pretraining outperforms ImageNet baselines except for two downstream tasks (RESISC45 and PRL-Aircraft) when pretraining is performed with ViT-Base.

TABLE V
SUMMARY OF DOWNSTREAM TASK PERFORMANCE FOR ALL METHODS

Downstream task	Dataset	(fMoW, PRL-363k, PRL2.5M) vs baselines	PRL-363k vs fMoW*	PRL-2.5M vs fMoW*	MAE vs MoCo-v3
Segmentation	SpaceNetV1	+0.9	+0.3	+1.5	+1.4
Classification	RESISC45	0	-0.1	0	+0.19
	fMoW	+1.2	+0.25	+0.8	+1.3
Detection	DIOR	+0.2	+0.8	+1.1	+6.8
	Vehicle	+2.9	+0.1	+3.9	+5.1
	Aircraft	-4.1	+3.9	+1.4	+2.7

*Comparing average metric considering ViT-Base and ViT-Large backbones. SatMAE and SatMAE++ always excluded.

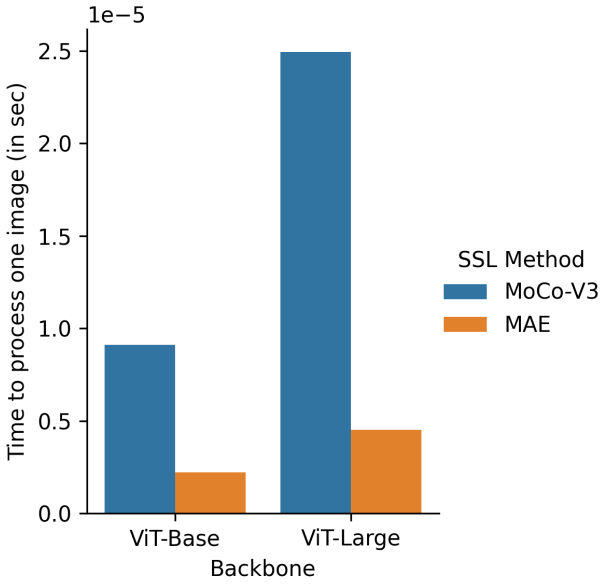


Fig. 2. Time required to process one image during the pretraining, depending on the SSL Method and backbone type. The measured time includes forward and backward passes.

- **MAE outperforms MoCo-v3 overall** MAE is able to consistently outperform MoCo-v3 while being computationally more efficient. Furthermore, it is able to scale better when using a bigger backbone or pretraining dataset.
- **Publicly available weights are solid competitors** Without any additional pretraining, using publicly available weights is enough to achieve competitive performance. When working with a limited budget, it seems fine to use ImageNet, SatMAE, or SatMAE++ pretrained weights. However, performing a dedicated pretraining remains interesting when pushing for the best performance on dense downstream tasks.

VI. CONCLUSION

In this paper, we study and compare the downstream performance of commonly used pretraining methods in the context of remote sensing imagery. We select a reference method for two families of self-supervised approaches and investigate their benefits on different downstream tasks. We pretrain MoCo-v3 and MAE on several datasets of different scale and composition and show that increasing the amount of pretraining data significantly improves the performance in downstream tasks. Experimental results also show that MAE is a strong competitor that achieves the best overall performance on the chosen downstream tasks while exhibiting better backbone scaling capabilities, and that using publicly available ImageNet weights is usually sufficient to achieve satisfactory performance. Finally, we show that pretraining methods that are specific to remote sensing are competitive alternatives, but do not manage to outperform more generic approaches by a large margin. Future directions of this work include expanding the number of benchmarked SSL methods, as well as studying the impact of GSD in the pretraining data. Furthermore, the study of other backbones may prove useful, as other architectures may yield different results. At last, methods to build datasets that best benefit the pretraining phase also constitute a promising direction of research.

ACKNOWLEDGEMENTS

This project was provided with computer and storage resources by GENCI at IDRIS thanks to the grant 2023-A0151014554 on the supercomputer Jean Zay's A100 and V100 partitions.

REFERENCES

[1] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," *CoRR*, vol. abs/1911.05722, 2019. [Online]. Available: http://arxiv.org/abs/1911.05722

[2] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learmning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 1597–1607. [Online]. Available: https://proceedings.mlr.press/v119/chen20j.html

[3] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020.

[4] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9620–9629, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:233024948

[5] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[6] Y. Wang, C. M. Albrecht, N. A. A. Braham, L. Mou, and X. X. Zhu, "Self-supervised learning in remote sensing: A review," *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 4, pp. 213–247, 2022.

[7] K. Ayush, B. Uzkent, C. Meng, K. Tanmay, M. Burke, D. Lobell, and S. Ermon, "Geography-aware self-supervised learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 181–10 190.

[8] Y. Cong, S. Khanna, C. Meng, P. Liu, E. Rozi, Y. He, M. Burke, D. Lobell, and S. Ermon, "SatMAE: Pre-training transformers for temporal and multi-spectral satellite imagery," *Advances in Neural Information Processing Systems*, vol. 35, pp. 197–211, 2022.

[9] C. J. Reed, R. Gupta, S. Li, S. Brockman, C. Funk, B. Clipp, K. Keutzer, S. Candido, M. Uyttendaele, and T. Darrell, "Scale-MAE: A scale-aware masked autoencoder for multiscale geospatial representation learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4088–4099.

[10] M. Noman, M. Naseer, H. Cholakkal, R. M. Anwer, S. Khan, and F. S. Khan, "Rethinking transformers pre-training for multi-spectral satellite imagery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 27 811–27 819.

[11] Y. Wang, C. M. Albrecht, N. A. A. Braham, L. Mou, and X. X. Zhu, "Self-supervised learning in remote sensing: A review," 2022. [Online]. Available: https://arxiv.org/abs/2206.13188

[12] I. Corley, C. Robinson, R. Dodhia, J. M. L. Ferres, and P. Najafirad, "Revisiting pre-trained remote sensing model benchmarks: Resizing and normalization matters," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2024, pp. 3162–3172.

[13] K. Cha, J. Seo, and T. Lee, "A billion-scale foundation model for remote sensing images," 2024. [Online]. Available: https://arxiv.org/abs/2304.05215

[14] X. Guo, J. Lao, B. Dang, Y. Zhang, L. Yu, L. Ru, L. Zhong, Z. Huang, K. Wu, D. Hu, H. He, J. Wang, J. Chen, M. Yang, Y. Zhang, and Y. Li, "Skysense: A multi-modal remote sensing foundation model towards universal interpretation for earth observation imagery," 2024. [Online]. Available: https://arxiv.org/abs/2312.10115

[15] G. Christie, N. Fendley, J. Wilson, and R. Mukherjee, "Functional map of the world," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6172–6180.

[16] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch sgd: Training imagenet in 1 hour," *arXiv preprint arXiv:1706.02677*, 2017.

[17] C. J. Reed, X. Yue, A. Nrusimha, S. Ebrahimi, V. Vijaykumar, R. Mao, B. Li, S. Zhang, D. Guillory, S. Metzger, K. Keutzer, and T. Darrell, "Self-supervised pretraining improves self-supervised pretraining," *CoRR*, vol. abs/2103.12718, 2021. [Online]. Available: https://arxiv.org/abs/2103.12718

[18] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1865–1883, 2017.

[19] M. Neumann, A. S. Pinto, X. Zhai, and N. Houlsby, "In-domain representation learning for remote sensing," *arXiv preprint arXiv:1911.06721*, 2019.

[20] A. Van Etten, D. Lindenbaum, and T. M. Bacastow, "SpaceNet: A remote sensing dataset and challenge series," *arXiv preprint arXiv:1807.01232*, 2018.

[21] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," *arXiv preprint arXiv:2003.10555*, 2020.

[22] H. Bao, L. Dong, S. Piao, and F. Wei, "BEiT: BERT pre-training of image transformers," *arXiv preprint arXiv:2106.08254*, 2021.

[23] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[24] Y. Li, H. Mao, R. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*. Berlin, Heidelberg: Springer-Verlag, 2022, p. 280–296. [Online]. Available: https://doi.org/10.1007/978-3-031-20077-9_17

[25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[26] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[27] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, pp. 98–136, 2015.

[28] M. Mendieta, B. Han, X. Shi, Y. Zhu, and C. Chen, "Towards geospatial foundation models via continual pretraining," 2023. [Online]. Available: https://arxiv.org/abs/2302.04476

Comparative Analysis of Machine Unlearning Approaches for Data Protection

Maria Salgado Herrera
ThereSIS
Thales SIX GTS France
Palaiseau, France
msalgadoh@unal.edu.co

Vincent Thouvenot
ThereSIS
Thales SIX GTS France
Palaiseau, France
vincent.thouvenot@thalesgroup.com

Alice Héliou
ThereSIS
Thales SIX GTS France
Palaiseau, France
alice.heliou@thalesgroup.com

Adrien Becue
ThereSIS
Thales SIX GTS France
Gennevilliers, France
adrien.becue@thalesgroup.com

***Abstract*—The increasing sensitivity and widespread adoption of machine learning models in various applications have led to a growing need for "machine unlearning" - the process of removing the influence of specific data used to train a model without retraining from scratch the model. As machine learning models handle personal and sensitive data, it is crucial to develop robust and adaptive unlearning approaches that can defend against attacks. This paper provides a comprehensive overview of the latest tools and approaches used in machine unlearning, focusing on short execution time and good performance. We present a standardized comparison of different automatic unlearning methods, highlighting their differences, advantages, and limitations. Our work aims to contribute to the development of more efficient and effective machine unlearning techniques, addressing the challenges of user privacy, bias removal, and confusion resolution.**

***Index Terms*—Deep neural networks, Image classification, Machine Learning, Security, Privacy, Machine Unlearning**

I. Introduction

Machine learning models emerged with the objective of training a dataset to learn parameters and create relationships in the data. This process involves feeding the model with a substantial amount of data, allowing it to recognize patterns and correlations. Once the algorithm is created, it can be used to make predictions on some input data. The goal of machine learning (ML) is to create accurate models that can generalize well to new data [1].

In recent years, machine learning models have evolved significantly, becoming increasingly sensitive and specific in their capabilities. These models are now commonly used in a wide variety of applications that involve critical information about users, tasks, and processes. For example, in sectors such as biometrics [2] and security in Cyber Physical systems, such as electric power grid [3], machine learning models handle personal and/or sensitive data that require careful and ethical management. With the increase in the sensitivity of these models and the amount of personal data they process, a new need has arisen: machine unlearning. Machine unlearning refers to the process of selectively removing specific data points from a machine learning model, to effectively "forget" information.

Moreover, the development of automatic unlearning models is constantly growing and a common challenge is their defense against attacks, so many approaches have been proposed to develop models that are as adaptive and robust as possible. This has fueled a debate on the feasibility of approximate unlearning, which would enable models to forget learned patterns without requiring complete retraining from scratch, thereby significantly reducing the computational costs. The adoption of approximate machine unlearning techniques over exact machine unlearning is justified by their superior efficiency and scalability, as well as their adequacy for numerous applications. Although exact machine unlearning, which entails manual removal of the information to be forgotten and retraining from scratch, achievies ideal results, its implementation is often hindered by high computational costs and time requirements. In contrast, approximate machine unlearning offers a more practical and effective solution for mitigating the influence of unwanted data in a model, particularly in scenarios where complete removal of the unwanted data's influence is not feasible or necessary [4].

In this paper, we will focus on explaining the development of different automatic approximate unlearning approaches with the aim of revealing the differences and possible advantages depending on the tools used to perform the forgetting of the information used by the model to be forgotten. Our contribution consists in showing in a standardized way an overview of the latest tools used in information forgetting in machine learning models, with a focus on short execution time while maintaining a good performance. Additionally, we seek to evaluate the robustness of the models through attacks that typically aim to exploit vulnerabilities, with the goal of maintaining a good performance. Our paper is structured in 7 sections. Following this introduction, Section 2 provides insights on business needs and motivations, Section 3 introduce the theoretical background, Section 4 describes the methodology, Section 5 establishes the experimental datasettings, Section 6 provides study results and Section 7 concludes.

II. Business need/Motivations

Machine unlearning refers to the process of selectively removing specific data points from a machine learning model, effectively "forgetting" information. This ability is crucial in the defense and security sector, where sensitive data may need to be purged from models for various reasons such as:

the revocation of classified information [1], compliance with data protection regulation [5], mitigation of data poisoning attacks [6], controlled information disclosure [7], updating operational intelligence [8], demilitarization of dual use technology [9] or compliance with export control regulation [10]. While the concept is promising, there are several gaps and challenges in the current state of the art: limited efficiency and scallability [11], insufficient accuracy and integrity [11], confidentiality and security concerns [1], legal and compliance issues [12], limited versatility and generalization [13], lack of theoretical foundations [1]. Security concerns regarding unlearned models, particularly the potential retrieval of "forgotten" data classes, revolve around the risk that sensitive or protected information may still be inferred or reconstructed, even after attempting to remove it [11]. For these reasons, a comparative assessment of diverse unlearning techniques aiming to evaluate their robustness to membership inference attacks is of paramount importance for applications in the defense and security sector. Moreover, to justify the use of unlearning techniques, rather than retraining techniques, the run-time efficiency of these different approaches needs to be assessed.

III. Background

Machine Unlearning, introduced in 2015 by [14], is an emerging topic of Machine Learning. The unlearning approaches can be approximate or exact. For the latter the resources and time required can be dissuasive. In [7] and [15] , the authors propose a framework for unlearning that relies on data shading and slicing to reduce the computational overhead of unlearning. Such approaches ask to create several sub samples to limit the impact of forgetting an observation. However, it can still be costly if the amount of information to forget is high.

Most of the researches in the field tend to design a quick and efficient way to perform unlearning that approximate as well as possible a retraining from scratch. [16] propose the SCRUB method, that see the unlearning problem as a teacher-student problem. Moreover, they illustrate several tasks of unlearning, in particular removing bias, resolving confusion and user privacy. [17] propose a method called Amnesias Unlearning, where during a training, the model owner keeps a list of which examples appeared in which batches as well as the parameter updates from each batch. To remove a sensitive information, the model owner undoes the parameter updates coming from the batch. [18] propose a linear filtration for logit-based classifier.

Recent research on machine unlearning, such as [19] has identified several key approaches for unlearning requests, including data removal, feature removal, class removal, task removal, and stream removal. Among the advanced techniques, the data remover and class remover are particularly emphasized. The data remover technique unlearns specific data points from the training dataset, crucial for privacy compliance and error correction. The class remover technique eliminates entire classes of data, useful for maintaining relevance and ethical integrity when certain categories are no longer needed or pose ethical concerns [1]. Our paper focuses on data removal.

Verification of unlearning performance is a complex topic (see [20]). As machine learning models have advanced in complexity, the attacks targeting these models have also evolved, aiming to undermine both their security and privacy. Two of the most notable types of attacks are Model Inversion Attacks and Membership Inference Attacks [1]. Model Inversion Attacks focus on reconstructing sensitive features of the training data by exploiting correlations with the model's output, ultimately aiming to recreate the input data, particularly the most private features [21]. Membership Inference Attacks [22] are a method used by adversaries to determine whether a specific example was part of a model's training dataset. These attacks are particularly useful in verifying if a group of data has been successfully forgotten by a model. If unlearning is effective, the attacker should not be able to guess whether a data was forgotten or never used (part of the test dataset). The attack (e.g. [23]) involves identifying whether a particular datum was included in the training dataset, often assessed using "average case" scenarios, where the likelihood that a data point belongs to the target dataset is compared against the test dataset. Key metrics such as true positive rate and false positive rate are crucial in evaluating the effectiveness of these attacks [24].

IV. Methodology

A. Unlearning task and type of model analyzed

1) Unlearning task: Based on the challenge proposed by [20], we consider the challenge illustrated by Figure 1. First, a deep neural classifier, called original model, is trained on a training dataset. From this original model, we train a new model, called unlearned model, whose objective consist in forgetting part of the information while preserving the performance on the other part of the data. During this second stage, the training dataset is divided in two datasets:

- The forget dataset, for which the aim is for the unlearned model to output predictions that are indistinguishable from the predictions made for the test dataset.
- The retain dataset, for which the aim is for the unlearned model to obtained on it similar performance as the original model.

In Figure 1, the test dataset correspond to data never seen during the training of the original and the unlearned models.

Then, information are extracted from the outputs of the unlearned model to evaluate if the outputs for data from forget dataset is closed to the ones of the test dataset.

The evaluation process is explained in Section V-C.

2) Model analyzed: a ResNet18: We consider the model ResNet18 [25]. It is a deep convolutional neural network used for image classification. It contains 18 main layers, including convolutional 2D layers (called Conv2D below), pooling layers and fully connected layers. 2D convolutions start with a kernel, which is simply a small matrix of weights. This kernel "slides" over the 2D input data, performing an elementwise multiplication with the part of the input it is currently on, and

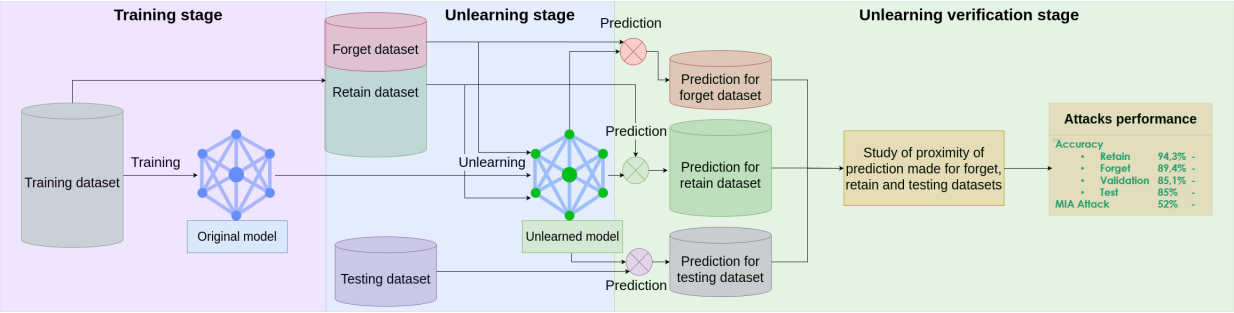


FIG. 1: Unlearning challenge addressed in the paper.

then summing up the results into a single output pixel. Pooling layers provide an approach to down sampling feature maps by summarizing the presence of features in patches of the feature map. Moreover, in ResNet18, there are several residual blocks, each consisting of few convolutional layers. These blocks help mitigate vanishing gradient problem [26].

B. *Unlearning approaches studied*

The models presented in this section were approaches conceived by participants in the NeurIPS 2023 challenge [20], which considers a realistic scenario in which an age predictor has been trained on private face images, and, after training, a certain subset of the training images must be forgotten to protect the privacy or rights of the individuals concerned [20]. The datasets used in the kaggle competition have not been disclosed, so we have studied, adapted and compared theses approaches on another dataset, CIFAR10 [27].

1) Distillation approach [28]: The distillation model is based on the machine learning technique known as knowledge distillation, which is used to transfer knowledge from a large and complex deep neural network (referred to as the "teacher model") to a smaller and more efficient model (referred to as the "student model") [29], as illustrated in Figure 2.

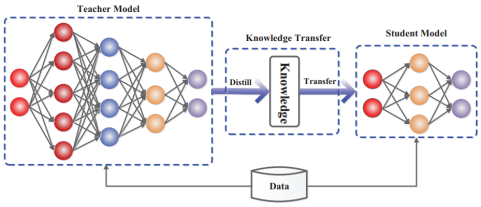


FIG. 2: The generic teacher-student framework for knowledge distillation [29].

In this approach (see Figure 3) after a partial re-initialisation of the model weights, the main idea is to directly mimic the final prediction of the teacher model on the retain dataset. The aim is to maintain the high accuracy and performance of the original model for the information we wish to retain, while reducing the model's capacity to correctly predict the group of data that has been decided to be forgotten.

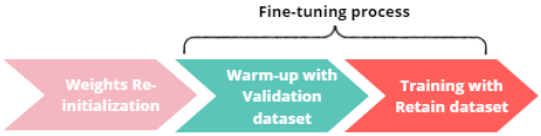


FIG. 3: Steps for the distillation approach.

The initial phase involves resetting the first and last layers of the original model. These two layers are chosen because the first layer significantly influences the rest of the model layers, and the last layer determines the model's final output distribution. With this reset step, we enable the model to deviate from its original state.

The next step will be fine-tuning the model, starting with a warm-up process where the knowledge distillation can be evidenced by obtaining the predictions of the teacher model with respect to the data of the validation dataset and proceed to use the Kullback-Leibler (KL) divergence loss function, or relative entropy, normally used to compare two data distributions corresponding to predicted data and true labels [30]. In this case the teacher information is considered as true labels and is compared with the student's prediction, allowing the student to get as close as possible to it. Continuing with the fine tuning on the retain dataset, the same knowledge distillation procedure performed previously is carried out, but in this case soft predictions will be used and not hard predictions. The predictions will be used to feed the loss in three different ways: Soft Cross entropy Loss, Cross Entropy loss [31] and KL-loss. This final procedure allows to maintain the performance of the original model.

2) Rotate approach [32]: The rotation approach involves retraining the model using a modified version of the original model, maintaining high accuracy and performance. The process is divided into two important steps, as shown in the Figure 4.

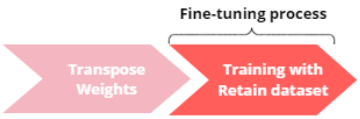


FIG. 4: Steps for the rotate approach.

Initially, the model will undergo unlearning with a modification that involves transposing all weights in Conv2D layers. This process helps in forgetting samples in the forget-set, enabling the reuse of valuable features from the original model. Finally, to refine the model, fine-tuning is performed using Cross Entropy Loss [31].

3) Pseudolabeling approach [32]: The pseudolabeling approach involves retraining the model using a modified version of the retain dataset taking into account the performance of the forget dataset with the original model. The process carried out by this approach, which can be seen in the Figure 5, starts with first, they store the inference result on the forget dataset using the original model in three different ways:

- Store the inference of the original model on the forget dataset;
- Perform a naive unlearning by fine tuning the original model on the retain dataset alone and store the inference of this model on the forget dataset;
- Re-initialize the original model and retrain it on the retain dataset during a few epochs and store the inference of the retrained model on the forget dataset.

Afterwards they define pseudo labels for the forget dataset, such that data on which the classification is not quickly learned from scratch are defined erroneously. Pseudo-labels are set to the predictions of the fine tuned model except when the fine tuned model is correct and the retrained from scratch is wrong with a low logits entropy. In that case, pseudo labels are set to the retrained from scratch model predictions.

Finally, they re-initialize the weight of the Linear layer of the original model and train it on the forget dataset pseudo-labeled using Cross Entropy Loss [31].

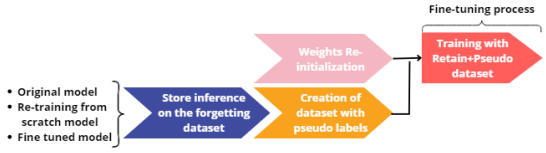


FIG. 5: Steps for the pseudolabeling approach.

This approach was combined with the Rotate approach, describe above in IV-B2.

4) Pruning approach [33]: In this approach, the unlearning is achieved by increasing the sparsity of the model, guided by the data pruning process, which consists of identifying and removing unnecessary weights and connections from the model [34]. This approach is performed in two main steps, the first one of weight re-initialization using weights pruning and the second one fine tuning the model, as shown in the Figure 6.

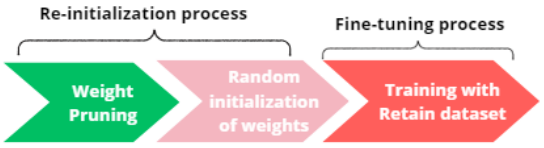


FIG. 6: Steps for the pruning approach.

Regarding the first step of re-initialization we will increase the dispersion on Conv2D and Linear layers. For this, the weights corresponding to these layers are collected, then the pruning criterion L1 Unstructured is chosen, in which the L1 norm of each weight is calculated. They are listed in ascending order and the smallest values are eliminated. In the approach 99% of the weights were chosen to be set to zero, and only 1% of the most relevant weights for the model are kept. A random reset of these zero weights is then performed. Finally, following the same process as previous approaches, we proceed with fine-tuning on the retain dataset, combining Cross Entropy Loss [31] and Custom Loss (Mean squared Error (MSE) on logits entropy) [31].

5) Deviation approach [35]: This approach consists of preserving, on average, the global information present in the original trained model, while introducing noise to the model weights.

This approach is performed in two main steps, first by deviating randomly the parameters of the convolutional layers from their real state, then performing fine tuning on the retain dataset.

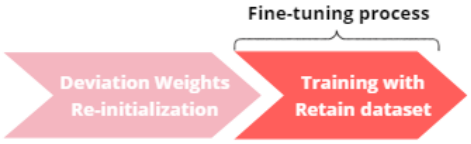


FIG. 7: Steps for the deviation approach.

In the first step, we replace each weight w_i of the convolutional layers by an observation taken from the Gaussian distribution with expectation w_i and standard deviation σ. σ, is chosen arbitrarily at 0.6. This approach aims to mimic the inherent uncertainty in the weights of the neural network from the outset, providing a statistical basis that can influence the convergence and overall performance of the model during training. For the second step, we continue the training process on the retain dataset using an optimizer Stochastic gradient descent (SGD) algorithm [36] and the Cross Entropy Loss. Besides, before starting the final training epoch we introduce a little noise in the weights, in this case using $\sigma = 0.005$. This additional perturbation helps prevent the optimizer from getting stuck in local minima of the loss function by modifying the values of the model's weights.

6) Gradient approach [37]: This approach aims to use the forget dataset to analyze how it influences the gradients in comparison with the retain dataset similarly to the Single-shot Network Pruning (SNIP) method [38]. The gradients collected

are used as input to perform the re-initialization of the model. This approach is performed in three main steps.

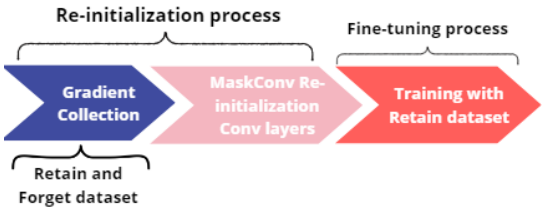


FIG. 8: Steps for the gradient approach.

The first step consists in analyzing the gradients obtained from the predictions of both the forget dataset and the retain dataset. They uses the cross-entropy loss and compare the gradient descent on the retain dataset with the gradient ascent on the forget dataset (see illustration on Figure 9).

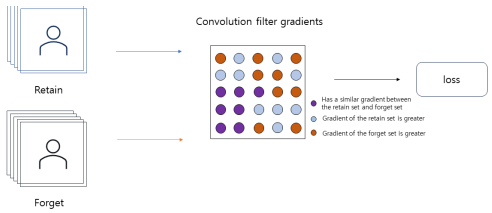


FIG. 9: First Step for the gradient approach [37].

Based on the gradient information collected in the first step, for each convolution filter a mask is created to keep only the 30% of its most similar gradients (see illustration on Figure 10). The weights corresponding to the most similar gradients are re-initialized using HE Initialization [39]. Besides, the convolutions are replaced by MaskConv [40] using these masks, in order allow to focus the training on the selected weights.

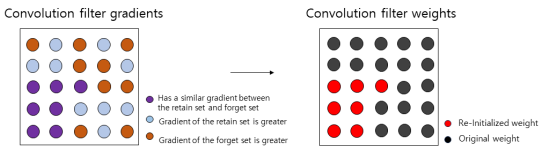


FIG. 10: Second Step for the gradient approach [37].

Finally, following the same process as previous approaches, we proceed with fine-tuning on the retain dataset, using Cross Entropy Loss with Linear Scheduler. The cosine annealing scheduler was also used but did not provide better results for the kaggle competition nor in our experiment

7) Divergence approach [41]: This approach aims to use the forget dataset, to do both the forgetting of it and to do the fine tuning. It is composed of three fundamental steps described in Figure 11.

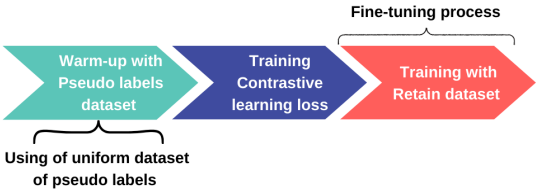


FIG. 11: Steps for the divergence approach.

In this case, the first step is a warm-up process that involves training the model on the forget dataset to maximize the divergence between the model's predictions and the correct labels. This will be achieved training the model using KL-loss between output logits and uniform pseudo labels, which are defined as ones.

Finally in the process to maximize prediction accuracy, each epoch involves two key procedures. First, we apply the forget and retain datasets to implement Contrastive Learning Loss [42]. This method refines the model by adjusting the relational distance between the samples in these datasets, either by pushing them apart or pulling them closer together. The second procedure is a fine tuning phase o the retain dataset using Cross Entropy Loss. This dual-step ensures optimal adjustments for improved prediction accuracy.

8) Comparison: All the approaches contain similarities and differences. A synthesis of the different mechanisms used in each approach are summarized in the Table I.

V. Experimental Settings

A. Dataset

In order to illustrate our approach in a standard and reproductible context, we decide to use the dataset CIFAR-10 [27] dedicated to image classification. This dataset contains 60000 32×32 color images divided in 10 classes : airplanes, automobiles, birds, cats, deers, dogs, frogs, horses, ships and trucks. We have an initial division where the training dataset corresponds to 50,000 images, and the remaining 10,000 images correspond to the held-out dataset. From this initial training dataset, 90% will correspond to the retain dataset, and the other 10% corresponding to 5,000 images are the data to be forgotten (forget dataset). As for the held-out dataset, it will be divided into 80/20 distribution for the test and validation datasets respectively.

B. Experiment Objective

The focus of this article is about comparing the performance of different machine unlearning approaches. The comparison consider the runtime, the final outcomes of the model on the different datasets and its robustness. Our aim is to highlight the approaches offering the best trade-off.

C. Evaluation Metrics

To evaluate the performance of our models, we use common metrics typically employed in the training of neural networks, such as accuracy and loss. However, in this case, we pay attention not only to the training and validation datasets but

Approach	Re-init. Weights	Re-init. Gradients	Fine Tuning on Valid	Fine Tuning on Forget	Fine Tuning on Retain	Competition Place
Distillation	✓		✓		✓	6
Rotate	✓				✓	5
Pseudo				✓	✓	5
Pruning	✓				✓	4
Deviation	✓				✓	3
Gradient	✓	✓			✓	2
Divergence				✓	✓	1

TABLE I: Synthesis of the Different Mechanisms Used in Each Approach.

also to the test, retain, and forget datasets to observe their behavior during each epoch. In addition to these metrics, we analyze runtime efficiency and AUC (Area Under the Curve) scores of ROC (Receiver Operating Characteristic) Curve in the case of a membership inference attack (MIA), which will be explained below.

1) Accuracy on retain, forget and validation datasets: The accuracy measure the ability to the model to well predict. Our objective is double:

- Preserve (or increase) the accuracy for the validation and the retain datasets compare to the original model.
- Decrease the accuracy for the forget dataset compare to the original model. Although it should not be significantly lower or similar than the validation accuracy.

Denote that we consider the value on the different losses, but for the sake of brevity and clarity, in this article we will only display accuracies.

2) Run Time Efficiency: Run Time Efficiency is a measure of the computational resources and time required to perform a specific task, such as training, inference, or unlearning in machine learning models.In machine unlearning, run time efficiency is critical as it affects the practicality and scalability of the unlearning processes [1]. Efficient unlearning techniques ensure that data can be removed quickly and with minimal computational overhead, making it feasible to implement unlearning in real-world applications where timely compliance with data removal requests is essential. This also highlights that we want the unlearning to take much less time than retraining the model from scratch.

3) AUC Score of ROC Curve under MIA Attack: The Membership Inference Attack (MIA) process begins by grouping data from the test and forget datasets, followed by their comparison. It seeks to determine if a specific sample was part of the model's forget dataset, based on the loss assigned by the model. Losses for each sample are computed, and binary labels (1 for forget, 0 for test) are assigned. These losses and labels are then used to train a logistic regression model that learns to differentiate between the two datasets. Stratified cross-validation is used to train and evaluate the attack model, ensuring that each subset maintains the original proportion of test and forget samples. Finally, the attack model's accuracy is calculated for each subset, reflecting how well it distinguishes between forget and test samples.

By analysing the classification threshold of the attack model with obtain the AUC of ROC Curve of a MIA attack. The ROC Curve represents the true positive rate against the false positive rate at each classifier threshold setting.

An AUC close to 0.5 means that the attack model is not able to distinguish between the forget and the test datasets. Thus it will be difficult for an attacker to infer information on the unlearned dataset from the unlearned model.

VI. Results

The variability of the results is assessed across different splits between the Retain dataset and the Forget dataset, i.e., how each model behaves with different subsets of data. This process was repeated over 10 iterations.

A. Model Performance

During the forgetting process performed by each approach, an analysis of accuracy and loss was conducted at each epoch, with the aim of observing particularly the decrease in accuracy and the increase in loss during the early stages of forgetting, due to the techniques that directly affect the model's head or layers. Subsequently, the increase in accuracy and the decrease in loss were observed during the fine-tuning process performed by each approach. The difference will be especially noticeable in the final accuracy result.
At the top of Figure 12, we provide the boxplots of the accuracies of each approach on the retain and validation datasets, both with a random and a target forget sets, to allow comparison with the original model, i.e.,before the unlearning process. In addition to the approaches already studied and the original model, we will also provide the accuracies of a model we have termed "simple", as its unlearning process involves only fine-tuning on the retain dataset without any additional processing, unlike the other approaches and of the original model. There, we do not represent gradient method, which is significantly worst than the other methods. Keeping it on would greatly reduce the readability of the figure. Useful unlearning is not supposed to degrade the model's performance, so the retain and validation accuracies should remain close to those of the original model.

At the bottom of Figure 12, we plot the ratio between the accuracies on the forget and validation sets. An effective unlearning approach should yield a ratio close to 1. As shown

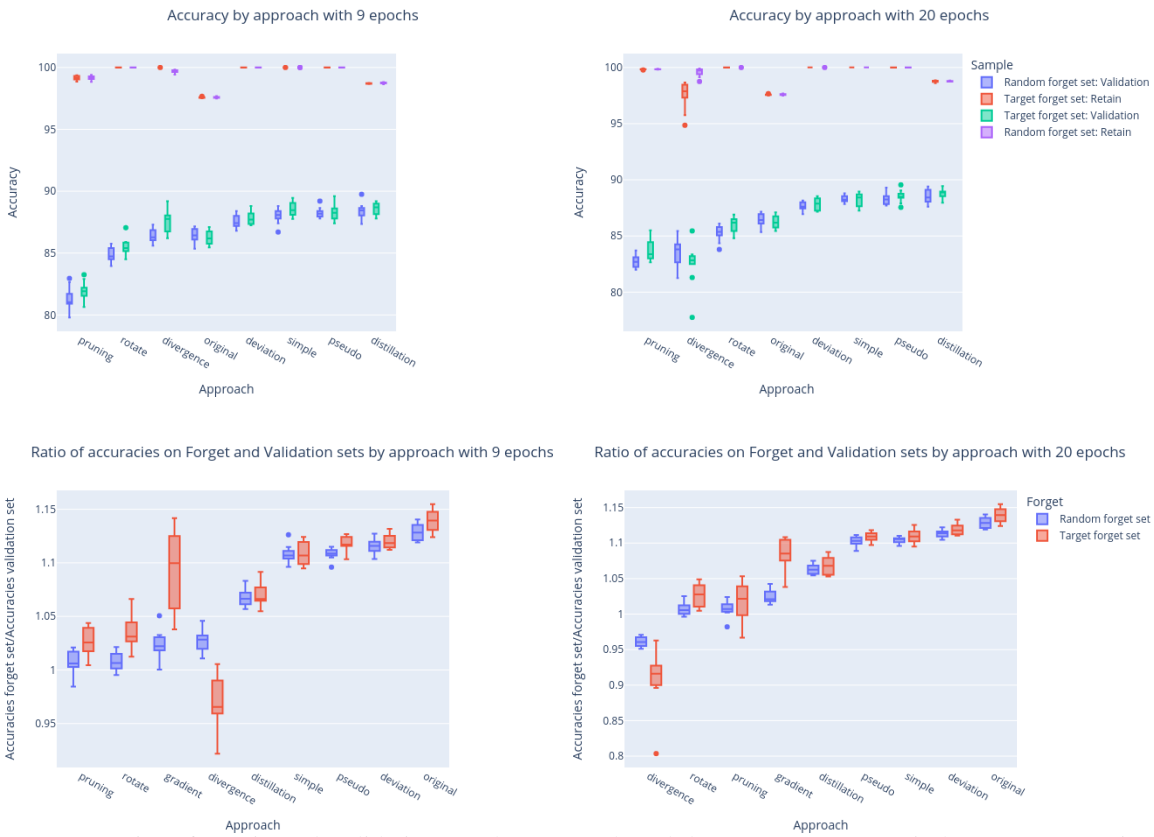


FIG. 12: (Top) Accuracies of Retain and Validation sets by approach and dataset. (Bottom) Ratio between accuracies on forget and validation sets by approach and dataset. Both respectively with 9 (Left) and 20 (Right) epochs.

in the figure, the original model displays differing accuracies between the validation and forget sets, which is expected, as no unlearning method has been applied and the forget dataset was randomly selected.

For the distillation, pseudo-labeling, simple, and deviation approaches, the accuracies between the forget and validation datasets are quite different, indicating incomplete unlearning. For all these approaches, considering 9 or 20 epochs lead to similar results. Gradient approach requires a greater number of epochs than the others unlearning approaches to attain a similar performance to that of the original model on the retain dataset. A plausible hypothesis is that the utilization of gradients for unlearning may be an effective strategy, yet it poses a challenge for the model to regain its original performance, potentially due to the complexity of the unlearning process. Rotate and pruning approaches allow to make accuracies of validation and forget datasets close. However, it degrades the performance on the validation datasets (pruning more than rotate). Considering more epochs allow a smaller degradation and reduce variability. Divergence is more dependent than the others approaches to the number of epochs. With 9 epochs, the accuracies on the validation datasets are a little stronger than the accuracies of the original model. The accuracies on the forget datasets are stronger but close to the accuracies on the validation datasets. With 20 epochs, the accuracies on the validation datasets are a little smaller than the accuracies of the original model (and with more variability). The accuracies on the forget datasets are smaller but close to the accuracies on the validation datasets. This let us think that with a good choice of the number of epoch, we should be able to have similar accuracies on the forget and validation datasets and accuracies on the validation dataset close to the accuracies of the original model.

Furthermore, we computed the approaches with a reduced number of epochs, specifically 5, and observed that the results exhibited negligible variation compared to those achieved at 9 epochs for most of the approaches, suggesting that the models had already converged to a stable performance at this earlier stage. This suggests that the models are able to learn and unlearn effectively within a relatively small number of epochs, and that further increases in the number of epochs may not yield significant improvements in performance. Among the approaches evaluated, Rotate, Pruning, and Divergence emerged as the best performing.

B. Run time efficiency

Figure 13 provides the run time of each approach for the ten repetitions with 9 epochs. First, all approaches exhibit a significant reduction in computational time, with none requiring more than half the time needed to re-train the original model,

which takes around 41 minutes and 41 seconds. Furthermore, the simple, deviation and divergence models demonstrate the fastest unlearning times, a characteristic that underscores their robustness in terms of run time efficiency. Pseudo labeling takes more time and divergence is faster than the others unlearning approaches.

FIG. 13: Run Time by approach with 9 epochs, as a percentage of the training time of the original model.

C. MIA

Consistent with the previous metrics, we evaluated the robustness of the unlearning approaches against MIA Attacks considering the 10 experiment repetition. The results, presented in the left part of Figure 14, illustrates the variability in AUC across each approach, as well as the original model. It is noteworthy that an AUC closer to 0.5 indicates a more effective unlearning process, as the attack is unable to differentiate between samples from the forget dataset and those from the test dataset, suggesting that the forget data is indistinguishable from unseen data to the model. This evaluation provides a robust assessment of our approach's ability to protect sensitive information, and the results demonstrate its effectiveness in mitigating MIA Attacks. Some results of Figure 14 are coherent with Figure 12. All models demonstrate a reduction of MIA AUC Score compared to the original model. The MIA AUC Score of simple, deviation, pseudo and distillation is stronger than for the other unlearning approaches. Moreover, the MIA AUC Score of rotate, pruning, gradient and divergence is close to 0.5, the attack perform almost as well as a random attack. The attack on the pruning seems the closer to a random attack.

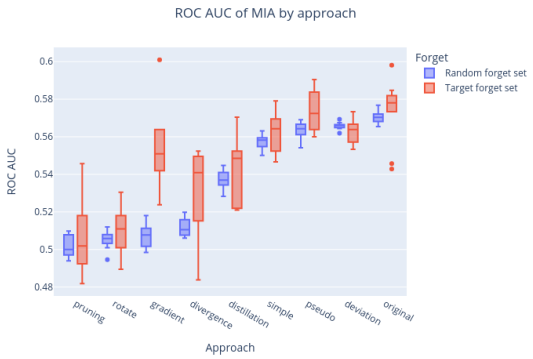


FIG. 14: AUC scores in case of MIA Attack by approach with 9 epochs with forget set randomly chosen and forget set targeted on red images.

D. Realistic forget dataset

To push our benchmark further we have compared the different approaches with a target forget dataset. We have selected 525 pictures having a lot of red, to be part of the forget dataset. To select these images, we have used the process proposed by [43]. The accuracies on the retain and the validation sets as well the ratio between the accuracies on the forget and the validation sets are given in Figure 12. In the majority of case, the range of the boxplots are larger than when the forget set is chosen randomly. Despite this, the overall behavior of the methods remains close to that observed in the random choice context. Methods like rotate and pruning whose forget and validation accuracies are close in the case of random forget set selection keep them close in the case of targeted forget set selection. For the sake of clarity, we have not shown the results with 5 epochs in the Figure 12. However, for the divergence method, with 5 epochs, the accuracies of the forget and validation sets are the closest, while maintaining a high level of performance on the retain set. On the right part of Figure 14, we see that the AUC with target forget dataset has more variability than random forget dataset. Besides, the MIA is almost random when rotate or pruning methods are used.

VII. Conclusion

Our study highlights the importance of striking a balance between performance, time, and defense against attacks in machine unlearning models. While achieving high performance is crucial, it must be accomplished within a reasonable time frame to maintain efficiency. Moreover, defense against attacks is vital to prevent malicious actors from manipulating the model's outputs or stealing sensitive data. Our results identify that rotate, pruning, and divergence approaches offer a promising balance between these three aspects, although the variability of the divergence method's results must be considered. These findings provide a foundation for future research and development in the field of data removal and machine learning.

We consider both a randomly selected forget dataset and a more realistic representation of real-world data privacy scenarios where the forget dataset is based on specific image characteristics, such as color. Identifying these features allows for more accurate data removal, as demonstrated in [21] for future researches.

Future research directions include exploring class removal and the use of alternative datasets, as well as developing metrics such as the Amnesis index [44] and MIA attack based on LiRA [23] to determine the most robust method for defending against attacks.

References

[1] T. Shaik, X. Tao, H. Xie, L. Li, X. Zhu, and Q. Li, "Exploring the landscape of machine unlearning: A comprehensive survey and taxonomy," 2024. [Online]. Available: https://arxiv.org/abs/2305.06360

[2] S. Minaee, A. Abdolrashidi, H. Su, M. Bennamoun, and D. Zhang, "Biometrics recognition using deep learning: A survey," 2021. [Online]. Available: https://arxiv.org/abs/1912.00271

[3] C. Ahmed, M. Umer, S. Liyakkathali, M. T. Jilani, and J. Zhou, *Machine Learning for CPS Security: Applications, Challenges and Recommendations*, 12 2020, pp. 397–421.

[4] K. Z. Liu, "Machine unlearning," https://ai.stanford.edu/~kzliu/blog/unlearning, 2020, accessed: 2023-02-20.

[5] A. Golatkar, A. Achille, and S. Soatto, "Eternal sunshine of the spotless net: Selective forgetting in deep networks," 2020. [Online]. Available: https://arxiv.org/abs/1911.04933

[6] B. Buet, G. P. Leonardi, and S. Masnou, "Weak and approximate curvatures of a measure: a varifold perspective," 2020. [Online]. Available: https://arxiv.org/abs/1904.05930

[7] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot, "Machine unlearning," 2020. [Online]. Available: https://arxiv.org/abs/1912.03817

[8] D. M. Sommer, L. Song, S. Wagh, and P. Mittal, "Towards probabilistic verification of machine unlearning," 2020. [Online]. Available: https://arxiv.org/abs/2003.04247

[9] N. Li, C. Zhou, Y. Gao, H. Chen, A. Fu, Z. Zhang, and Y. Shui, "Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects," 2024. [Online]. Available: https://arxiv.org/abs/2403.08254

[10] L. Riecke, "Unmasking the Term 'Dual Use' in EU Spyware Export Control," *European Journal of International Law*, vol. 34, no. 3, pp. 697–720, 09 2023. [Online]. Available: https://doi.org/10.1093/ejil/chad039

[11] J. Xu, Z. Wu, C. Wang, and X. Jia, "Machine unlearning: Solutions and challenges," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 8, no. 3, p. 2150–2168, Jun. 2024. [Online]. Available: http://dx.doi.org/10.1109/TETCI.2024.3379240

[12] Y. Xu, "Machine unlearning for traditional models and large language models: A short survey," 2024. [Online]. Available: https://arxiv.org/abs/2404.01206

[13] Y. Qu, X. Yuan, M. Ding, W. Ni, T. Rakotoarivelo, and D. Smith, "Learn to unlearn: A survey on machine unlearning," 2023. [Online]. Available: https://arxiv.org/abs/2305.07512

[14] Y. Cao and J. Yang, "Towards making systems forget with machine unlearning," in *2015 IEEE Symposium on Security and Privacy*, 2015, pp. 463–480.

[15] C. Chen, F. Sun, M. Zhang, and B. Ding, "Recommendation unlearning," 2022. [Online]. Available: https://arxiv.org/abs/2201.06820

[16] M. Kurmanji, P. Triantafillou, J. Hayes, and E. Triantafillou, "Towards unbounded machine unlearning," 2023. [Online]. Available: https://arxiv.org/abs/2302.09880

[17] L. Graves, V. Nagisetty, and V. Ganesh, "Amnesiac machine learning," 2020. [Online]. Available: https://arxiv.org/abs/2010.10981

[18] T. Baumhauer, P. Schöttle, and M. Zeppelzauer, "Machine unlearning: Linear filtration for logit-based classifiers," 2020. [Online]. Available: https://arxiv.org/abs/2002.02730

[19] T. T. Nguyen, T. T. Huynh, P. L. Nguyen, A. W.-C. Liew, H. Yin, and Q. V. H. Nguyen, "A survey of machine unlearning," 2022. [Online]. Available: https://arxiv.org/abs/2209.02299

[20] Triantafillou, Pedregosa, Hayes, Kairouz, Guyon, Kurmanji, Dziugaite, Triantafillou, Zhao, S. Hosoya, J. C. S. J. Junior, Dumoulin, Mitliagkas, Escalera, D. Wan, Demkin, and Reade, "Neurips 2023 - machine unlearning," 2023. [Online]. Available: https://kaggle.com/competitions/neurips-2023-machine-unlearning

[21] Y. Zhang, S. Yao, S. Shen, S. Xu, T. Yang, B. Li, X. Zhang, Z. Shao, and Z. Gu, "The secret revealer: Generative model-inversion attacks against deep neural networks," 2020. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/papers/Zhang_The_Secret_Revealer_Generative_Model-Inversion_Attacks_Against_Deep_Neural_Networks_CVPR_2020_paper.pdf

[22] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," 2017. [Online]. Available: https://arxiv.org/abs/1610.05820

[23] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, "Membership inference attacks from first principles," 2022. [Online]. Available: https://arxiv.org/abs/2112.03570

[24] S. Jiang, Y. Luo, S. Zheng, Y. Yu, Z. Xu, Y. Tan, and J. Zhao, "Membership inference attacks against recurrent neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9833649

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[26] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 1998.

[27] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:18268744

[28] E. Karypidis, V. Perifanis, C. C. Nikolaidis, N. Komodakis, and P. Efraimidis. (2023) 6th place solution. [On-

line]. Available: https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/458740
[29] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation : A survey," 2021. [Online]. Available: https://arxiv.org/abs/2006.05525
[30] Encord. (2023) Kl divergence in machine learning. [Online]. Available: https://encord.com/blog/kl-divergence-in-machine-learning/
[31] A. Mao, M. Mohri, and Y. Zhong, "Cross-entropy loss functions: Theoretical analysis and applications," pp. 23 803–23 828, 2023. [Online]. Available: https://proceedings.mlr.press/v202/mao23b.html
[32] Marvelworkd and Toshi-k. (2023) 5th place solution. [Online]. Available: https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/458531
[33] S. Oleszko. (2023) 4th place solution. [Online]. Available: https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/459148
[34] J. Jia, J. Liu, P. Ram, Y. Yao, G. Liu, Y. Liu, P. Sharma, and S. Liu, "Model sparsity can simplify machine unlearning," 2024. [Online]. Available: https://arxiv.org/pdf/2304.04934
[35] S. Achour. (2023) 3rd place solution. [Online]. Available: https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/459334
[36] "Backpropagation and stochastic gradient descent method," *Neurocomputing*. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/0925231293900060
[37] D. Lee, J. Bae, and J. Kim. (2023) 2nd place solution. [Online]. Available: https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/459200
[38] N. Lee, T. Ajanthan, and P. H. Torr, "Snip: Single-shot network pruning based on connection sensitivity," in *International Conference on Learning Representations*, 2019.
[39] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *arXiv preprint arXiv:1502.01852*, 2015.
[40] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1601.06759*, 2016. [Online]. Available: https://arxiv.org/abs/1601.06759v3
[41] Fanchuan. (2023) 1st place solution. [Online]. Available: https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/458721
[42] T. Wang and P. Isola, "Understanding the behaviour of contrastive loss," 2021. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/html/Wang_Understanding_the_Behaviour_of_Contrastive_Loss_CVPR_2021_paper.html
[43] K. Bhanot, "Color identification in images," 2018. [Online]. Available: https://towardsdatascience.com/color-identification-in-images-machine-learning-application-b26e770
[44] M. M. M. K. Vikram S Chundawat, Ayush K Tarun, "Zero-shot machine unlearning," 2022. [Online]. Available: https://arxiv.org/abs/2201.05629

Machine Learning Toolbox for Anomaly Detection in Low-Flying Aircraft Surveillance

1st Melvyn Pirolley
Université de Franche-Comté
FEMTO-ST Institute, CNRS
Belfort, France
melvyn.pirolley@univ-fcomte.fr

2nd Raphaël Couturier
Université de Franche-Comté
FEMTO-ST Institute, CNRS
Belfort, France
raphael.couturier@univ-fcomte.fr

3rd Aymeric Cretin
Smartesting
Besançon, France
aymeric.cretin@smartesting.com

4th Antoine Chevrot
Smartesting
Besançon, France
antoine.chevrot@smartesting.com

5th Thomas Dubot
Universite de Toulouse
ONERA, DTIS
Toulouse, France
thomas.dubot@onera.fr

***Abstract*—The increasing density of low-flying aircraft and the development of new technologies for light aviation bring new challenges for air controllers. However, an important part of today's air traffic control relies on the insecure Automatic Dependent Surveillance-Broadcast (ADS-B) protocol. It cannot be trusted because anyone can read, emit, or modify ADS-B messages with a little equipment. This weakness can give rise to various attacks and a new security system should be developed for ADS-B to avoid a complete stop of the traffic.**

This work will present two attack scenarios and provide multiple methods based on machine learning to face those attacks automatically. The two scenarios cover spoofing attacks and ghost detection. This article also presents a few additional tools, developed in the context of our research project, for anomaly detection and ADS-B visualization. All those algorithms have the objective of constituting a complete toolbox for low-altitude ADS-B anomaly detection. Overall, all attacks can be detected with an accuracy of over 96%.

***Index Terms*—Cybersecurity, Machine learning, ADS-B, low-altitude air traffic**

I. Introduction

With the regular arrival of new low-altitude technologies, the density of low-altitude traffic increases. More and more projects are attempting to develop new lightweight aerial vehicles for transporting people or goods. In recent years a lot has been written about parcel delivery by drones [1]. More recently, during the Paris 2024 Olympics Games, the city has expressed its ambition to equip itself with an electric vertical take-off and landing aircraft (eVTOL) to provide rapid transport between a few points of interest. All these examples testify to the enthusiasm for the development of light transport systems. It will not be long before air traffic control systems are heavily impacted. These solutions are lighter than conventional air traffic, and will probably rely on protocols such as ADS-B (Automatic Dependent Surveillance-Broadcast) due to the very low altitude of these aircraft.

However, as those protocols are completely open, they are highly likely to be attacked [1]. These light transportation systems could be used to lead attacks with potentially dangerous consequences for the population. For example, eVTOL could be targeted by saturation attacks. As they act like taxis, they always follow the same route, and saturation of an eVTOL route by ghost aircraft would lead to the impossibility of ensuring air traffic control in the area, which could result in the suspension of eVTOL service.

That is why this study focuses on securing low-altitude air traffic control. Numerous attacks are possible, such as injecting phantom aircraft to flood the network, distorting trajectories, emitting false alarm signals, making an aircraft disappear, impersonating another aircraft, etc. The aim is therefore to provide a complete architecture based on deep learning and other technologies for detecting and fixing anomalies in low-altitude ADS-B data. This work is part of a research project, founded by the DGA (French defense procurement agency), named DApIA. It is the culmination of earlier work carried out as part of the GeLeaD (see Acknowledgements on the bottom part) [2, 3]. These three publications deal with ADS-B anomaly detection, but their main limitation is that they concentrate on commercial traffic, whereas the present study focuses on low-level traffic.

This article begins by presenting some other works on ADS-B anomaly detection. It then details the dataset's format used for training and evaluating our models. It continues with an explanation of the methods developed to cope with spoofing and flooding attacks. The next section presents the results obtained by our models. It ends with a conclusion and

This work was supported by the DGA (French defense procurement agency) in the context of the DApIA project (project number ANR-22-ASM2-0001) related to the ANR ASTRID Maturation program (specific support for follow-up research works of projects that have received a grant from the French Ministry of Armed Forces - GeLeaD project number ANR-18-ASTR-0011). It was also partially supported by the EIPHI Graduate School (contract ANR-17-EURE-0002). Computations have been performed on the supercomputer facilities of the "Mésocentre de Franche-Comté".

[1] https://www.bbc.com/news/business-67132527

prospects for improvement.

II. Related works

In the literature, some other works have presented models dealing with anomaly detection in ADS-B time series.

In [4], the authors present an unsupervised LSTM Encoder-Decoder to detect abnormal ADS-B messages with less than 4.5% of false positive detection. The model works by encoding windows of ADS-B messages in a vector and then reconstructing the trace with a decoder model. The reconstructed trace is compared with the original trace to assess model errors. Anomaly detection is based on the fact that an abnormal trace would be incorrectly reconstructed and amplified, producing a peak error. This reconstruction error is due to the fact that the model has only been trained on normal trajectories and would therefore not react correctly to modified trajectories, since it has never seen any during training. The implementation of this type of encoder-decoder model is part of an idea that we have not yet experimented with. A similar model could be useful for confirming our model's prediction for saturation attacks. This model also has similarities with [5].

In [6], the author presents another anomaly detection method. In this situation, the next ADS-B message is predicted based on a history of n messages. When the next message is received, the algorithms compare it with the model's prediction. When the difference between the prediction and the actual message does not exceed a certain threshold, the message is normal. This study is relevant to our work because it is very similar to our model for saturation attacks. The only difference is that it predicts all ADS-B features, whereas our model only predicts latitude and longitude for saturation detection. The advantage of predicting all features is that their model will be able to detect a wide variety of attacks. To achieve that, they have defined different error thresholds for each ADS-B feature to adapt to each anomaly. On the other hand, the advantage of our model is that it specializes in fixing saturation attacks.

III. Dataset

The dataset used in this study was mainly built up from the OpenSky network history database [7]. It contains several years of ADS-B message history worldwide, making it an inexhaustible source of ADS-B data for our algorithms. In addition, other data sources were used, such as OpenStreetMap tiles and an airport dataset to obtain airport coordinates. [8].

From this database, flight data was collected around Toulouse (from 0.72561 latitudes and 43.11581 longitudes to 2.16344 latitudes and 44.07449 longitudes) at altitudes of less than 10,000 feet, to retain only low-level traffic. Messages were collected between 2022 and 2023. Finally, messages are grouped by registry code and split between landing and take-off to form flights. Each flight is saved in one CSV file containing the following rows:

- **timestamp**: Date of the message
- **icao24**: Transponder identifier
- **latitude**: Coordinates
- **longitude**: Coordinates
- **groundspeed**: Horizontal speed
- **track**: Orientation (0° is north)
- **vertical rate**: Ascensional speed
- **callsign**: Registry code of the aircraft
- **onground**: True if on ground
- **alert**: True if in alarm state
- **spi**: Special position indicator
- **squawk**: A status code
- **altitude**: Barometric altitude
- **geoaltitude**: GNSS altitude

Low-quality flights were removed from the dataset based on criteria such as duration, amount of missing values, total flight length ... Each flight lasts a minimum of 15 minutes. As the aircraft transmits ADS-B every second, flights generally contain one message per second, but due to coverage problems, some messages may be missing. The final data set contains 10,158 flights for training (12% of the training set is used for testing) and 819 flights for evaluation.

IV. Method

Anomaly detection in low-level air traffic presents many challenges compared with anomaly detection in commercial aviation. The main difference lies in the great variety of aircraft and their respective trajectories. Therefore, to address this variety, multiple attack scenarios were established, and related anomaly detection models were developed.

A. Scenario A: Spoofing attacks

Our first scenario takes place in the context of the 2023 Rugby World Cup in Toulouse or a similarly large event. Toulouse has very dense low-altitude traffic, with a lot of SAMU (French emergency) helicopters, two heliports, low-altitude flight clubs, and military areas. In this traffic, an attacker could use a drone to target the stadium. In flight, the drone would emit false ADS-B messages, to impersonate a SAMU helicopter and make air surveillance believe it to be friendly. As a result, air traffic could be severely disrupted, preventing aircraft from taking off for safety reasons.

To face spoofing attacks, a deep Convolutional Neural Network (CNN) was developed. The model is a classifier that determines the aircraft type based on its trajectory only. Then, by comparing the prediction of the model and the registry code of the aircraft, it is possible to check if the aircraft is not spoofing the identity of another kind of aircraft. For example, if a plane uses the identity of a helicopter, it will be unmasked because the model will label its trajectory as a plane.

The model architecture is described in Figure 1. To make predictions, it combines four different inputs: an ADS-B window, a take-off context, a geographic context, and the distance between the aircraft and near airports. As output, the model gives three probabilities, one for each of the defined classes:

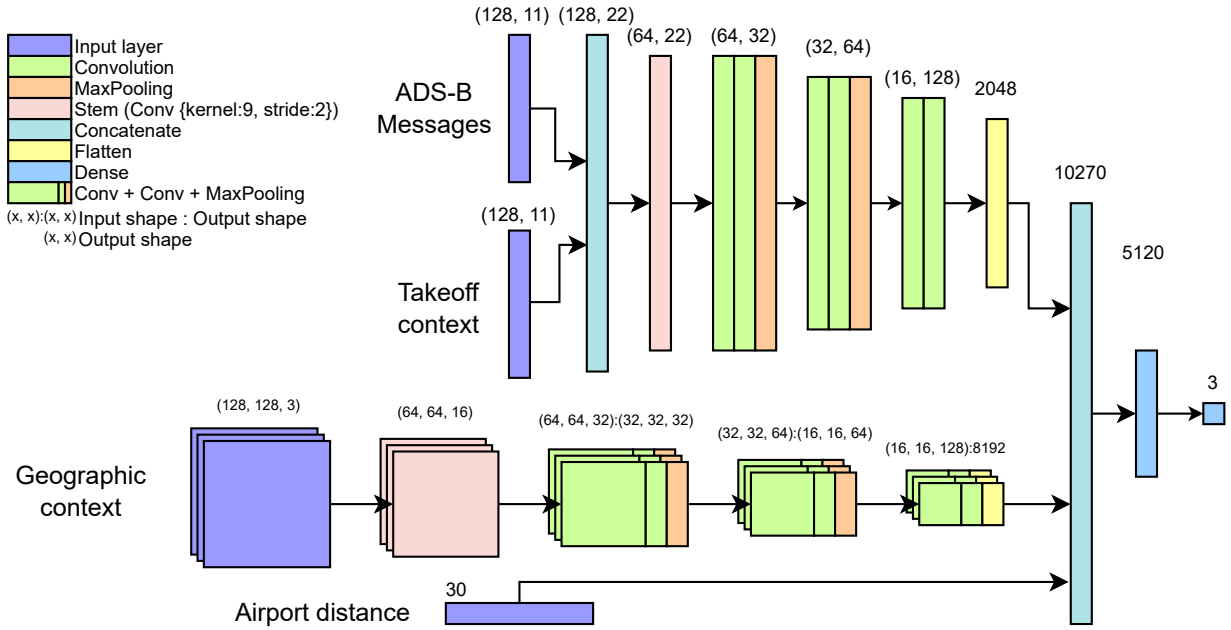


Fig. 1: CNN model architecture for Spoofing detection

- **Commercial aircraft**: at take-off and landing, when they cross the low altitude traffic.
- **Light aircraft**: regroup small aircraft, used by aeronautic clubs, tourism, transportation ...
- **Helicopter**: mainly SAMU helicopters, can also be police helicopters.

The ADS-B window corresponds to the last 128 timestamps, i.e. about 2 minutes of flight. The features used are timestamp, latitude, longitude, altitude, geoaltitude, ground speed, track, vertical speed, ground, alert, and spi. The trajectory latitude and longitude are transformed to relative coordinates on (0°, 0°). This normalization greatly helps the model as it will get all trajectories in the same format. The timestamp field is set at 0 for the current message and with negative values for the historical messages. Finally, all features are scaled between 0 and 1 using MinMaxScalers. Among the secondary data, the takeoff context corresponds to the first 128 time steps of the trajectory. This contextual data remains the same throughout the flight. It helps the model to remember the shape of the take-off so that it can continue to make good predictions in monotonous areas such as high-altitude flights. The geographic context is a small tile from the OpenStreetMap. It gives the model information about the environment (city, river, forests, ...) around the aircraft. This information is very important because aircraft regularly follow some geographic structures. For example, on take-off, SAMU helicopters follow the Garonne River to reduce noise in the city center. Finally, the distance from the airport helps the model to understand in which zone the aircraft is flying. It gives a sort of absolute coordinate location as the latitude and longitude of the aircraft are set to relative coordinates.

B. Scenario B: Saturation attacks

Our second scenario is about detecting and fixing flooding attacks. It involves the injection of ghost aircraft into the system. As there are multiple ways to realize a flooding attack this scenario is divided into two main categories: saturation and replays.

Those two types of attacks could have a large impact on dense air-lines. This is the case between Nice, Monaco, and Cannes, where many private helicopters act as taxis between these three destinations. The helicopters follow very similar trajectories, making them easier to target. With the advent of eVTOL technologies, these helicopters could be transformed into eVTOLs, for economic and noise reasons. In this case, traffic would increase drastically, making a saturation attack even more dangerous.

This first section discusses the methodology used to address saturation attacks. Saturation attacks refer to any flooding attack that injects false ADS-B messages around an aircraft's position to disrupt its tracking. These attacks can resemble those shown in Figure 2, and the goal is to identify ghost signals without filtering out real aircraft.

To achieve this, messages that are not coherent with the rest of the trajectory must be identified. A residual LSTM model was developed (Figure 3) to predict the next position of the aircraft based on its trajectory. Trained on normal trajectories, this model detects anomalies when the prediction error increases significantly.

For example, in Figure 4, the model's predictions were compared for a ghost diverging 30° from the original aircraft, another diverging 10°, and the real aircraft. The results show that the more the trajectory is modified, the farther the predictions are, leading to a higher model error.

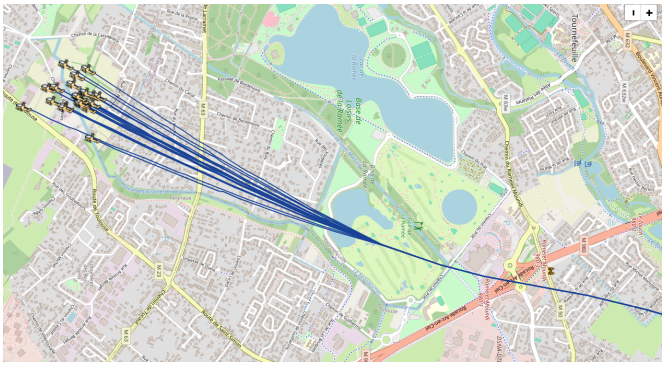


Fig. 2: Saturation exemple

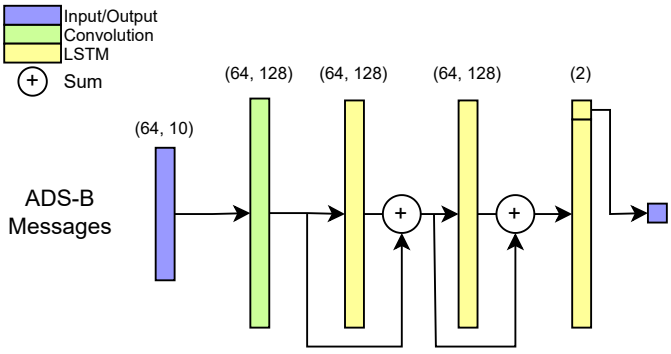
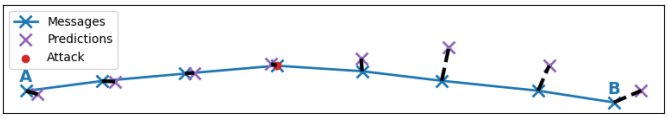
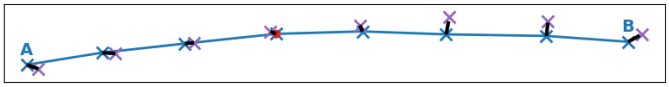


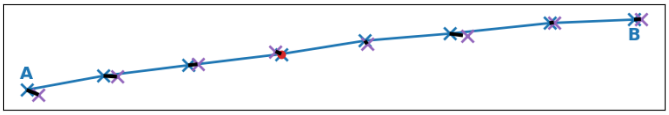
Fig. 3: Residual LSTM architecture for Saturation detection



(a) Divergence=30°



(b) Divergence=15°



(c) Real aircraft

Fig. 4: Comparison between model predictions and truth values

Based on this approach, it is possible to define an error threshold that, once exceeded, triggers an anomaly. This threshold must be fine-tuned to detect the maximum number of anomalies possible, without too many false positives.

C. Scenario B: Replay attacks

The second type of flooding attack is replay. The main difficulty in replaying trajectories lies in the fact that they are real trajectories that have occurred in the past. Replay attacks could disrupt the air traffic control of an entire airline, as an attacker could regularly inject replays. In this case, air traffic controllers would not be able to distinguish between real and fake aircraft, preventing them from ensuring the safety of real aircraft.

To detect if a flight is a replay from the past, the only reliable possibility is to compare it with a historical database and check if this precise flight already happened. Since a historical database represents a vast amount of data, pair-wise comparisons between flights would be too slow. Therefore, research focused on developing a hash table system specifically for storing trajectories. The approach was inspired by the Shazam app, which addresses a similar task by recognizing short extracts of time series in very large databases [9]. However, the Shazam hash method differs significantly because it deals with audio data, whereas our focus is on trajectories.

The hash algorithm must meet certain constraints. Firstly, the algorithms must be able to detect within a short window of a few minutes whether the trajectory is a trade-in or not. Indeed, it would be too late to know whether a trajectory is a trade-in once it has been completed. So, trajectories are split into small sub-windows of 32 timesteps. Secondly, as the attacker may use replays from a trajectory that happened somewhere else, our hash function should be resilient to basic transformations such as translation, rotation, scaling, and symmetries. To follow this constraint, every trajectory window is converted into a fingerprint before being hashed. The fingerprint is a series of micro right and left turns. The benefit of this transformation is that it makes trajectories invariant to nearly every transformation except symmetries. Additionally, it simplifies the trajectory significantly while maintaining its uniqueness. Finally, even though fingerprints are affected by symmetries, they can still be managed. Mirroring a trajectory will have the effect of inverting every right and left turn in its fingerprint. Hence, the hash function should be defined to generate the same hash value for opposite fingerprints. This can be achieved by associating right turns with the value 1 and left turns with 0, allowing the computation of a number based on the binary value of the fingerprint. This number is then reversed using the XOR operation if its value is greater than the median. The XOR operation ensures that opposite fingerprints produce the same hash value.

One weakness of the first version of this algorithm was its sensitivity to straight flights. When three points were almost aligned, the algorithm started to confuse right and left turns due to floating-point precision issues. To address this, a straight label was added to the fingerprint, acting as a wildcard. The idea is that when it is unclear whether the current message forms a right or left turn, it is ignored. The straight label generates a match whether compared with a right or left turn. This behavior is achieved by generating every sub-combination

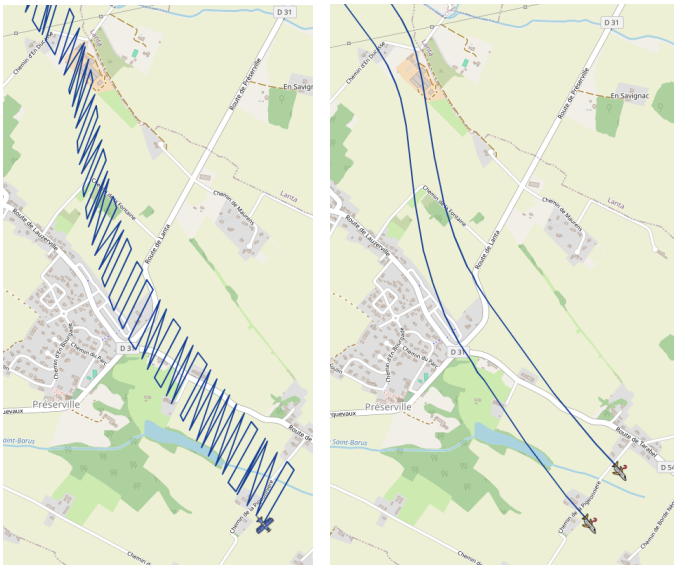
of a fingerprint by replacing wildcards with right and left and then comparing each sub-fingerprint with the hash table.

D. Additional tools

During the realization of the project, some secondary tools were developed to ease the development of the project or to fix some practical problems.

1) Trajectory Separator: has been developed to separate messages for flights having the same registry code. This can happen when ghost messages use the same registry code of a plane while it flight. Without a system to separate duplicated registry code, the flight could look like Figure 5a. The screen displays messages as if they were part of the same trajectory.

In such a situation, when several messages are received with the same register code, the algorithm creates sub-trajectories for each conflicting message. It aims to assign each of these messages to a sub-trajectory, trying to keep the trajectory as smooth and coherent as possible. To achieve this, the algorithm predicts for each sub-trajectory the position where the next message should be. Then it assigns the message to the trajectory that has the nearest predicted position. As a result, the algorithm will reconstruct trajectories as in Figure 5b.



(a) Before (b) after

Fig. 5: Before and after duplicated registry code separation

To predict the next position the algorithm applies the speed vector of the aircraft from its position (using spherical calculation). The prediction could be more accurate with a machine-learning model however it would be more time-consuming.

2) ADS-B visualizer: is a tool for visualizing ADS-B files (Figure 6). Its goal is to ease the visualization of our ADS-B flight's content. It can be used to understand the specificity of some attacks. It is also used to demonstrate the project by allowing the visualization of model predictions.

It provides a large range of functionalities such as play, pause, increasing or decreasing the time speed, going backward, and jumping to a specific time. It can manage multiple aircraft at once, and add the ability to filter aircraft by type or name. It can display all trajectories at once, to highlight the busiest airline. It can graphically display speed, altitude, and other aircraft profiles.

An online version of the visualizer is available at https://adsb-visualizer.web.app/ and the source of the project can be found in GitHub at: https://github.com/DApIA-Project/ADSB-Visualizer

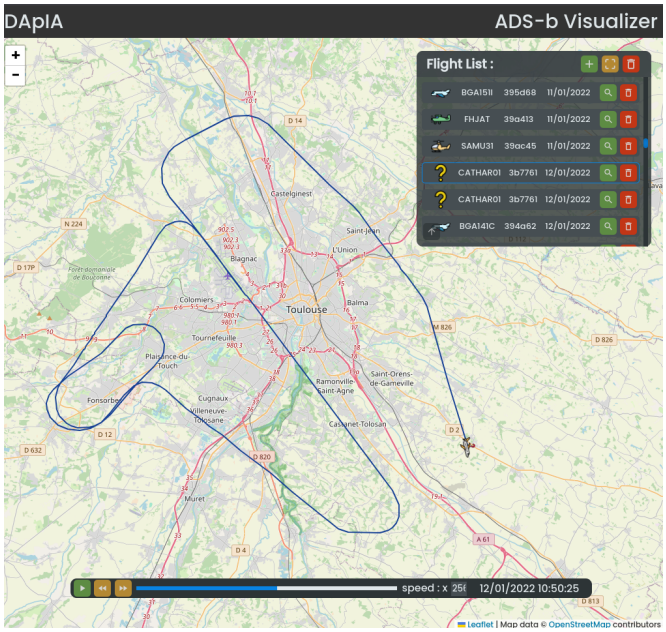


Fig. 6: ADS-B visualiser interface

3) A Python library for anomaly detection: The following text should be kept in mind:

"This tool has been developed to ensure reproducibility and simplify the use of our detection tool. It has been designed for easy integration and consists of only one function called "predict()". This function takes all ADS-B messages received in the last second as input." The input format of "predict()" is a list of dictionaries, each dictionary represents one ADS-B message and should contain the following keys: timestamp, icao24, latitude, longitude, groundspeed, track, vertical_rate, callsign, on-ground, alert, spi, squawk, altitude, geoaltitude. The function will output the same ADS-B messages, annotated with the following flags:

- **Spoofing**: True if spoofing anomaly has been detected
- **Flooding**: True if the aircraft is a ghost
- **Replay**: True the flight has been detected as a replay

To make predictions the library stores in a buffer the messages from the last call to be able to access the historical data needed by the models.

Hence, it is easy to predict every source of ADS-B. The code in figure 7 is an example of predicting an ADS-B record saved as CSV.

The library code is available in our GitHub: https://github.com/DApIA-Project/Anomaly-Detection and can

be installed using the command: `pip install AdsbAnomalyDetector`

```python
from AdsbAnomalyDetector import predict
import pandas as pd

data = pd.read_csv("./record.csv")
start = data["timestamp"].iloc[0]
end = data["timestamp"].iloc[-1]

out_df = pd.DataFrame()
for t in range(start, end):
    messages = data[data["timestamp"] == t]
        .to_dict("records")

    messages_out = predict(messages)
    for message in messages_out:
        out_df.loc[len(out_df)] = message

out_df.to_csv("output.csv")
```

Fig. 7: Code example to check anomalies on an ADS-B record saved as CSV

V. Results

A. Scenario A: Spoofing attacks

To verify the efficiency of the model, it was tested under real conditions by streaming actual flight messages and checking the model's classification. Since the model makes predictions on flight windows, it provides one prediction for each message. To aggregate predictions, the most confident prediction of the model is used to determine the aircraft type. This method yields the following confusion matrix (Figure 8).

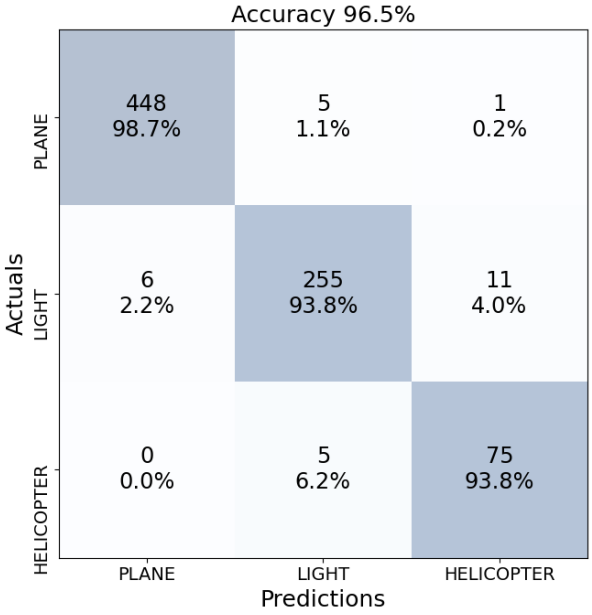


Fig. 8: Confusion matrix of the CNN model for Spoofing detection

The majority of the model's mistakes occur between light planes and helicopters. This confusion is mainly because the trajectory of a flight at a certain altitude can sometimes be very linear and exhibit similar patterns between helicopters and light planes. Given that commercial planes are heavier, it is not surprising that the model easily recognizes them.

To compare with other models, multiple experiments were conducted using other classical machine learning techniques.

Model	MSE	Accuracy (%)
Transformer	0.0373	95.6%
LSTM	0.0323	96.0%
CNN	**0.0181**	**96.5%**

TABLE I: Spoofing models comparison

Surprisingly, time series models such as LSTM and Transformers perform less effectively. This may be due to the fact that the problem is a classification task, while LSTM and Transformers are generally more suited for regression tasks. An ablation study of this model was conducted previously to demonstrate the necessity of each sub-module. As shown in Table I the best approach is to use CNN models. This model requires approximately 45 minutes of training on an Intel i7 CPU over 150 epochs. Other models utilize the same architecture as the CNN, but the Take-Off and ADS-B modules are replaced with LSTM or Transformers. The Transformers model is inspired by [10].

B. Scenario B: Saturation attacks

Detecting the correct airplane among a mass of ghost aircraft can be challenging under saturation. That is why our model focuses on detecting ghosts rather than finding the right aircraft. In an ideal situation, all the ghosts would be detected, leaving only the correct aircraft.

To evaluate the efficiency of a model, its accuracy was computed based on the percentage of ghosts that incur a higher loss than the real aircraft. The metrics used depend significantly on the difficulty of the attack and were derived from a dataset of saturation attacks generated by ourselves. Under actual saturation attacks, the model's performance could vary. Multiple models are compared in Table II.

Model	MSE	Distance error (m)	Accuracy (%)
Transformer	0.0058	22.46m	82.6%
CNN	0.0061	17.21m	89.8%
Reservoir	0.0044	13.99m	98.5%
Residual-LSTM	**0.0034**	**13.60m**	**98.5%**

TABLE II: Saturation models comparison

The residual LSTM architecture gives the best result with an accuracy of 98.5

The reservoir computing model performs surprisingly well, despite being simpler than classical neural network techniques. It achieves the same accuracy as the LSTM, with a slightly higher error distance. Its simple architecture allows for quick

predictions due to its short execution time. The reservoir architecture consists of 1000 units connected to a fully connected layer of neurons as readout [11].

While the Transformer models have good generalization, they do not outperform the LSTM or the Reservoir. The Transformer model used here comes from [12], using the classical variant from their GitHub. Its underperformance could be due to the short forecasting horizon, as transformers are typically effective for long-term forecasting. Gradient boosting algorithms were also tested but did not perform better.

C. Scenario B: Replay attacks

The hash table developed to solve replay attacks is one of our best models. It can generally determine if a trajectory is a replay with a one-minute flight window. Depending on the quality of the ADS-B coverage, it can need a bit more time to make its prediction. Thanks to the characteristics of hash tables, false positive detections are never generated. Hash collisions are possible, as the number of different hashes is $2^{31} \approx 2B$. It seems to be small as common hash systems use larger ranges like 2^{128} combination for MD5 however, it provides better reactivity of detection. Furthermore, our model's predictions are not based solely on a single match. For reliability reasons, the model only qualifies a flight as a replay if several messages correspond consecutively to the same flight in the past. So, even if it is possible to have a collision for a message, it is impossible to have several consecutive collisions with the same flight, and the model therefore never generates a false-positive detection.

In terms of detection capacity, the system is very accurate. It can determine if a trajectory is a replay even when deformed by common matrix transformations. Providing a precise accuracy score is challenging as it depends on the strength of the attack. However, with raw replay attacks, the system achieves 100% correct predictions, and on the test dataset, it correctly detects ghosts with 98% accuracy.

Finally, the algorithm is really fast, thanks to the hash table which has $0(1)$ complexity. Hence, even if the database is really large the algorithms will be able to process a flight of 15 minutes in 134ms.

One limitation of our system is that the hash table is stored on the computer's RAM. As it can become very large, it would be better to store it on the disk so as not to overflow the computer's RAM and to allow memory space for our neural network algorithms. However, this change would slow down the detection.

VI. Conclusion and perspectives

In this paper, several models for detecting anomalies in ADS-B time series are proposed. These models can work together to provide reliable detection of ghosts using ADS-B reruns or ghosts with abnormal trajectories. It also includes a tool to automatically detect whether aircraft are using the correct registration code, and to prevent spoofing attacks. All these models are brought together and run in real time in a library called AdsbAnomalyDetector.

In the future, we plan to improve the variety of attacks that could be handled by the model. For example, we could train the models to falsify trajectories generated by interpolation, which would be too smooth. We will also try to add redundancy to confirm the model's predictions by using additional models such as autocoders. Finally, we will try to fine-tune the architecture of our model and go beyond our current accuracies.

References

[1] Andrei Costin and Aurélien Francillon. *Ghost in the Air(Traffic): On insecurity of ADS-B protocol and practical attacks on ADS-B devices*. July 2012.

[2] Ralph Karam, Michel Salomon, and Raphaël Couturier. "A comparative study of deep learning architectures for detection of anomalous ADS-B messages". In: *2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*. Vol. 1. IEEE. 2020, pp. 241–246.

[3] Ralph Karam, Michel Salomon, and Raphaël Couturier. "Supervised ADS-B Anomaly Detection Using a False Data Generator". In: *2022 2nd International Conference on Computer, Control and Robotics (ICCCR)*. 2022, pp. 218–223. DOI: 10.1109/ICCCR54399.2022.9790149.

[4] Edan Habler and Asaf Shabtai. "Using LSTM encoder-decoder algorithm for detecting anomalous ADS-B messages". In: *Computers Security* 78 (2018), pp. 155–173. ISSN: 0167-4048. DOI: 10.1016/j.cose.2018.07.004.

[5] Antoine Chevrot, Alexandre Vernotte, and Bruno Legeard. "CAE: Contextual auto-encoder for multivariate time-series anomaly detection in air transportation". In: *Computers & Security* 116 (2022), p. 102652.

[6] Yunkai Zou Jing Wang and Jianli Ding. "ADS-B spoofing attack detection method based on LSTM". In: *J Wireless Com Network* 160 (2020). DOI: 10.1186/s13638-020-01756-8.

[7] *OpenSky's Historical Database*. 2013. URL: https://opensky-network.org/data/historical-flight-data.

[8] David Megginson. *OurAirports*. 2007. URL: https://ourairports.com/.

[9] Avery Wang. "An Industrial Strength Audio Search Algorithm." In: Jan. 2003.

[10] Theodoros Ntakouris. *Timeseries classification with a Transformer model*. 2021. URL: https://keras.io/examples/timeseries/timeseries_classification_transformer/.

[11] Nathan Trouvain, Nicolas Rougier, and Xavier Hinaut. "Create Efficient and Complex Reservoir Computing Architectures with ReservoirPy". In: *From Animals to Animats 16*. Cham: Springer International Publishing, 2022, pp. 91–102. DOI: 10.1007/978-3-031-16770-6_8.

[12] Haixu Wu et al. *Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting*. 2022. arXiv: 2106.13008 [cs.LG].

Improving binary diffing through similarity and matching intricacies

1st Roxane Cohen
Quarkslab
LAMSADE, CNRS, Université Paris Dauphine - PSL
Paris, France
rcohen@quarkslab.com

2nd Robin David
Quarkslab
Paris, France
rdavid@quarkslab.com

3rd Riccardo Mori
Quarkslab
Paris, France
rmori@quarkslab.com

4th Florian Yger
LITIS, INSA Rouen Normandy
Rouen, France
florian.yger@insa-rouen.fr

5th Fabrice Rossi
CEREMADE, CNRS, Université Paris Dauphine - PSL
Paris, France
fabrice.rossi@dauphine.psl.eu

***Abstract*—Reverse-engineering represents a key aspect of cybersecurity as it helps to understand unknown software or systems. From a defender's point of view, it may detect suspicious binaries or existing vulnerabilities inside organization systems. From an attacker's point of view, it offers insights into potential threats or weaknesses of a target. In particular, reverse-engineering relies on binary diffing, whose goal is to find similarities and differences between different binaries or program variants. Such a task is essential as software are constantly evolving over time. However, it is notoriously difficult, despite the large number of research papers on this subject. Many approaches have been explored, ranging from rule-based decision algorithms to advanced Deep Learning (DL) models. Such difficulty can be explained by the inherent nature of binary code, which is unstable and prone to many syntactic differences while semantics are unaltered. Determining if two binary functions are similar is already a complex task as such similarity should describe the semantics and not the syntactic properties. Obtaining a diffing, a correspondence between functions of two binaries, is even more difficult.**

In this work, we present a binary differ, called QBinDiff, and detail how it is adapted for a modular and fine-grained diffing. We conduct an empirical study about its properties, from its algorithm to its implementation, through an ablation study. We present the diffing discipline, the existing approaches, some of them being state-of-the-art, and establish a comparison benchmark between them on standard binaries. We show in particular that QBinDiff performs better than existing differs, thanks to its modularity.

***Index Terms*—Binary Similarity, Binary diffing, Graph Neural Networks, word2vec**

I. Introduction

Reverse-engineering has been widely used for defense purposes such as malware or cryptography analysis and vulnerability search. More specifically, binary diffing, a subdomain of reverse-engineering, consists in identifying similarities and differences between two binaries. It is used for malware diffing [25], patch analysis [31], program similarity [4], backdoor detection, anti-plagiarism, clustering different malware by their families or establishing a malware lineage that could be attributed to a specific Advanced Persistent Threat (APT). Diffing usually operates at the function level and tries to find an assignment between the functions of two binaries. The underlying assumption is that disassembly and function recovery are feasible. However, in practice, this is known as a complex problem [20] and in this paper, we rely on IDA-Pro [1] or Ghidra [2] for disassembly purposes. In addition, the initial binary representation may be considerably altered during the compilation process, due to optimization passes. Among them, inlining or loop unrolling alters the function control-flow logic.

Usually, binary diffing strongly relies on graphs extracted from disassembly, such as Control-Flow Graph (CFG) where nodes are Basic Blocks (BB) and edges represent execution flow within the function scope and, Call Graph (CG) where nodes are program functions and edges denote inter-procedural relationships. Consequently, many graph-centered works have been published with an increasing attention dedicated to Machine Learning (ML) and Deep Learning (DL) based approaches [3], [4], [17], [18], [26], [30]. If these methods show promising results for solving binary similarity, DL methods require large computational resources. Retraining a model or simply using it for inference may be unaffordable. Besides, binary similarity models output, given a candidate function, its closest counterpart inside a pool of functions. They do not compute the direct matching between two binaries. Consequently, obtaining a final diffing between two binaries requires to further apply a matching step using computed similarity scores. Furthermore, DL model source code or datasets are not always available and reproducibility is difficult. For these reasons, mostly BinDiff [5], [7] and Diaphora [14] are used in practice. In fact, in comparison to advanced DL models, they scale relatively well using criteria-based matching. However, for those who want finer and more adapted control over the diffing results, they may not be adapted as they are not modular or difficult to parametrize. In order to provide a more

[1]https://hex-rays.com/ida-pro
[2]https://ghidra-sre.org/

accessible diffing solution, QBinDiff[3] was open-sourced as a modular differ.

The main contributions of this work are:

- We present QBinDiff, an open-source network alignment solver, that can be applied for binary diffing. We analyze its underlying algorithm, in particular its modular functionality given the available features and parameters that can be combined to obtain fined-grained results tailored for each use case.
- We realize an ablation study of QBinDiff components to highlight how they are intertwined and how they contribute to QBinDiff results.
- We perform an empirical comparison between standard binary differs, such as BinDiff and Diaphora. We also analyze more recent DL-based methods such as Graph Matching Networks [17], Asm2vec [3], PalmTree [16] and JTrans [30]. In particular, we show that QBinDiff offers the best performances compared to other approaches.

Section II presents the concepts of binary diffing and similarity and the current approaches. Section III details the QBinDiff algorithm and its ablation study. A fair comparison between state-of-the-art diffing solutions is performed in Section IV whereas Section V establishes some discussions about the limitations of this work and further research that could be conducted. Finally, Section VI concludes this work.

II. Binary analysis

Binary diffing involves identifying similarities and discrepancies between two binary programs. Minor modifications or slight patches resulting from version updates or compilation differences should be detected by diffing tools. Binary diffing is defined as a one-to-one mapping $\phi : (\mathcal{P}, \mathcal{S}) \longmapsto \rho$, where $\mathcal{P}$ represents the primary function set of size n, $\mathcal{S}$ represents the secondary function set of size m, and $\rho : \mathcal{P} \longrightarrow \mathcal{S}$ is a partial and injective assignment function. This ensures that each function in $\mathcal{P}$ is matched to at most one function in $\mathcal{S}$. Other definitions consider various program granularities, such as BB [4]. Additionally, the mapping could be expanded to one-to-many [14] or many-to-many correspondences. Usual diffing is performed without access to source code or symbols. BinDiff [5], [7] and Diaphora [14] are widely used binary diffing tools in the reverse-engineering community. Common metrics for evaluating diffing include recall, precision, and f1-score.

Binary similarity is applied in order to find the most similar function to f inside a pool of candidate functions. A natural application is vulnerability search, because for a given vulnerable function, it is possible to find, from a database of functions, the most similar one, which probably also contains the same vulnerability. It is an active research field relying heavily on Machine Learning (ML) and Deep Learning (DL). Usual ML methods rely on precomputed features derived from assembly code or CFG. For example, TIKNIB [13] computes similarity scores using a specific distance combining various handcrafted features and BinShape [28] starts by extracting features and sorts them to obtain the top-ranked ones that are given to a decision tree.

Most importantly, DL techniques have become prevalent in this research area and are inspired by Natural Language Processing (NLP). Asm2Vec [3] is based on a refined and enhanced version of the word2vec model [24] applied on assembly text. Trex [26] and JTrans [30] are motivated by the recent success of transformers for large language models. The same holds for PalmTree [16], based on BERT but pre-trained on several assembly representation tasks, such as instruction reordering, as it is possible in assembly to switch several instructions without modifying the general semantics of the code. Graph Neural Network (GNN) is a new promising research area, that is gaining more and more popularity. The latest research articles mostly use increasingly complex GNN, with a pretrained language model used to produce initial GNN features and that relies on node assembly instructions [19]. Graph Matching Network (GMN) [17] is the first work that jointly learns graph embeddings on similar graph pairs rather than independent embeddings. Based on this principle, more GNN architectures or language models are explored [8], [18], [29]. Despite the large amount of new academic solutions, few of them provide a maintained implementation. Moreover, training these models can be quite challenging, even with the source code and the dataset. Additionally, such binary similarity models are not exempt from failures, as some adversarial attacks inspired by the ML field have been applied to disrupt specific binary similarity tools [1].

Remark. Binary diffing and binary similarity are two distinct problems. Even though they share many common aspects, their purposes are distinct. Binary diffing aims to find an assignment between the functions of two binaries and may rely on similarity scores to establish matches. Conversely, binary similarity only outputs similarity scores between pairs of candidate functions. To be further used to perform diffing, binary similarity models should be followed by a matching process to perform binary diffing.

III. QBinDiff: a modular differ

This Section presents QBinDiff and the corresponding ablation study of its components. Such ablation experimentation helps understand the intricacies of programs and their representation as graphs.

A. QBinDiff

QBinDiff [21]–[23] is a modular one-to-one differ. Given two graphs, respectively called *primary* and *secondary*, it solves an instance of the network alignment problem [6], namely finding a matching between the respective nodes of the two graphs, by expressing the link between objects similarities (as graph nodes) and relationships between these objects (graph edges), using in particular a belief propagation algorithm [21]. This algorithm represents this alignment problem using a graphical model, where nodes indicate variables of the problem and edges denote dependencies between these

[3]https://github.com/quarkslab/qbindiff

variables. It iteratively updates values associated to the variables using "messages" sent through graph edges. Once it has converged, marginal probabilities are used to determine the optimal solution of the problem. This paper focuses on the experimental aspects of QBindiff, while mathematical foundations and details are already provided [22].

Algorithm 1 QBinDiff algorithm

Require: Primary binary p , Secondary binary s, (*features*, *weights*), *parameters*=(d, s_{ratio}, α, ϵ), Optional list of *pre-passes* and *post-passes*
Ensure: Matching between p and s functions
1: $\mathcal{S} \leftarrow Anchoring(p, s)$
2: **for** $pass_i \in$ *pre-passes* **do**
3: $\mathcal{S} \leftarrow pass_i(p, s, S)$ ▷ similarity matrix refinement passes before feature extraction
4: **end for**
5: $features_p \leftarrow FeaturesExtraction(p, features)$
6: $features_s \leftarrow FeaturesExtraction(s, features)$
7: $\mathcal{S} \leftarrow Similarity(features_p, features_s, \mathcal{S}, weights, d)$
8: **for** $pass_i \in$ *post-passes* **do**
9: $\mathcal{S} \leftarrow pass_i(p, s, S)$ ▷ refinement passes after feature extraction
10: **end for**
11: $\mathcal{S} \leftarrow Decimation(\mathcal{S}, s_{ratio})$
12: *squares-matrix* $\leftarrow Squares(\mathcal{S}, GetCG(p), GetCG(s))$
13: $match \leftarrow$ *Belief Propagation*$(\mathcal{S}$, *squares-matrix*$, GetCG(p), GetCG(s), \alpha, \epsilon)$
14: **return** *match*

QBinDiff, whose algorithm is shown in Algorithm 1, consists of three main components and several parameters:

- a similarity matrix (S) encoding the similarity value between each pair of nodes between the two graphs, computed using a preset of heuristics, named *features* that describes the function and the whole program.
- a weight matrix (*squares matrix*) that encodes the induced common edges in both graphs for each possible node assignment,
- a number of user-configurable parameters, among which we distinguish the tradeoff denoted α, the sparsity s_{ratio} and the relaxation parameter ϵ.

First of all, the anchoring phase is used to pre-match imported functions. Optional passes can be defined for a specific initialization of the similarity matrix. Then, features for primary and secondary functions are computed, where *features* denote the data that will be extracted from the binaries: it can be CFG structural features such as the number of BB per function or related to the CG topology like the callee number of a function, or even the assembly instruction mnemonics or the function constants. QBinDiff offers 33 different features in total, that represent the full features set.[4] The similarity matrix S is computed with a weighted linear combination of distances over the primary and secondary feature vectors. Distance candidates are: Canberra, Euclidean, cosine and Haussmann[5]

The similarity matrix S represents the similarity between each node of the primary and each node of the secondary. Intuitively, it should encode domain-specific knowledge coming from the problem instance that the graphs are representing. For binary diffing, an entry $S[i, j]$ close to 1 means that the node i in the primary is really similar to the node j in the secondary, depending on a given metric. On the other hand, a value close to 0 indicates a high dissimilarity between the two. In most cases, the similarity matrix tends to be too large and has to be decimated by using the sparsity ratio $s_{ratio} \in [0, 1]$ that removes the lowest similarity scores that will probably not lead to a match.

The *squares matrix* can be directly derived from the graph structure and the similarity matrix S. A square is defined as a tuple of nodes (A, B, C, D) such that all the following conditions are true:

- Nodes A and D belong to the primary graph.
- Nodes B and C belong to the secondary graph.
- (A, D) is a directed edge in the primary graph.
- (B, C) is a directed edge in the secondary graph.
- Similarity scores for square nodes are positives: S[A,B]>0 and S [D,C]> 0.

The tradeoff parameter $\alpha \in [0, 1]$ is similar to a cursor that insists either on the similarity or the graph topology. $\alpha = 0.0$ means that only the graph structure is considered to compute the matches, while the similarity is disregarded. On the contrary, $\alpha = 1.0$ indicates that only the similarity matters. $\epsilon \in [0, 1]$ is a relaxation parameter that helps the Belief Propagation [21] to converge.

B. Ablation : experimental settings

We detail the experimental settings that support the ablation study of QBinDiff. We use the *Dataset-1* [18] that contains various binaries compiled with different options (compilers, compiler versions, optimization levels) from different projects: `zlib`, `unrar`, `curl`, `clamav`, `nmap` and `openssl`. For each project, we randomly split the associated binaries into two sets, named *A* and *B*. As an example, `x64-clang-7-O1-libz.so.1.2.11` will be part of *Dataset-1A* and `x64-clang-5.0-O3-libz.so.1.2.11` of *Dataset-1B*. Similarly, `x64-gcc-7-O0-unrar` will be part of *Dataset-1A* and `x64-clang-3.5-O1-unrar` of *Dataset-1B*.[6] The intersection of binaries from *Dataset-1A* and *Dataset-1B* is empty. In this ablation study, we only use the *Dataset-1A*. It contains 366 binaries from the previous projects, compiled with `clang` or `gcc`, with different versions, from optimization level `-O0` to `-Os`, that represents 425,523 functions. Only `x86-64` binaries are kept. We only diff a binary against another version of itself: this means we can diff a `x64-gcc-7-O2-nmap` against a `x64-clang-3.5-O1-nmap` but not against a `x64-clang-3.5-O1-nping`.[7] Diffing pairs were established at random given the previous conditions. Once they are established, we automatically create ground-truth by matching

[4]A complete list is available on the QBinDiff documentation website [27].

[5]The Haussmann distance is a unique function defined by QBinDiff that combines both the Jaccard index and the Canberra metric, see https://diffing.quarkslab.com/qbindiff/doc/source/params.html#haussmann

[6]Notice that each project has a different number of corresponding binaries. `unrar` has only one binary while `openssl` has 10 binaries.

[7]Such assumption is valid for binary diffing as a reverse-engineer rarely tries to find the exact mapping between two completely different binaries (as `nmap` and `nping` for example), even though they may share some functions. This assumption does not always hold for other binary problems, such as the binary similarity problem [18].

	zlib	curl	clamav	unrar	nmap	openssl
No anchoring	0.72	0.71	0.71	0.72	0.68	0.69
With anchoring	0.81	0.84	0.85	0.79	0.79	0.79
% gain	+12.5	+18.4	+19.8	+9.8	+16.2	+14.5

TABLE I: f1-score anchoring results. Parameters are default ones and the feature set is full.

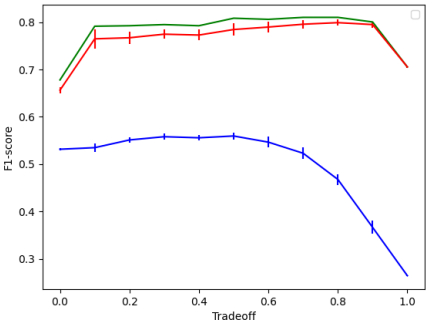


Fig. 1: QBinDiff f1-scores on `zlib` for various tradeoffs and settings.

functions with the same name. This ground-truth is globally reliable but may have limits. Compilers can inline or outline functions which add noise to the ground-truth. Next, we strip the binaries so that function names cannot be used for the diffing. At this point, there is only one remaining step before the proper diffing: the export. Most differs rely on binary exporters: BinDiff uses BinExport, Diaphora has its own export format. QBinDiff was created in order to handle different exporters and it now supports BinExport and Quokka [2], respectively developped by Google and Quarkslab. The default QBinDiff parameters p_s were previously defined as the Canberra distance, $\alpha = 0.75$, $s_{ratio} = 0.75$ and $\epsilon = 0.5$ [21].

To evaluate the differ performances, we consider three usual metrics: the recall ($\mathcal{R}$), the precision ($\mathcal{P}$) and the f1-score. They are defined as follows:

$$\mathcal{P} = \frac{TP}{TP+FP} \quad \mathcal{R} = \frac{TP}{TP+FN} \quad \text{f1-score} = \frac{2\times\mathcal{P}\times\mathcal{R}}{\mathcal{P}+\mathcal{R}}$$

with *TP* denoting True Positive, *FP* False Positive and *FN* False Negative.

Intuitively, the precision denotes how many retrieved items are relevant whereas the recall indicates how many relevant items are retrieved. Precision and recall being complementary metrics, we focus on maximizing the f1-score as it measures a trade-off between precision and recall and requires both metrics to be high.

C. Anchoring

Anchoring (step 1 in Algorithm 1) aims to use imported functions as reliable anchors, especially for dynamically-linked binaries, before any further matching step. We analyze

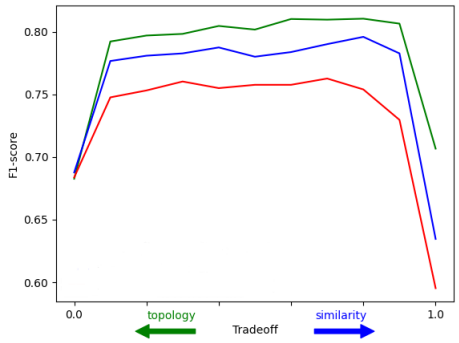


Fig. 2: QBinDiff features impact on the `zlib` project.

this functionality to determine if it helps QBinDiff to converge, and to evaluate the performance gain compared to the case where we have to match every function candidate. Results are displayed in Table I. f1-score is averaged per project, not per binary. Notice that the f1-score gain with anchoring is significant: instead of having to match every function, we can rely on the imported functions as anchor points to rather match only function clusters that depend on these anchors, which is computationally much easier.

D. Similarity and CG topology impact

A core QBinDiff parameter is the tradeoff α (step 13 in Algorithm 1) that determines how much QBinDiff should focus on the similarity or the CG topology. Then, a user may decide to focus more on the similarity than the CG structure if it provides better diffing. We choose to observe what happens when we make the tradeoff α vary and respectively, use the default QBinDiff configuration, QBinDiff with a disturbed similarity matrix, and QBinDiff with disturbed adjacency matrices. We disturb the similarity matrix by adding a uniform random noise over its elements. We replace the original adjacency matrices by modified ones using Metropolis-Hasting algorithm [10] (swapping is repeated for 2,000 iterations and self-loops are not allowed). Because these perturbations are mainly built over randomness, we repeat the diffing process with different seeds and average the results. Figure 1 plots results for the `zlib` project. We observe several aspects:

- With standard QBinDiff, the f1-score shows two brutal variations: when we increase the tradeoff from $\alpha = 0.0$ to $\alpha = 0.1$ and from $\alpha = 0.9$ to $\alpha = 1.0$. With $\alpha = 0.1$, the similarity starts to be incorporated to the diffing process. This is highlighted by the significant performance increase. Similarly, when we switch the tradeoff from $\alpha = 0.9$ to $\alpha = 1.0$, the CG topology is not considered anymore, resulting in a lack of information from the CG. Consequently, the f1-score drops suddenly. This demonstrates the necessity to consider both the CG topology and the similarity given by the features extracted from the binaries.
- When we disturb the adjacency matrices and set a lot of weight on the topology, we rely almost only on noisy

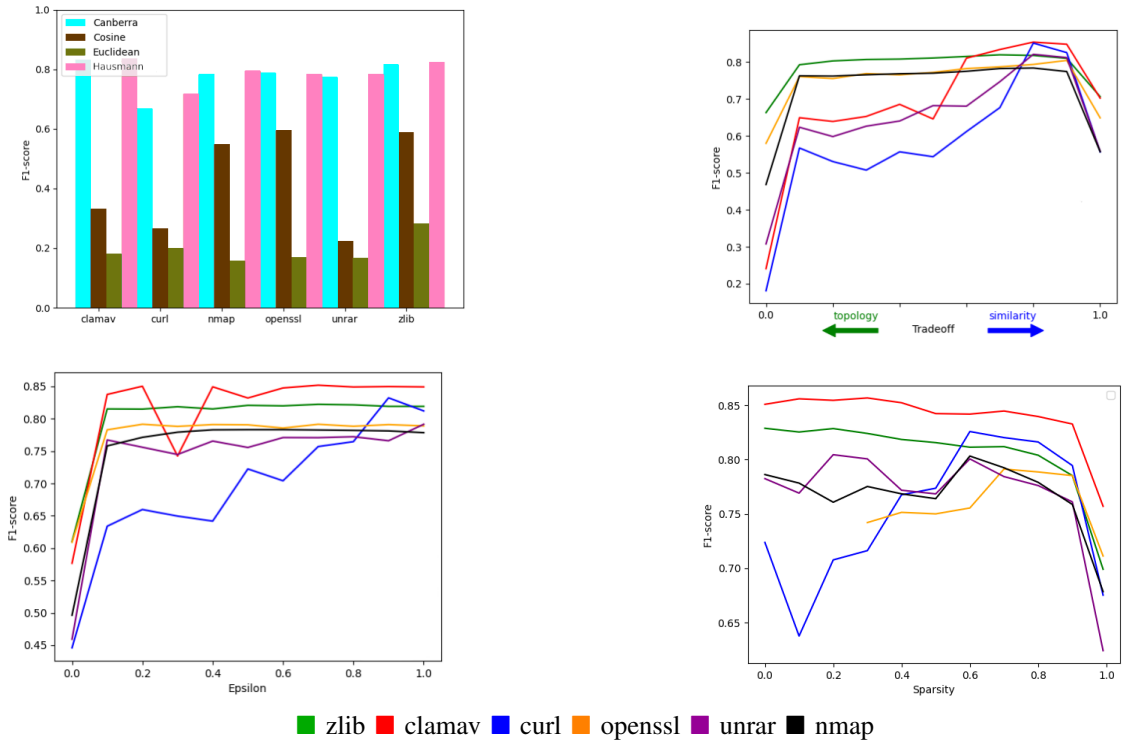


Fig. 3: Distance (d), epsilon (ϵ), tradeoff (α) and sparsity (s_{ratio}) impact on the different projects.

structural information, resulting in a lower f1-score. By increasing α, namely by according more weight on the true similarity, we improve the performances until we reach the standard QBinDiff score. This means that we can deal with a noisy adjacency by using a high tradeoff centered on similarity.

- When the similarity matrix is disturbed, increasing the tradeoff from $\alpha = 0.0$ to around $\alpha = 0.5$ helps to improve the f1-score. Then, the similarity, even noisy, is still necessary for a small α. However, according too much weight on the noisy similarity leads to a significant performance decrease.

E. Features impact

QBinDiff proposes more parameters to adjust than α. Indeed, the similarity matrix is computed by applying a weighted linear combination of distances over features vectors (step 5-6 in Algorithm 1). These features, chosen by the user, describe several aspects of a binary: data-related information, such as the feature *DatName* that indicates data references, CFG-related data like the feature *BBlockNb* which denotes the number of BB inside a function or CG-based features as the *ChildNb* feature, which indicates the number of callees of a given function inside the CG. For this experiment, we define three feature sets, each of them including features from different categories.

The f1-score results for `zlib` binaries, using these features sets and the default parameters p_s, are shown in Figure 2. We observe that with $\alpha = 0.0$, because we do not consider similarity to compute matches, no matter what the features are, the f1-scores are similar for all the features sets. When we start to consider the similarity ($\alpha > 0$), several differences appear. When $\alpha = 1.0$, since we are not considering the CG topology anymore, it becomes particularly useful to choose features extracted from the CG, that make use of that structural information.

The rest of this paper will always consider the default feature set, which includes all the available QBinDiff features at this date[8], especially data, CFG and CG features.

F. Best parameter search

As mentioned before, QBinDiff can be tuned by carefully choosing its parameters $d, \epsilon, \alpha, s_{ratio}$. Finding the best parameters would require testing every combination over the search space, which is not affordable in practice. For this reason, we decided to start with the default parameters p_s (Canberra distance, $\alpha = 0.75$, $s_{ratio} = 0.75$ and $\epsilon = 0.5$) and modify only one parameter at a time (replace the p_s Canberra distance with Haussmann distance for example), to observe the differ's behavior. We make this parameter search on the *Dataset-1A*. Respective plots for d, ϵ, α and s_{ratio} are shown in Figure 3.

From these plots, we deduce that:

- A tradeoff highly focused on the similarity, such as $\alpha = 0.8$ or $\alpha = 0.9$, is better.
- Choosing a high ϵ, such as 0.9 or 1.0, helps QBinDiff to converge faster.

[8]https://diffing.quarkslab.com/qbindiff/doc/source/api/features.html

	zlib	unrar	curl	clamav	nmap	openssl
BinExport	0.73	2.52	5.04	8.89	9.03	2.41
Sqlite	73	223	450	523	968	496
Quokka	0.71	2.19	4.21	8.37	7.93	2.10

TABLE II: Averaged exporting time (s) depending on the exporter for each project of Dataset-1A.

- Canberra or Haussmann are the best distances.
- Increasing parameter s_{ratio} does not imply a significant performance decrease. Indeed, for small projects, such as `zlib` or `unrar`, we notice a slow f1-score loss. For bigger projects, such as `curl`, it may help to converge: indeed, decimating the similarity matrix reduces the number of unlikely candidate matches the differ has to consider, resulting in clear improvement. Choosing a value too high ends up erasing the similarity matrix and matches becomes not possible anymore. We conclude that, for large projects, it may be helpful to choose a middle value (like 0.6) for a better convergence, smaller memory usage and faster computation time.

Each project has its own best parameter set denoted *ppb*: indeed, `zlib` gives the best performance with a Haussmann distance, $\epsilon = 0.7$ (even though the difference with other ϵ values strictly above 0.0 is negligible), $\alpha = 0.8$ and $s_{ratio} = 0$. Moreover, from the best parameters set for all these projects, we can obtain the best averaged parameter set, called *avb*: Haussmann distance, $\epsilon = 0.9$, $\alpha = 0.8$, $s_{ratio} = 0.6$. We finally end up with different parameters sets: *ppb*-zlib, *ppb*-unrar, *ppb*-curl, *ppb*-clamav, *ppb*-nmap, *ppb*-openssl and *avb*, based on *Dataset-1A*.

G. Computational resources

Due to its large complexity, diffing may require a lot of computational resources. We analyze both time and space complexity of diffing solutions using a dedicated server with 64GB of RAM and 16 CPU cores. First, exporting times are compared in Table II for BinExport (used by both QBinDiff and BinDiff), Quokka (used only by QBinDiff) and Diaphora's own exporter, that produces a sqlite database. It is worth mentioning that QBinDiff is not attached to a single exporter but can be used with either Quokka or BinExport. Notice that Quokka is slightly faster than BinExport, whereas the Diaphora sqlite database implies a significant computational overhead. Second, the proper diffing tends to be costly, both in terms of memory usage and computation time. Peak memory usage and computation times can be found in Table III. We especially observe the QBinDiff sparsity effect on diffing. Notice that a low s_{ratio} may be more computationally intensive than higher ratios (leading to an out-of-memory error on large binaries belonging to the `openssl` project) and give lower results due to slower convergence. Contrarily, in some cases, higher ratios help to converge faster and obtain better results, with `openssl` scores that are higher than `nmap` ones for example.

		clamav		nmap		openssl	
		Time	RAM	Time	RAM	Time	RAM
BinDiff		110	298	69	834	56	388
Diaphora3		151	29	651	30	265	30
QBinDiff	$s = 0$	672	823	8675	8339	-	-
	$s = 0.1$	689	801	8702	8329	-	-
	$s = 0.2$	684	712	7424	7167	-	-
	$s = 0.3$	639	617	6712	6171	3844	4884
	$s = 0.4$	656	523	4933	5071	3784	4022
	$s = 0.5$	541	430	4549	3902	2279	3121
	$s = 0.6$	516	347	3146	2900	1509	2301
	$s = 0.7$	489	272	2188	2009	1060	1569
	$s = 0.8$	461	204	1725	1294	1135	999
	$s = 0.9$	447	153	1205	762	528	535
	$s = 0.99$	440	134	880	673	398	408

TABLE III: Required time (s) and memory peaks (MB) needed on the three largest projects for different differs. - means the computation was stopped due to an out-of-memory error.

	ppb-zlib	*ppb*-unrar	*ppb*-curl	*ppb*-clamav	*ppb*-nmap	*ppb*-openssl
zlib	0.83	0.84	0.82	0.84	0.83	0.81
unrar	0.80	0.82	0.81	0.83	0.81	0.78
curl	0.70	0.84	0.83	0.81	0.83	0.80
clamav	0.83	0.77	0.83	0.83	0.82	0.81
nmap	-	-	0.76	-	0.76	0.73
openssl	-	-	-	-	0.75	0.74

TABLE IV: Cross-validation results

IV. DIFFER'S EMPIRICAL COMPARISON

This Section is dedicated to establishing a comparison between existing binary diffing solutions and QBinDiff. We select BinDiff [5], [7] and Diaphora [14] as they are the most widely used binary differs in reverse-engineering. We also consider GMN, a state-of-the-art binary similarity tool [17], [18], Asm2vec [3], PalmTree [16] and JTrans [30]. Our goal is to show that the QBinDiff capability of arbitrating between CG and similarity with well-chosen features can compete and even outperform standard differs.

A. Experimental setting

The experimental settings are almost identical to the ones used in Section III-B. Because the ablation study was performed on *Dataset-1A*, we consider *Dataset-1B* in this experiment. Notice again there are no binaries in common between the two datasets. *Dataset-1B* contains 770,544 functions distributed in 474 binaries, compiled with either `clang` or `gcc`, for various versions and optimization levels.

The best different parameter sets specific for each project and the best averaged parameter set were previously found in Section III-F. Ideally, we would like to re-use them in this experiment on *Dataset-1B*. However, data leakage is a usual issue when using parameters from one data set to another. However, the performed parameter search is different from hyperparameter search that can be applied on DL models, as there is no learning in the QBinDiff algorithm, which means no overfitting can occur. QBinDiff is before all an optimization algorithm relying on Belief Propagation, which is a statistical ML algorithm.

To demonstrate it, we perform a cross-validation on parameter sets: for each project p, we use its best corresponding

		BinDiff	Diaphora3	QBinDiff-ppb (BinExport)	QBinDiff-ppb (Quokka)	QBinDiff-avb (BinExport)	QBinDiff-avb (Quokka)	GMN	Asm2vec	PalmTree	JTrans
zlib	libz.so.1.2.11	0.85	0.65	0.84	**0.89**	0.82	0.88	0.71	0.19	0.67	0.69
openssl	libssl.so.3	0.81	0.64	0.85	0.85	0.83	**0.86**	0.56	0.17	0.63	0.67
	openssl	0.95	0.68	0.96	**0.98**	0.92	**0.98**	0.59	0.54	0.76	0.72
	libcrypto.so.3	0.76	0.78	0.63	0.80	0.67	**0.82**	0.58	0.01	0.55	0.46
nmap	nping	0.59	0.52	0.74	**0.77**	0.73	**0.77**	0.17	0.17	0.41	0.52
	ncat	0.73	0.58	0.86	**0.92**	0.86	**0.92**	0.24	0.17	0.56	0.67
	nmap	0.8	0.8	0.73	**0.82**	0.73	**0.82**	0.66	0.10	0.61	0.43
clamav	libclamav	0.58	0.46	0.77	**0.81**	0.76	**0.81**	0.43	0.10	0.51	0.53
curl		0.65	0.56	0.83	**0.88**	0.83	**0.88**	0.24	0.22	0.50	0.57
unrar		0.68	0.62	0.82	**0.88**	0.81	0.87	0.22	0.14	0.57	0.69
Averaged		0.74	0.63	0.80	**0.86**	0.80	**0.86**	0.44	0.18	0.58	0.60

TABLE V: f1-scores for different binaries and differs. *ppb* stands for *per project best* and *avb* for *averaged best*.

parameter set *ppb* computed on *Dataset-1A* to perform diffing on *Dataset-1B*. For example, we use *ppb*-zlib to perform diffing on other binaries from *Dataset-1B*. Cross-validation results are available in Table IV, with no significant f1-score differences, meaning there is not artificial inflated f1-score, a phenomenon that could happen with data leakage.

Diaphora results were obtained with its latest version, without any decompiler features. GMN was trained with the default hyperparameters of the available source code and graph attributes (Bag-of-Words over the assembly instruction mnemonics) [18]. Because we cannot train GMN on *Dataset-1B* as it is used for testing, we train GMN using *Dataset-1A*. Once the embeddings are obtained, we compute matches using the Hungarian Algorithm [15]. The same principle is applied to train Asm2vec [3], except the number of random walks is set to 3. PalmTree [16] and JTrans [30] are used with their default parameters. QBinDiff is tested with the two available binary exporters, BinExport [9] and Quokka [2] to see which exporter performs the best. QBinDiff is using all the available features at this date, and we test the two parameter sets found on *Dataset-1A*, as explained in Section III-F: the best parameter set per project, denoted as *ppb* and the best averaged parameter set denoted as *avb*.

B. Results

F1-score results using main standard binaries are displayed in Table V. Several aspects can be highlighted:

- Using QBinDiff with Quokka is much more efficient than using QBinDiff with BinExport, with a f1-score difference of 0.17 for `libcrypto` with *ppb*. This advocates for an increased use of Quokka. Such results can be explained by the fact Quokka exports more information (such as specific cross-references) that BinExport does not.[9]
- Diaphora exhibits lower f1-score results, compared to BinDiff and QBinDiff, because it tends to privilege precision over recall.
- Binary similarity tools show lower scores than standard differs. This can be explained by the fact that HA cannot output a correct mapping if the similarity scores, equivalent to cost, lie in the same range of values with very little standard deviation, which is the case for these trained models. However, PalmTree and JTrans almost compete with Diaphora, with their enriched representations, compared to Asm2vec or GMN.
- There is very little difference in terms of f1-score between results obtained when using *ppb* or *avb*. This means that a QBinDiff user does not have to do the hyperparameter search done in Section III-F, and can simply take the best averaged parameters found earlier to diff its own binaries. Notice that for the `libss.so.3` and `libcrypto.so.3` binaries, *ppb* configuration leads to lower results than the *avb* one. Intuitively, it should not be the case. In fact, if we had performed a complete parameter search over all the parameters $d, \alpha, \epsilon, s_{ratio}$, *ppb* should always produce better results. However, we simply start from a default parameter set, denoted, p_s and make one parameter change at the time. Consequently, *ppb* reflects the best parameter set for each parameter dimension given a default parameter set. Then, it is possible that *avb* outputs slightly better results than *ppb*.

We conclude that QBinDiff using BinExport and even better Quokka, significantly outperforms the other differs and has a f1-score of 12 points better than BinDiff.

V. Discussions

A. Limitations

Initially, we assumed that disassemblies generated by tools like IDA-Pro or Ghidra are accurate. However, these tools may encounter challenges in producing correct disassemblies, especially when faced with techniques like obfuscation or unusual compiler optimizations. Overall, all tools are affected by the limitation of relying solely on disassemblies.

Secondly, these diffing experiments only address one-to-one matching, which may not be adequate for obfuscated or optimized functions where a one-to-many approach would be more appropriate. This issue is quite intricate, and only

[9] See https://blog.quarkslab.com/quokka-a-fast-and-accurate-binary-exporter.htmlformoredetails.

a few solutions have been proposed by researchers, with the effectiveness of these solutions are yet to be definitively established [14].

Third, binaries compiled in -O1 up to -Os apply several optimizations, in particular inlining, which may impact the diffing evaluation as several functions are missing. Studying inlining impact on binary diffing is still an unexplored research area and only few research papers starts to focus it [11], [12].

B. Future work

Extending such a comparison study with cross-architecture binaries seems to be a natural step for binary diffing analysis.

Similarly, adding more binary differs such as SAFE can help to further analyze why binary similarity tools output lower scores compared to binary differs.

VI. Conclusion

In this work, we presented QBinDiff and its core algorithm. We performed an ablation study on its parameters and features and detailed how each of them influences the diffing results. Using the best average parameters, we established a comparison between standard differs and similarity-based diffing and showed that QBinDiff significantly outperforms other differs, especially when the Quokka exporter is used.

References

[1] Gianluca Capozzi, Daniele Cono D'Elia, Giuseppe Antonio Di Luna, and Leonardo Querzoni. Adversarial attacks against binary similarity systems. *arXiv preprint arXiv:2303.11143*, 2023.

[2] Alexis Challande, Robin David, and Guënaël Renault. Quokka: A fast and accurate binary exporter. In *GreHack 2022-10th International Symposium on Research in Grey-Hat Hacking*, 2022.

[3] Steven HH Ding, Benjamin CM Fung, and Philippe Charland. Asm2vec: Boosting static representation robustness for binary clone search against code obfuscation and compiler optimization. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019.

[4] Yue Duan, Xuezixiang Li, Jinghan Wang, and Heng Yin. Deepbindiff: Learning program-wide code representations for binary diffing. In *Network and distributed system security symposium*, 2020.

[5] Thomas Dullien and Rolf Rolles. Graph-based comparison of executable objects (english version). *Sstic*, 5(1):3, 2005.

[6] Frank Emmert-Streib, Matthias Dehmer, and Yongtang Shi. Fifty years of graph matching, network alignment and network comparison. *Information sciences*, 346:180–197, 2016.

[7] Halvar Flake. Structural comparison of executable objects. *DIMVA 2004, July 6-7, Dortmund, Germany*, 2004.

[8] Hao Gao, Tong Zhang, Songqiang Chen, Lina Wang, and Fajiang Yu. Fusion: Measuring binary function similarity with code-specific embedding and order-sensitive gnn. *Symmetry*, 2022.

[9] Google. Binexport. https://github.com/google/binexport, 2016. Accessed: 2023-08-21.

[10] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

[11] Ang Jia, Ming Fan, Wuxia Jin, Xi Xu, Zhaohui Zhou, Qiyi Tang, Sen Nie, Shi Wu, and Ting Liu. 1-to-1 or 1-to-n? investigating the effect of function inlining on binary similarity analysis, 2022.

[12] Ang Jia, Ming Fan, Xi Xu, Wuxia Jin, Haijun Wang, and Ting Liu. Cross-inlining binary function similarity detection, 2024.

[13] Dongkwan Kim, Eunsoo Kim, Sang Kil Cha, Sooel Son, and Yongdae Kim. Revisiting binary code similarity analysis using interpretable feature engineering and lessons learned. *IEEE Transactions on Software Engineering*, 49(4):1661–1682, 2022.

[14] Joxean Koret. Diaphora. https://github.com/joxeankoret/diaphora, 2015. Accessed: 2023-08.

[15] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[16] Xuezixiang Li, Yu Qu, and Heng Yin. Palmtree: learning an assembly language model for instruction embedding. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3236–3251, 2021.

[17] Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph matching networks for learning the similarity of graph structured objects. In *International conference on machine learning*, pages 3835–3845. PMLR, 2019.

[18] Andrea Marcelli, Mariano Graziano, Xabier Ugarte-Pedrero, Yanick Fratantonio, Mohamad Mansouri, and Davide Balzarotti. How machine learning is solving the binary function similarity problem. In *31st USENIX Security Symposium (USENIX Security 22)*, 2022.

[19] Luca Massarelli, Giuseppe A Di Luna, Fabio Petroni, Leonardo Querzoni, Roberto Baldoni, et al. Investigating graph embedding neural networks with unsupervised features extraction for binary analysis. In *Proceedings of the 2nd Workshop on Binary Analysis Research (BAR)*, pages 1–11, 2019.

[20] Xiaozhu Meng and Barton P Miller. Binary code is not easy. In *Proceedings of the 25th International Symposium on Software Testing and Analysis*, pages 24–35, 2016.

[21] Elie Mengin. *Binary Diffing as a Network Alignment Problem*. PhD thesis, Universite Paris 1-Pantheon Sorbonne, 2021.

[22] Elie Mengin and Fabrice Rossi. Binary diffing as a network alignment problem via belief propagation. In *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 967–978. IEEE, 2021.

[23] Elie Mengin and Fabrice Rossi. Improved algorithm for the network alignment problem with application to binary diffing. *Procedia Computer Science*, 192:961–970, 2021.

[24] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[25] Jiang Ming, Dongpeng Xu, and Dinghao Wu. Memoized semantics-based binary diffing with application to malware lineage inference. In Hannes Federrath and Dieter Gollmann, editors, *ICT Systems Security and Privacy Protection*, pages 416–430, Cham, 2015. Springer International Publishing.

[26] Kexin Pei, Zhou Xuan, Junfeng Yang, Suman Jana, and Baishakhi Ray. Trex: Learning execution semantics from micro-traces for binary similarity. *arXiv preprint arXiv:2012.08680*, 2020.

[27] Quarkslab. Qbindiff - github. https://diffing.quarkslab.com/qbindiff/doc/source/features.html#id1, 2023. Accessed: 2023-09-7.

[28] Paria Shirani, Lingyu Wang, and Mourad Debbabi. Binshape: Scalable and robust binary library function identification using function shape. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 301–324. Springer, 2017.

[29] Sami Ullah and Heekuck Oh. Bindiff nn: Learning distributed representation of assembly for robust binary diffing against semantic differences. *IEEE Transactions on Software Engineering*, 48(9):3442–3466, 2021.

[30] Hao Wang, Wenjie Qu, Gilad Katz, Wenyu Zhu, Zeyu Gao, Han Qiu, Jianwei Zhuge, and Chao Zhang. Jtrans: Jump-aware transformer for binary code similarity detection. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2022, page 1–13, New York, NY, USA, 2022. Association for Computing Machinery.

[31] Lei Zhao, Yuncong Zhu, Jiang Ming, Yichen Zhang, Haotian Zhang, and Heng Yin. Patchscope: Memory object centric patch diffing. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 149–165, 2020.

Graph Representation and Feature Selection using Sparse Coding and Dictionary Learning. Application to clustering in Cybersecurity.

Barbara Pilastre, Tristan Bitard-Feildel

***Abstract*—In recent year, graph signal processing have received an increasing interest in many fields and cybersecurity has not escaped to this trend as complex cybersecurity data can often be depicted as graph. This paper introduces GABAIN (Graph leArning Based on spArse codINg), a new graph representation method based on sparse coding. The method takes advantage of sparse coding to perform similarity search and infers relationships between sample data preserving the local structure of the data. Additionally, the proposed method includes a dictionary learning step with feature selection constraint via $\ell_{2,1}$ norm regularization to identify informative features and offer some explainability and data knowledge. We evaluate the proposed method for clustering application in cybersecurity use cases and demonstrate the competitiveness of the proposed method and the efficiency of its feature selection option.**

***Index Terms*—Similarity Graph Leaning, Sparse Coding, Dictionary Learning, Clustering, Explainability.**

I. Introduction

GRAPH signal processing is an active area of research for many applications such as anomaly detection, classification or pattern recognition [1], [2]. Indeed, representing data as graph can be very interesting to highlight some useful structure for processing, analysis and visualization of the data. Formally, the problem of data representation as graph is the following : transform a dataset classically represented as matrix data $\boldsymbol{Y} \in \mathbb{R}^{N \times P}$ with N observations of P variables or features into a graph composed of nodes connected by edges.

Graph representation appears straightforward in various application domains where relational data is available [3]. As example, it is quite classical to represent social media data as graph in which users are represented as nodes and edges describe relationships or interactions between them [4], [5]. In the same way in cybersecurity for intrusion detection [6], system activity data is regularly represented as graph in which system objects (*Process*, *Thread* or *Files*) are represented as nodes and edges describe actions between them (creation, reading, etc.). In other cases, the representation as graph is not obvious. To address this drawback, numerous approaches, referred as graph learning (GL), have been designed to infer relationships between features (i.e., variables of the original dataset are represented as nodes in the learned graph). They are commonly based on statistical or physical motivated models and are useful for applications such as image encoding and compression applications or brain signal analysis [7]. In contrast, clustering or outlier detection applications often infer relationships between observations by exploiting data similarity. Resulting graph, called similarity graph, is a directed and weighted graph in which each node corresponds to a sample or observation of the original dataset, and weighted edges connect similar samples. For clustering task, for instance, the similarity graph learning (SGL) strategy as data preproccessing step can be interesting since the clustering can be performed on the learned graph using efficient graph clustering or community detection methods that not require the number of cluster to be provided a priory. This paper introduces a new SGL method based on sparse coding. The proposed method, inspired by the works conducted in [8], [9], takes advantage of sparse coding to detect similar data and capture the local structure of the data.

Furthermore, explanability and data knowledge are important issues in many field, especially in cybersecurity. Indeed, cybersecurity analyst require to be provided with explanations about system alerts or events that occurred to make decisions fixing problems and preserving system performance and functionality. Thus, although automatic processing can be very helpful, especially to deal with large amounts of data, it is important to make available model explanability to understand how processing end result is reached and improve data knowledge. To this end, a good strategy is to identify informative features using feature selection techniques based on sparse coding and dictionary learning [10], [11]. Aware of this, we augmented the GABAIN method with a selection feature option using dictionary learning step via a $\ell_{2,1}$ norm regularization. Finally, the interest of the GABAIN method is twofold : the similarity graph learning to represent original matrix data as graph on one hand, and feature selection on the other.

To sum-up, the main contributions of this paper are:

- A similarity graph learning preserving the local structure of the data based on sparse coding.
- A feature selection using dictionary learning.
- An evaluation of the proposed method on clustering problems in cybersecurity applications and a comparison to the state-of-the-art approaches.

The paper is organized as follows. Similarity graph learning methods and feature selection approaches are reviewed in section 2. Section 3 details the proposed method and its optimization. Experimental results from clustering problems

Barbara Pilastre and Tristan Birtard Feildel are with Agence Ministerielle de l'Intelligence Artificielle de Defense, French Ministry of Defense (MoD), France.

in cybersecurity domain are compared to the state-of-the art in section 4. Finally, conclusion and future works are given in section 5.

II. Related Works

A. graph learning from data

Graph data representation is often very interesting since the obtained data structure can improve processing, analysis and visualization of the data. A generalized graph representation method can represent a real challenge given the variety of application domains. We distinguish two main strategies to infer graph from data : graph learning (GL) [7], which infer relationships between features, and similarity graph learning (SGL) [12], which infer relationships between data samples. This paper focus on SGL in order to investigate solutions for clustering applications. As a results of SGL, each rows of the original data matrix $\boldsymbol{Y} \in \mathbb{R}^{N \times P}$, is defined in the node set of the estimated graph. Node connections by weighted edges are given using a similarity function or distance metric. Simple SGL methods build fully connected graph. The idea is to use similarity function to compute similarity matrix and infer a fully connected graph with weighted edges. Weights correspond to the similarity measure of each pair of nodes. The Gaussian kernel is one of the most used similarity function [13] since it allows complex and non-linear relationships to be captured and local structure of the data to be preserved. As demonstrated in [14] for clustering application, the Laplacian kernel can be interesting. Popular SGL methods are based on nearest neighbors approaches. A nearest-neighbor graph (NNG) [15] is a sparse graph in which nodes are connected to its nearest neighbor, i.e., to the data sample of $\boldsymbol{Y}$ whose similarity from it is maximum among all the given points other than itself. An extension to the NNG is the k-NNG [16] in which each node is connected to its k nearest neighbors, meaning the k samples whose similarity is in the top-k highest similarity from it. In the recent years, many variants of the K-NNG have been proposed. Some of them are devoted to adjust the k value [17], [18], [19], [20] and other defend the use of other similarity function, especially the Gaussian similarity function whose the parameter σ can be better tuned [21], [22], [23]. Other well used SGL methods are sparse subspace similarity graph building methods. These methods assume that the data points close to each other can be expressed in the data space through the linear combination of them [24]. These approach based on sparse coding define an objective function to exploit self-expressiveness of the data and develop strategies to preserve local and global structure of the data [25] or to make the sparse representation more discriminative [26].

B. Feature selection

Feature selection is one of the two main dimensionality reduction strategies used to reduce the number of features in a dataset. There are multiple benefit attached to reduction dimension since it can solve problems of data sparsity in high dimension (curse of dimensionality) [27], scalability or explainability. As mentioned, dimensionality reduction methods can be divided into two categories: feature extraction (FE) and feature selection (FS) [10], [11]. FE is to find a new lower-dimensional space than the original space and project the original data in such new space. The new data are described by features of the new space estimated by linear or non-linear transformation. These techniques obtained promising results on real-world data, especially the non-linear transformation strategies[28], but they raise difficult questions about results interpretation since features of the new space, computed from a mathematical optimization problem, are no longer interpretable. Conversely, FS methods aims to identify and select relevant features in the original space and eliminate less informative ones. As original data have commonly be extracted from operational process or physical measurements, FS ensures an easier understanding and interpretation of the results. FS approaches, which are, for the main part, based on sparse coding are studied in this paper.

Traditional unsupervised FS algorithms aim to remove redundant or irrelevant features from the dataset. The selected features should preserve the overall data structure and properties and are more discriminatory with respect to a given criteria than the removed features. Most FS algorithms can be categorized into sparse coding or dictionary learning approaches. Each method formulates an objective function as an optimization problem including a $\ell_{2,1}$-norm regularization on which the FS is based. The approaches based on sparse coding select features by estimating a FS matrix which is sparse in rows. It is the case of UDF [29] which proposes a linear classifier based on sparse coding to exploit discriminatory information of the data. NDFS [30] extends UDF [29] by adding spectral clustering to exploit better these information. Regularized models are also proposed to preserve structure of the data. JELSR [31] takes advantage of spectral regression and Laplacian matrix to preserve manifold of the data. SRFS [32] preserves the local structure of features by sparse regression and the global structure among samples and among features using a low-rank constraint. LDSSL [33] adopts a twofold strategy and proposes a sparse representation as linear model that preserves both the local informative structure and local geometric structure of the data.

The methods based on dictionary learning approaches try to learn a basis matrix with feature selection and provide a new data representation along with the elimination of uninformative or redundant features. The DLUFS [34] method uses a dictionary learning approach with a low-rank constraint to obtain a sparse representation of the data and eliminate uninformative and redundant features. In CDLFS [35], dictionary learning aims to select the features that can well preserve the data distribution. According to recent results, the DLUFS [34] solution is FS state-of-the-art. Unfortunately, the actual implementation of the DLUFS method is not scalable and could not be applied to the data of these experiments that can count hundreds samples.

III. The GABAIN Algorithm

A. Notations

We summarize notations used in this paper in Table I.

TABLE I: Used notations.

λ	scalars
$\boldsymbol{v}$	vector v
$\boldsymbol{v}_i$	the ith element of $\boldsymbol{v}$
$\boldsymbol{M}$	matrix M
$\boldsymbol{M}^T$	the transpose of M
$\boldsymbol{M}^{-1}$	the inverse of M
$\boldsymbol{M}(i,j)$	the element in the ith row and jth column of M
$\|\boldsymbol{v}\|_1$	the l_1-norm of $\boldsymbol{v}$, i.e., $\|\boldsymbol{v}\|_1 = \sum_i \|\boldsymbol{v}_i\|$
$\|\boldsymbol{M}\|_{1,1}$	the $l_{1,1}$-norm of M, i.e., $\|\boldsymbol{M}\|_{1,1} = \sum_j \|\boldsymbol{M}(:j)\|_1$
$\|\boldsymbol{M}\|_{2,1}$	the $l_{2,1}$-norm of M, i.e., $\|\boldsymbol{M}\|_{2,1} = \sum_i \sqrt{\sum_j \boldsymbol{M}(i,j)^2}$
$\|\boldsymbol{M}\|_F$	the Frobenius norm of M, i.e., $\|\boldsymbol{M}\|_F = \sqrt{\sum_{i,j} \boldsymbol{M}(i,j)^2}$

B. Sparse Representation & Feature Selection

The main idea of the proposed sparse coding based graph data representation is to express each of the N data samples of the data matrix $\boldsymbol{Y} \in \mathbb{R}^{N\times P}$ as a sparse linear combination of the rest of the dataset. More precisely, we consider the transpose matrix of $\boldsymbol{Y}$, denoted as $\boldsymbol{Y}^T$ and a data-driven dictionary $\boldsymbol{D} \in \mathbb{R}^{P\times L}$ ($L < N$) corresponding to a subset of $\boldsymbol{Y}^T$ since $\boldsymbol{D}$ is composed of L atoms (i.e., columns) drawn randomly from the data $\boldsymbol{Y}^T$. This dictionary exploits the self-expressiveness of the data to represent data samples by the others. Using a sparsity constraint via a ℓ_1 norm regularization, the number of samples used in the representation is limited in order to promote similarity search. The highest coefficients of the resulting sparse representation allow pairs of similar data to be detected and edges of the graph to be identified. Furthermore, in order to identify the more relevant features, the GABAIN algorithm includes an optional dictionary learning step. The idea is to divide the dictionary $\boldsymbol{D}$ into two dictionaries, $\boldsymbol{D}_1 \in \mathbb{R}^{P\times L}$ and $\boldsymbol{D}_2 \in \mathbb{R}^{P\times L}$, sparse in rows such as $\boldsymbol{D} = \boldsymbol{D}_1 + \boldsymbol{D}_2$. This dictionary step aims to identify and distinguish informative features which will compose dictionary $\boldsymbol{D}_1$ from uninformative features which will compose dictionary $\boldsymbol{D}_2$. This dictionary learning step is performed via a suitable $\ell_{2,1}$ norm regularization. Finally, The proposed strategy decomposed the data as follows:

$$\boldsymbol{Y}^T = \boldsymbol{D}_1\boldsymbol{X}_1 + \boldsymbol{D}_2\boldsymbol{X}_2 + \boldsymbol{B} \tag{1}$$

where $\boldsymbol{X}_1 \in \mathbb{R}^{L\times N}$ is the sparse coefficient matrix corresponding to the representation of the data matrix $\boldsymbol{Y}$ in the space of the dictionary $\boldsymbol{D}_1$, $\boldsymbol{X}_2 \in \mathbb{R}^{L\times N}$ is the coefficient matrix corresponding to the representation of the data matrix $\boldsymbol{Y}$ in the space of the dictionary $\boldsymbol{D}_2$ and $\boldsymbol{B} \in \mathbb{R}^{P\times N}$ is an additive noise.

Thus, the GABAIN algorithm is to solve the following problem:

$$\begin{aligned}\widehat{\boldsymbol{X}_1}, \widehat{\boldsymbol{X}_2}, \widehat{\boldsymbol{D}_1}, \widehat{\boldsymbol{D}_2} = \arg\min_{\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{D}_1,\boldsymbol{D}_2} &\|\boldsymbol{Y}^T - \boldsymbol{D}_1\boldsymbol{X}_1 - \boldsymbol{D}_2\boldsymbol{X}_2\|_F^2 \\ &+ \lambda\|\boldsymbol{X}_1\|_{1,1} + \beta\|\boldsymbol{D}_2\boldsymbol{X}_2\|_{2,1} \\ \text{s.t. } &\boldsymbol{D} = \boldsymbol{D}_1 + \boldsymbol{D}_2.\end{aligned} \tag{2}$$

where λ and β are regularization parameters that control respectively the level of sparsity of $\boldsymbol{X}_1$ and $\boldsymbol{D}_2\boldsymbol{X}_2$. The greater the value of λ the less the number of non-zero value in $\boldsymbol{X}_1$ and the greater the value of β the less the number of non-zero rows in $\boldsymbol{D}_2\boldsymbol{X}_2$ (i.e., the more the number of selected features).

This strategy is inspired by similar successful works conducted on anomaly detection [8], [9]. Note that the model integrates two constraints via ℓ_1 and $\ell_{2,1}$ norm regularizations : one for each objective. The ℓ_1 regularization limits the number of dictionary atoms (i.e., the number of samples). The selected samples in the sparse representation can be used for the graph construction by creating edges connecting the similar samples. The $\ell_{2,1}$ norm regularization limits the number of available rows in $\boldsymbol{D}_2$, i.e., the number of features of the dataset that can be used to estimate $\boldsymbol{D}_2\boldsymbol{X}_2$ which can be seen as the residual of the sparse representation. Through this game of constraints, the features distribution between the two dictionaries $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ is mechanically conducted. Indeed, since the ℓ_1 norm constraints the $\boldsymbol{X}_1$ estimation and no constraint affect the estimation of $\boldsymbol{X}_2$, the informative features will be automatically selected in $\boldsymbol{D}_1$ whereas uninformative features will be selected in $\boldsymbol{D}_2$. Details about problem 2 iterative optimization and update equations of the different variables at the kth iteration are provided in the next section.

C. Optimization

By adding the variable $\boldsymbol{E}$, the problem (2) can reformulated as follow :

$$\begin{aligned}\widehat{\boldsymbol{X}_1}, \widehat{\boldsymbol{X}_2}, \widehat{\boldsymbol{D}_1}, \widehat{\boldsymbol{D}_2}, \widehat{\boldsymbol{E}} = \arg\min_{\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{D}_1,\boldsymbol{D}_2} &\|\boldsymbol{Y}^T - \boldsymbol{D}_1\boldsymbol{X}_1 - \boldsymbol{E}\|_F^2 \\ &+ \lambda\|\boldsymbol{X}_1\|_{1,1} + \beta\|\boldsymbol{E}\|_{2,1} \\ \text{s.t. } &\boldsymbol{D} = \boldsymbol{D}_1 + \boldsymbol{D}_2, \;\; \boldsymbol{E} = \boldsymbol{D}_2\boldsymbol{X}_2.\end{aligned} \tag{3}$$

and corresponds to an extension of [36]. the problem (3) can be solved by the Alternative Direction Method of Multipliers (ADMM) [37] by adding the auxiliary variable $\boldsymbol{Z}$:

$$\begin{aligned}\widehat{\boldsymbol{X}_1}, \widehat{\boldsymbol{X}_2}, \widehat{\boldsymbol{D}_1}, \widehat{\boldsymbol{D}_2}, \widehat{\boldsymbol{E}}, \widehat{\boldsymbol{Z}} = \arg\min_{\boldsymbol{X}_1,\boldsymbol{X}_2,\boldsymbol{D}_1,\boldsymbol{D}_2} &\|\boldsymbol{Y}^T - \boldsymbol{D}_1\boldsymbol{X}_1 - \boldsymbol{E}\|_F^2 \\ &+ \lambda\|\boldsymbol{Z}\|_{1,1} + \beta\|\boldsymbol{E}\|_{2,1} \\ \text{s.t. } &\boldsymbol{D} = \boldsymbol{D}_1 + \boldsymbol{D}_2, \;\; \boldsymbol{E} = \boldsymbol{D}_2\boldsymbol{X}_2, \;\; \boldsymbol{Z} = \boldsymbol{X}_1.\end{aligned} \tag{4}$$

(5)

and the constraint $\boldsymbol{Z} = \boldsymbol{X}_1$. Note that, contrary to Problem (3), the first and second terms of (4) are decoupled, which allows an easier estimation of the matrix $\boldsymbol{X}_1$. However, this problem is not convex in all variables to estimate, but it is convex in each variable by fixing the others. It can be solved using the ADMM algorithm by minimizing the following augmented Lagrangian:

$$\begin{aligned}\mathcal{L}_A(\boldsymbol{X}_1, \boldsymbol{E}, \boldsymbol{Z}, \boldsymbol{M}, \mu) = &\frac{1}{2}\|\boldsymbol{Y}^T - \boldsymbol{D}_1\boldsymbol{X}_1 - \boldsymbol{E}\|_F^2 \\ &+ \lambda\|\boldsymbol{Z}\|_{1,1} + \beta\|\boldsymbol{E}\|_{2,1} + \boldsymbol{M}(\boldsymbol{Z} - \boldsymbol{X}_1) + \frac{\mu}{2}\|\boldsymbol{Z} - \boldsymbol{X}_1\|_F^2\end{aligned} \tag{6}$$

The update of each variable at the kth iteration is detailed below.

Updating of $\boldsymbol{X}_1$. $\boldsymbol{X}_1$ is update by minimizing the following problem:

$$\boldsymbol{X}_1^{k+1} = \arg\min_{\boldsymbol{X}_1} \|\boldsymbol{Y}^T - \boldsymbol{D}_1^k\boldsymbol{X}_1 - \boldsymbol{E}^k\|_F^2 + \boldsymbol{M}^k(\boldsymbol{Z}^k - \boldsymbol{X}_1) + \frac{\mu^k}{2}\|\boldsymbol{Z}^k - \boldsymbol{X}_1\|_F^2 \tag{7}$$

Simple algebra leads to:

$$\boldsymbol{X}_1^{k+1} = ({\boldsymbol{D}_1^k}^T\boldsymbol{D}_1^k + \mu^k I)^{-1}({\boldsymbol{D}_1^k}^T\boldsymbol{R}^k + \boldsymbol{M}^k + \mu^k\boldsymbol{Z}^k) \tag{8}$$

where $\boldsymbol{R}^k = \boldsymbol{Y} - \boldsymbol{E}^k$.

Updating of $\boldsymbol{Z}$. The update equation of $\boldsymbol{Z}$ is obtained from:

$$\boldsymbol{Z}^{k+1} = \arg\min_{\boldsymbol{Z}} \lambda\|\boldsymbol{Z}\|_{1,1} + \boldsymbol{M}^k(\boldsymbol{Z} - \boldsymbol{X}_1^{k+1}) + \frac{\mu^k}{2}\|\boldsymbol{Z} - \boldsymbol{X}_1^{k+1}\|_F^2 \tag{9}$$

Which can be simplified to:

$$\boldsymbol{Z}^{k+1} = \arg\min_{\boldsymbol{Z}} \frac{1}{2}\|\boldsymbol{P} - \boldsymbol{Z}\|_F^2 + \gamma\|\boldsymbol{Z}\|_{1,1} \tag{10}$$

where $\boldsymbol{P} = \boldsymbol{X}_1^{k+1} - \frac{1}{\mu^k}\boldsymbol{M}^k$ and $\gamma = \frac{\lambda}{\mu^k}$. The solution to this problem is given by the soft-threshodling operator:

$$\boldsymbol{Z}^{k+1} = \mathcal{S}_\lambda(\boldsymbol{P}) \tag{11}$$

$$\text{with } \mathcal{S}_\lambda(a) = \begin{cases} a-\lambda & if \ \ a > \lambda \\ 0 & if \ \ |a| \leq \lambda \\ a+\lambda & if \ \ a < \lambda \end{cases}$$

Updating of $\boldsymbol{E}$. $\boldsymbol{E}$ is updated by solving the following problem:

$$\boldsymbol{E}^{k+1} = \arg\min_{\boldsymbol{E}} \frac{1}{2}\|\boldsymbol{Y}^T - \boldsymbol{D}_1^k\boldsymbol{X}_1^{k+1} - \boldsymbol{E}\|_F^2 + \beta\|\boldsymbol{E}\|_{2,1} \tag{12}$$

which results in:

$$\boldsymbol{E}^{k+1}(i,:) = \begin{cases} \frac{\|\boldsymbol{Q}(i,:)\|_2 - \beta}{\|\boldsymbol{Q}(i,:)\|_2} & if \ \ \|\boldsymbol{Q}(i,:)\|_2 > \beta \\ 0 & otherwise \end{cases} \tag{13}$$

where $\boldsymbol{Q} = \boldsymbol{Y}^T - \boldsymbol{D}_1^k\boldsymbol{X}_1^{k+1}$ and $\boldsymbol{Q}(i,:)$ is the ith row of $\boldsymbol{Q}$.

Simple algebra allows the variables $\boldsymbol{X}_2, \boldsymbol{D}_2$ to be inferred from $\boldsymbol{X}_1, \boldsymbol{Z}$ and $\boldsymbol{E}$ to respect constraints (5).

Updating of $\boldsymbol{D}_2$. The update of $\boldsymbol{D}_2$, using the updated matrix $\boldsymbol{E}$, is:

$$\boldsymbol{D}_2^{k+1}(i,j) = \boldsymbol{D}_E^{k+1} = \begin{cases} \boldsymbol{D}(i,j) & if \ \boldsymbol{E}^{k+1}(i,j) > 0 \\ 0 & otherwise \end{cases} \tag{14}$$

Updating of $\boldsymbol{D}_1$. According to the constraint $\boldsymbol{D} = \boldsymbol{D}_1 + \boldsymbol{D}_2$, the update of $\boldsymbol{D}_2$ is given by:

$$\boldsymbol{D}_1^{k+1} = \boldsymbol{D} - \boldsymbol{D}_2^{k+1} \tag{15}$$

Updating of $\boldsymbol{X}_2$. According to the constraint $\boldsymbol{E} = \boldsymbol{D}_2\boldsymbol{X}_2$, the update of $\boldsymbol{X}_2$ is given by:

$$\boldsymbol{X}_2^{k+1} = \boldsymbol{D}_2^{k+1}[\boldsymbol{E}^{k+1}]^{-1} \tag{16}$$

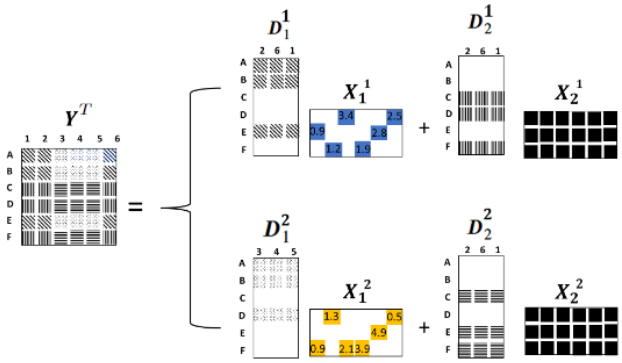
The main stages of the ADMM-based algorithm are described above to solve (4)(5) and summarized in Algorithm 1. Theoretical convergence properties of the ADMM algorithm are details in [38].

Algorithm 1 GABAIN ADDM-Based algorithm

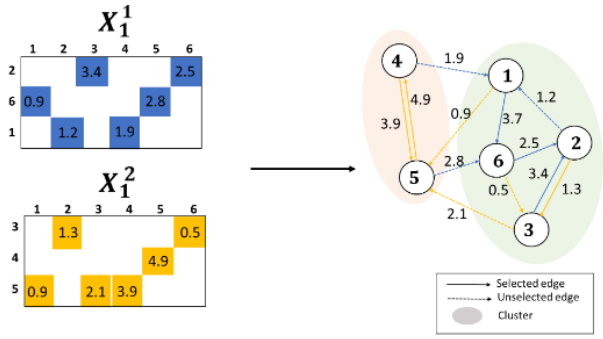
Input: the transpose of the matrix of data $\boldsymbol{Y}^T \in \mathbb{R}^{P\times N}$, the dictionary $\boldsymbol{D} \in \mathbb{R}^{P\times L}$, the scalars β and ϵ, $\rho = 1.1$.
Initialization: $k = 0$, $\boldsymbol{Z}^0$, $\boldsymbol{E}^0$, $\boldsymbol{M}^0$, $\boldsymbol{D}_1^0$, $\boldsymbol{D}_2^0$, $\boldsymbol{X}_2^0$, μ^0.
Repeat:
1) $\boldsymbol{X}_1^{k+1} = \arg\min_{\boldsymbol{X}_1} \mathcal{L}_A(\boldsymbol{X}_1, \boldsymbol{Z}^k, \boldsymbol{E}^k, \boldsymbol{M}^k, \mu^k)$
2) $\boldsymbol{Z}^{k+1} = \arg\min_{\boldsymbol{Z}} \mathcal{L}_A(\boldsymbol{X}_1^{k+1}, \boldsymbol{Z}, \boldsymbol{E}^k, \boldsymbol{M}^k, \mu^k)$
3) $\boldsymbol{E}^{k+1} = \arg\min_{\boldsymbol{E}} \mathcal{L}_A(\boldsymbol{X}_1^{k+1}, \boldsymbol{Z}^{k+1}, \boldsymbol{E}, \boldsymbol{M}^k, \mu^k)$
4) $\boldsymbol{M}^{k+1} = \boldsymbol{M}^k + \mu^k(\boldsymbol{Z}^{k+1} - \boldsymbol{X}_1^{k+1})$
5) $\boldsymbol{\mu}^{k+1} = \rho\mu^k$
6) $\boldsymbol{D}_2^{k+1} = \boldsymbol{D}_E^{k+1}$
7) $\boldsymbol{D}_1^{k+1} = \boldsymbol{D} - \boldsymbol{D}_2^{k+1}$
8) $k = k + 1$
Until: $\frac{\|\boldsymbol{Z}^k - \boldsymbol{X}_1^k\|_F^2}{\|\boldsymbol{X}_1^k\|_F^2} < \epsilon$
Output: $\boldsymbol{X}_1^k$

D. Illustrative example

This section details the GABAIN algorithm for clustering application through a simple example shown in Figure 2. As a reminder, The GABAIN algorithm aims to represent any data matrix as graph data preserving local structure of the data. In addition it includes a FS option to offer data knowledge and explainability. In the example of Figure 2, we consider a the transpose of a data matrix noted $\boldsymbol{Y}^T \in \mathbb{R}^{6\times 6}$. The first step of the GABAIN algorithm is to draw dictionaries by splitting the matrix $\boldsymbol{Y}^T$ into smaller dictionaries. In figure 2, two dictionaries are composed of three atoms each : samples 2, 6 and 1 compose columns of the first dictionary $\boldsymbol{D}_1^1$, and samples 3, 4, 5 compose columns of the second dictionary $\boldsymbol{D}_1^2$. The second step of the GABAIN algorithm is to apply the ADMM-based algorithm to each dictionary. In our example of figure 2, results reveal that features A,B and E have been selected when the ADMM-based algorithm have been performed on the dictionary $\boldsymbol{D}_1^1$, and features A,B and D have been selected when the ADMM-based algorithm has been performed on the dictionary $\boldsymbol{D}_1^2$. In the third step, the graph connections or edges are deducted from the estimation of $\boldsymbol{X}_1^1$ and $\boldsymbol{X}_1^2$ obtained in step 2. Indeed, we assume that the highest values



(a) GABAIN Step 1-2



(b) GABAIN Step 3-5

Fig. 2: GABAIN Algorithm

of sparse representation of each sample (i.e., highest values of each column of $\boldsymbol{X}_1^1$ and $\boldsymbol{X}_1^2$) highlight similarities which are extracted to connect corresponding nodes by weighted edges and build the graph as illustrated in figure 2. Note that self-loop, i.e., edges that connect a node to itself, are not included in the graph since they are not informative. Note that for each dictionary, N edges are identified, one per node. The dictionary split allows more edges to be identified. Furthermore, since the ADMM-based implies matrix inversion, the use of smaller dictionaries allows an easier estimation. The number of edges per node to select is an hyperparameter fix by user. In our example, two dictionaries are drawn but only one edge per node is selected : edges with the highest weight. In addition, to promote stochasticity, the GABAIN algorithm can be apply several times, denoted as epoch. Thus, more dictionaries are drawn and more interesting similarities can be identified.

IV. Experiments

A. Datasets

In these experiments, two cybersecurity use cases are studied. The first one is about facial expression recognition in the images of the popular Yale face database. The second use case is about android malware categorization. The corresponding dataset, called APK, has been built from open-source VirusTotal analysis reports of hundreds Android applications. The main specifications of the two datasets are reported in Table II.

Algorithm 2 GABAIN Algorithm for clustering applications

Input: the transpose of the matrix data $\boldsymbol{Y}^T \in \mathbb{R}^{P\times N}$, the scalars β, number of edges C and size of dictionaries L

1) Draw dictionaries by splitting $\boldsymbol{Y}^T$ into $\frac{N}{L}$ random dictionaries composed of L atoms each.
2) Repeat Algorithm 1 for each dictionaries and construct graphs.
3) Merge graphs returned by ADMM executions.
4) Select C edges per nodes.
5) Apply graph clustering algorithm.

Output: Labels of clusters, selected features

TABLE II: Characteristics of datasets.

Dataset	Samples	Features	Classes	Category
Yale	165	1024	15	Image
APK	934	47	5	Android Application

B. Compared methods

The GABAIN algorithm has been evaluated for clustering in two cybersecurity use-cases. Combined to Louvain community detection method [39], the GABAIN approach has been compared to three state-of-the-art similarity graph learning methods : k-NNG, RBF similarity graph, Laplacian similarity graph. In the same way as for GABAIN, this similarity graph learning have been combined to Louvain community detection to perform clustering. In addition, popular clustering methods have been applied, such as the K-means algorithm since its the most used clustering approach, and two clustering techniques which have the benefit of not requiring the number of clusters provided a priory, namely HDBSCAN [40] and OPTICS [41]. As mentioned before, the DLUFS [34] feature selection state-of-the-art solution is not evaluated in this work since the actual implementation is not scalable. The different methods are all described in the following:

- **K-NNG** : Graph similarity method which connect each node to its k nearest neighbors according to the euclidean distance.
- **AKNNG**[20] : A variant of K-NNG method that build the adaptive k-nearest neighbors similarity graph (AKNNG) by automatically adjusting k for different data points.
- **MAKNNG**[20] : A robust version of the AKNNG solution.
- **RBF SGL** : Graph similarity learning based on RBF similarity matrix.
- **Laplacian SGL** Graph similarity learning based on Laplacian similarity matrix.
- **K-means**: Centroid-based clustering method that minimizes the average distance between data samples within clusters.
- **HDBSCAN**: Hierarchical density-based clustering that finds core samples of high density and expands clusters from them.
- **OPTICS**: Density based clustering method that keeps cluster hierarchy for a variable neighborhood radius.

Methods	Yale	APK
K-means	0.90 ± 0.01	**0.90 ± 0.01**
HDBSCAN	0.28 ± 0.00	0.84 ± 0.00
OPTICS	0.64 ± 0.00	0.84 ± 0.00
k-NNG + Louvain	0.81 ± 0.01	0.88 ± 0.00
RBF-SGL + Louvain	0.85 ± 0.01	**0.90 ± 0.00**
Laplacian-SGL + Louvain	0.83 ± 0.00	0.86 ± 0.01
AKNNG + Louvain	0.87 ± 0.00	0.22 ± 0.00
MAKNNG + Louvain	0.90 ± 0.00	0.22 ± 0.00
GABAIN + Louvain	**0.93 ± 0.00**	**0.90 ± 0.00**

TABLE III: Clustering Results (Adjusted Rand Score ± std)

Methods	Yale	APK
K-means	0.49 ± 0.2	**0.80 ± 0.01**
HDBSCAN	0.12 ± 0.00	0.62 ± 0.00
OPTICS	0.45 ± 0.00	0.61 ± 0.00
k-NNG + Louvain	0.36 ± 0.02	0.75 ± 0.03
RBF-SGL + Louvain	0.46 ± 0.04	0.79 ± 0.00
Laplacian-SGL + Louvain	0.43 ± 0.02	0.74 ± 0.01
AKNNG + Louvain	0.48 ± 0.01	0.00 ± 0.00
MAKNNG + Louvain	0.57 ± 0.01	0.00 ± 0.00
GABAIN + Louvain	**0.58 ± 0.02**	0.75 ± 0.00

TABLE IV: Clustering Results (NMI ± std)

C. Evaluation measures

Two common measures used to evaluate clustering methods are the Adjusted Rand Score (ARS) and the Normalized Mutual Information (NMI), which can be computed by considering the clustering results, noted $\widehat{\boldsymbol{y}}$, and the ground-truth, noted $\boldsymbol{y}$. These two metrics are particularly useful since label correspondence between the clustering prediction and the ground-truth is not required.

The Adjusted Rand Score (ARS) is a corrected version of the rand index [42] which measures degree of overlapping between two partitions. It was introduced to determine whether two clustering results are similar to each other. The ARS can be defined as follow:

$$ARS = \frac{RI - ExpectedRI}{max(RI) - ExpectedRI}$$

In the formula, the RI stands for the Rand Index. The ARS is equal to 0 when samples are assigned into different clusters and it equals to 1 when the two clusters results are the same.

The second evaluation metric is the NMI which is defined as:

$$NMI(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = \frac{2I(\boldsymbol{y}, \widehat{\boldsymbol{y}})}{H(\boldsymbol{y}) + H(\widehat{\boldsymbol{y}})}$$

where H(.) is the entropy measure and $I(\boldsymbol{y}, \widehat{\boldsymbol{y}})$ [43] is the mutual information of $\boldsymbol{y}$ and $\widehat{\boldsymbol{y}}$. The more the NMI is close to 1, the better the clustering is.

D. Experimental setting

This section gives details about methods implementation and use. Graph similarity based on k-NNG or GABAIN have 20 or fewer edges per nodes since it represent a good setting for these methods applied on these two datasets. The HDBSCAN and OPTICS clustering algorithms have been applied with default values. Regarding the K-means algorithm, the number k of clusters has been set to the true number of classes given by the ground-truth (see in Table II). Finally, the GABAIN algorithm have been applied with the same value of its hyperparameters for the two datasets : $\beta = 12$, $L = \frac{N}{2}$, $C = 20$, $\alpha = 0.3$. The NMI and ARS measures are computed on 30 times running of each algorithms in two descriptive statistical quantities, mean and standard deviation (std).

E. Experiment results

This section evaluates the proposed solution applied for clustering in cybersecurity domain and compares it to state-of-the-art methods. Comparison results based on ARS and NMI are respectively reported in Table III and Table IV. The best results are marked by bold numbers and second-best results are marked by underlined numbers. Quantitative results demonstrate the interest of the SGL strategy combined with graph clustering, especially the GABAIN solution. Indeed, SGL approaches combined to Louvain clustering outperforms the HDBSCAN and OPTICS clustering and are competitive with the famous K-means clustering bu returning the best or second-best clustering for the two use-cases. In addition, note that the K-means method depends on the number of clusters provided a priory which is not the case of the Louvain method. Thus, since the SGL allows the use of efficient graph clustering methods, it seems more suitable in many real-world applications where the number of cluster is unknown. Beyond its great clustering performances, the GABAIN algorithm includes an efficient feature selection option which can represent an important advantage compared to the state-of-the-art. The obtained feature selection on the Yale dataset is displayed in figure 3. In this use-case about face expression recognition, the GABAIN method selects features (red pixels) corresponding to relevant face areas such as forehead face or cheekbones on which some specific wrinkles allow emotions to be detected. Regarding the APK dataset, the feature selection is not so visual but is also significant. Indeed, the GABAIN method applied on the APK dataset returns 22 features identified as informative, which is less than the half of the number of features in this dataset. Among the selected features the ones

(a) YALE

Fig. 3: Example of GABAIN feature selection. Yale sample image with selected features marked by red pixels.

relative to the permissions required by the android application can be found. Note that unexpected permissions required by android applications often represent an evidence of the presence of malware. Conversely, features such as the number of called libraries or the number of files in the application code source were not selected. This feature selection appeared as logical results to cybersecurity experts, confirming the efficiency of the GABAIN feature selection.

V. Conclusion

This paper investigates a new similarity graph learning method based on sparse coding. The proposed GABAIN algorithm identifies similarities between sample data and represent any matrix of data as a graph preserving the local structure of the data. In addition, the GABAIN method includes a selection feature option based on dictionary learning that can offer some explainability and data knowledge. The approach was combined to the Louvain community detection and evaluated for clustering in cybersecurity applications. The experiments results demonstrate the interest of the GABAIN strategy combined with community detection for clustering task, especially if limited knowledge on the data is available and the number of cluster is unknown. Then, the results showed the efficiency of its feature selection strategy which returns relevant features and improves user introspective insight into its data and facilitate clustering explainability. For future work, the interest of the GABAIN strategy for other application such as anomaly detection deserves to be investigated. Then, it could be interesting to quantify features importance of the selected features.

References

[1] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, P. Vandergheynst, The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains, IEEE Signal Process. Mag. 30 (3) (2013) 83–98.

[2] A. Sandryhaila, J. M. F. Moura, Discrete signal processing on graphs, IEEE Trans. Signal Process 61 (7) (2013) 1644–1656.

[3] A. Ortega, P. Frossard, J. K. J. M. F. Moura, P. Vandergheynst, Graph signal processing: Overview, challenges, and applications, Proceedings of the IEEE 106 (5) (2018) 808–828.

[4] I. Pitas, Graph-Based Social Media Analysis, Chapman and Hall/CRC, Minneapolis, Minnesota, U.S.A, 2016.

[5] M. Newman, Networks: An Introduction., Oxford Univ. Press, Oxford, U.K., 2010.

[6] F. Yang, J. Xu, C. Xiong, Z. Li, K. Zhang, Prographer: An anomaly detection system based on provenance graph embedding, in: Proc. USENIX'23, 2023.

[7] X. Dong, D. Thanou, M. Rabbat, P. Frossard, Learning graphs from data: A signal representation perspective, IEEE Signal Process. Mag. 36 (3) (2019) 44–63.

[8] A. Adler, M. Elad, Y. Hel-Or, E. Rivlin, Sparse coding with anomaly detection, in: IEEE International Workshop on Machine Learning for Signal Processing (MLSP'13), 2013.

[9] B. Pilastre, J. Y. Tourneret, S. D-Escrivan, L. Boussouf, Anomaly detection in mixed telemetry data using a sparse representation and dictionary learning, Signal Processing 168 (2020) 107320.

[10] P. Dhal, C. Azad, A comprehensive survey on feature selection in the various fields of machine learning, Applied Intelligence 52 (4) (2022) 4543–4581.

[11] J. Miao, L. Niu, A survey on feature selection, Procedia Computer Science 91 (2016) 919–926.

[12] U. V. Luxburg, A tutorial on spectral clusterin, Stat. Comput. 17 (2007) 395–416.

[13] Y. N. . M. S. Yang, Powered gaussian kernel spectral clustering, Neural Computing and Applications 31 (2019) 557–572.

[14] Z. Kang, C. Peng, Q. Cheng, Kernel-driven similarity learning, Neurocomputing 267 (2017) 210–219.

[15] F. P. Preparata, M. I. Shamos, Computational Geometry - An introduction, Springer-Verlag, New-York. USA, 1985.

[16] D. Eppstein, M. S. Paterson, F. F. Yao, On nearest-neighbor graphs, Discrete Computational Geometry 17 (1997) 263–282.

[17] X. Ye, T. Sakurai, pectral clustering using robust similarity measure based on closeness of shared nearest neighbors, in: Proc. International joint conference on neural networks (IJCNN'2015), 2015.

[18] M. Alshammari, J. Stavrakakis, M. Takatsuka, Refining a k-nearest neighbor graph for a computationally efficient spectral clustering, Pattern Recogn. 114 (23) (2021) 107869.

[19] K. Sharma, A. Seal, Spectral embedded generalized mean based k-nearest neighbors clustering with s-distance, Expert Syst. Appl. 169 (2021) 557–572.

[20] Y. Cai, J. Z. Huang, J. Yin, A new method to build the adaptive k-nearest neighbors similarity graph matrix for spectral clustering, Neurocomputing. 493 (2022) 191–209.

[21] L. Zelnik-Manor, P. Perona, Self-tuning spectral clustering, in: In Proc. Neural Information Processing Systems 17 (NIPS'2004), 2014.

[22] P. Favati, G. Lotti, O. Menchi, F. Romani, Construction of the similarity matrix for the spectral clustering method: Numerical experiments, J. Comput. Appl. Math (375) (2020) 112795.

[23] Y. Nataliani, M.-S. Yang, Powered gaussian kernel spectral clustering, Neural Comput. Appl. 31 (1) (2020) 557–572.

[24] E. Elhamifar, R. Vidal, sparse subspace clustering: Algorithm, theory, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2765–2781.

[25] X. Zhu, S. Zhang, R. Hu, Y. Zhu, J. Song, Local and global structure preservation for robust unsupervised spectral feature selection, IEEE Trans. Knowl. Data Eng. 3 (30) (2017) 517–529.

[26] J. Xu, M. Yu, L. Shao, W. Zuo, D. Meng, L. Zhang, D. Zhang, Scaled simplex representation for subspace clustering, IEEE Trans. Cybern. 51 (3) (2021) 1493–1505.

[27] D. Dohono, High-dimensional data analysis: The curses and blessings of dimensionality, AMS Math Challenges Lecture (Aug. 2000).

[28] P. D. V. Bhasha, Dimension reduction techniques: Current status and perspectives, Materials Today: Proceedings 62 (13) (2022) 7024–7027.

[29] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, X. Zhou, L2,1-norm regularized discriminative feature selection for unsupervised learning, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI'11), 2011.

[30] Z. Li, Y. Yang, J. Liu, X. Zhou, H. Lu, Unsupervised feature selection using nonnegative spectral analysis, in: Proc. AAAI Conference on Artificial Intelligence, 2012.

[31] C. Hou, F. Nie, X. Li, D. Yi, Y. Wu, Joint embedding learning and sparse regression: A framework for unsupervised feature selection, IEEE Trans. Cybern. 44 (6) (2014) 793–804.

[32] X. Zhu, S. Zhang, R. Hu, Y. Zhu, J. Song, Local and global structure preservation for robust unsupervised spectral feature selection, IEEE Trans. Knowl. Data. Eng. 30 (3) (2018) 517–529.

[33] R. Shang, Y. Meng, W. Wang, F. Shang, L. Jiao, Local discriminative based sparse subspace learning for feature selection, Pattern Recognition 92 (2019) 219–230.

[34] M. G. Parsa, H. Zare, M. Ghatee, Low-rank dictionary learning for unsupervised feature selection, Expert Systems with Applications 202 (Sept. 2022).

[35] Z. Pengfei, H. Qinghua, Z. Changqing, Z. Wangmeng, Coupled dictionary learning for unsupervised feature selection, in: Proc. AAAI Conference on Artificial Intelligence, 2016.

[36] A. Adler, M. Elad, Y. Hel-Or, E. Rivlin, Sparse coding with anomaly detection, J. Signal Process. Syst. 79 (2) (2015) 179–188.

[37] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via alternating direction method of multipliers, Foundations and Trends in Machine Learning 3 (1) (2010) 1–222.

[38] J. Eckstein, D. Bertsekas, On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators, Mathematical programming (Series A and B) 55 (3) (1992) 293–318.

[39] V. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, Journal of Statistical Mechanics Theory and Experiment 2008 (Apr. 2008).

[40] R. Campello, D. Moulavi, J. Sander, Density-based clustering based on hierarchical density estimates 7819 (2013) 160–172.

[41] M. Ankerst, M. M. Breunig, H.-P. Kriegel, S. Jörg, Optics: Ordering points to identify the clustering structure, in: In Proc. ACM SIGMOD International Conference on Management of Data, 1999.
[42] L. J. Hubert, P. Arabie, Comparing partitions, Pattern Recognition 2 (2-3) (1985) 193–218.
[43] N. X. Vinh, J. Epps, J. Bailey, Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance, Journal of Machine Learning Research 11 (95) (2010) 2837–2854.

Transformer-Based State Estimation for Multi-Target Tracking: Sensitivity Analysis against Varying Kinematic Parameters and Clutter Density

Valentin Sonntag[1,2], Jean-Marc Le Caillec[2], Alain Peres[1] and Stéphane Devaud[1]
[1] Thales Land and Air Systems, 3 Avenue Charles Lindbergh, 94150 Rungis, France
[2] Lab-STICC UMR CNRS 6285, IMT Atlantique, 655 Avenue du Technopôle, 29280 Plouzané, France
Email: {valentin.sonntag, alain.peres, stephane.devaud}@thalesgroup.com, jm.lecaillec@imt-atlantique.fr

***Abstract*—An exploration of a Transformer's behavior is proposed in the context of multi-target tracking. We investigate the behavior properties of the state predictions made by the Transformer-based method through sensitivity analysis. The experiments focus on varying kinematic parameters, clutter density, and number of objects. The Transformer-based method demonstrates consistent accuracy on the training domain and generalization capability relative to clutter density. The results show that the Transformer Tracker outperforms the Kalman Filter and Extended Kalman Filter in terms of overall accuracy. Moreover, it is capable of adapting to the training data, building data knowledge to improve prediction accuracy. The experiments highlight insufficient generalization on the number of objects. They also provide preliminary insight into the system's explainability. Finally, we discuss potential limitations and identify future research directions.**

***Index Terms*—State Estimation, Multi-target Tracking, Data Association, Attention Mechanism, Transformer, Kalman Filter**

I. Introduction

Air defense is facing increasingly complex scenarios. In fact, the challenges are growing in these theaters of operation with the increasing diversity of threats, ranging in size from different orders of magnitude, and which may be hypersonic, highly maneuverable, stealthy, and autonomous. Moreover, these threats are becoming more numerous and tend to work collaboratively. In a high-intensity conflict, control of the airspace provides enhanced intelligence capabilities and freedom of action for air, naval, and ground forces and assets, while limiting those of opposing forces [1]. Air defense aims to reduce the effectiveness of hostile airborne operations, such as reconnaissance and electronic warfare platforms, troop and material transport, fighters, bombers, conventional or nuclear missiles, in order to protect armed forces, civilians, military infrastructures, sensitive and high-value industry assets. It consists of detection, command, and control systems, as well as the means deployed to counter these threats: operational readiness, air interception, jamming, active air and missile defense. Therefore, these systems need to be upgraded accordingly to the complexity of emerging threats and air defense scenarios.

This work was supported by the French Defense Innovation Agency (AID) and Thales Land and Air Systems.

The state of an airspace is assessed using methods to detect and identify the various objects within it, while monitoring and tracking them by determining their kinematics with tracking techniques [2]. This also enables predicting the future state of this airspace. A major challenge lies in associating the available data from various sensors with previously determined kinematics. Managing the detection and tracking of multiple objects in an electromagnetically dense environment complicates the task, leading to false or missed detections. While some methods can associate this data based on specified criteria [3], their complexity often hinders the real-time processing constraint. Additionally, these methods may require information about the environment or objects that is often limited or unavailable [2]. Once the data association task is completed, techniques like Kalman Filters (KF) [4] are used to update the motion properties of the tracked objects. However, these methods are limited by assumptions about the kinematic models, which may not accurately reflect reality, posing a challenge for more complex movements such as those of maneuvering or hypersonic objects.

Recent methods based on Artificial Intelligence (AI), and in particular works emphasizing the use of deep learning, have attempted to overcome the limitations of non-AI methods, in order to perform data association, such as for the traveling salesman problem [5], [6] or the multi-object tracking problem [7]. These techniques are based on the use of an encoder-decoder architecture for sequence-to-sequence prediction. Other solutions [8]–[13] focus on improving certain elements or replacing the object kinematics update methods entirely, in order to overcome their weaknesses. These include reduced algorithmic complexity, more flexible and adaptable algorithms through the ability to learn properties of the environment and tracked objects through supervised model training. Finally, the attention mechanism used in Transformers [14] enables input data to be processed independently of the input sequence order, which is a limitation with sequential methods such as recurrent networks.

In this paper, we investigate a Transformer-based approach that aims to simultaneously address data association and state estimation [15], by adapting the architecture of the Transformer for the multi-target tracking problem. This work compares the proposed method with Kalman Filters, and studies

its performance under varying scenario parameters, such as clutter density, the number of objects, or their kinematics.

Transformers

The attention mechanism is the core process involved in Transformers. It enables the model to selectively focus on relevant information, while simultaneously considering the input elements. Thus it is permutation invariant and is capable of handling long-range dependencies. The attention mechanism computes attention scores that highlight the importance of different context elements. The attention scores are based on the compatibility between the input elements, the queries Q, and the context elements, the keys K. They are used to weigh the values V of the keys to compute the context of each query relative to the keys. This is similar to searching for a query in a database, with the queries being the elements of Q, and the database represented by K and V.

The attention mechanism can be applied simultaneously across n_{heads} multiple attention heads, unlocking the ability to focus on different aspects of the inputs, and therefore consequently weigh the context depending on the aspects that are learned during learning.

The Transformer follows an encoder-decoder architecture. The encoder is a stack of N encoder blocks which are the succession of a multi-head self-attention layer and a FeedForward Network (FFN) layer. The decoder is a stack of M decoder blocks, which are the succession of a multi-head self-attention layer, a multi-head cross-attention layer, and an FFN layer. In addition, each layer has a residual connection, which allows better propagation of the gradient and information through the neural network.

II. Transformer-Based Tracker

The following approach proposes to estimate the kinematics of the potential objects present in a spatial and temporal observation window, using single-sensor measurements.

A. Architecture

The approach is based on the Transformer architecture, which is suitable for approximating sequence-to-sequence functions. Thus, it comes down to a translation problem, where the input data represented in the "measurement language" of the sensor, are summarized into a "state language", which describes the characteristics of each detected object. In addition, we assume that a single common input token format for the data reduces the complexity of the system, making it less challenging to design, test, update, and maintain than more conventional tracking methods. The work presented in [15] aims at adapting this architecture for the multi-target tracking problem, as shown in Fig. 1.

The input data is transformed by affine transformations into the latent space, both in encoder and decoder, and are learned through training. Similarly, the output of the decoder passes from the latent space to the output space through a final affine transformation.

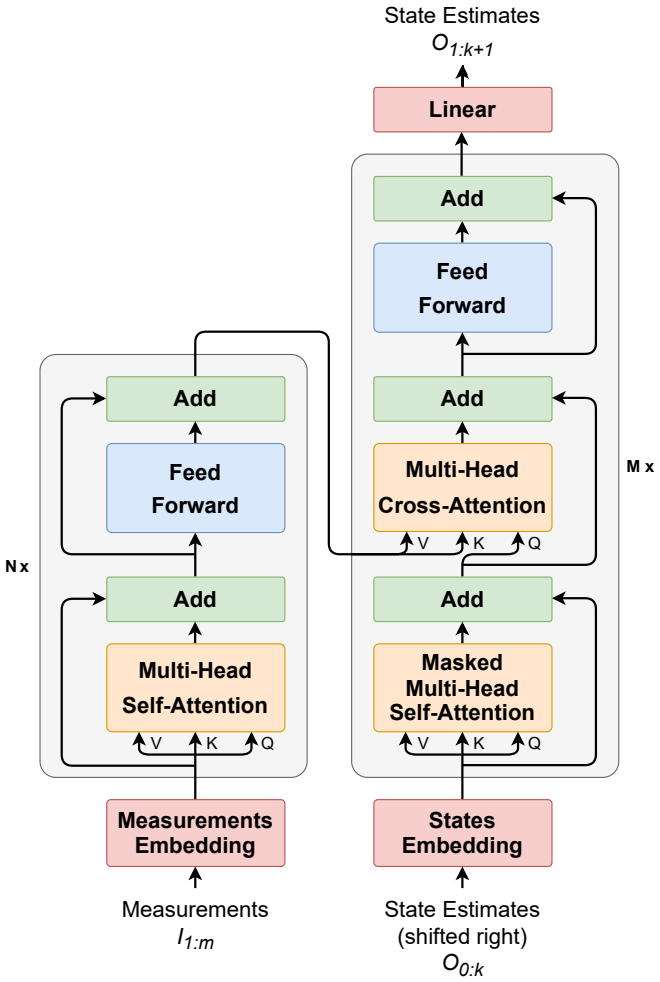


Fig. 1. Illustration of the Transformer architecture applied for multi-target tracking.

The encoder input $I_{1:m}$ of the Transformer, i.e. the input data, is composed of m measurements, which correspond to measurements occurring within a fixed time window and generated by a sensor, with $I_j = [t_j, x_j^{meas}, y^{meas}]$, $j \in [1..m]$, where t_j is the measurement time, x_j^{meas} is the measured position on the x-axis, and y_j^{meas} is the measured position on the y-axis.

The decoder output $O_{1:k+1}$, i.e. the "prediction" is composed of k states and the end state, indicating the end of the decoding process such as the end token in Natural Language Processing (NLP), and which all variables are equal to val_{end}. $O_{1:k}$ corresponds to the k current state estimates of the corresponding objects, with $O_i = [x_i, y_i, v_{xi}, v_{yi}, \dot{\theta}_i]$, $i \in [1..k]$, where x_i is the estimated x-axis position, y_i is the estimated y-axis position, v_{xi} is the estimated x-axis speed, v_{yi} is the estimated y-axis speed, and $\dot{\theta}_i$ is the estimated turn rate. We extend these notations to the end state O_{k+1}.

The input data are encoded to a better representation through learning in the encoder part. The prediction is then processed in the decoder part, according to the previous predictions and

the representation of the encoded input data. The previous predictions are provided through the decoder input $O_{0:k}$, which is a shifted view of the prediction $O_{1:k+1}$. Indeed, predicting the i-th prediction token will make the Transformer attend the i previous tokens $O_{0:i-1}$, with O_0 corresponding to a token indicating the start of the decoding process, such as the start token in NLP, and which all variables are equal to val_{start}. This enable using parallelism instead of sequentially computing each prediction tokens, which speeds up the learning process. This requires to prevent each prediction token to attend the previous ones, by putting their corresponding attention compatibility score to 0. However, this parallelism trick is applied during training only, otherwise autoregressive inference has to be performed.

The current states estimates $S_{1:k}$ are derived from $O_{1:k}$ following (1), with $S_i = [x_i, y_i, v_i, \theta_i, \dot{\theta}_i]$, $i \in [1..k]$, where v_i is the estimated speed norm, and θ_i is the estimated orientation, i.e. the object's heading. These state estimates $S_{1:k}$ are used to predict the future kinematics of objects.

$$\begin{cases} v_i = \sqrt{v_{xi}^2 + v_{yi}^2} \\ \theta_i = \arccos\left(\dfrac{v_{xi}}{v_i}\right) \ \mathrm{sgn}(v_{yi}) \end{cases} \tag{1}$$

The loss function used to train the model is set as

$$\mathcal{L} = \lambda_{pos}\mathcal{L}_{pos} + \lambda_{spd}\mathcal{L}_{spd} + \lambda_{\dot{\theta}}\mathcal{L}_{\dot{\theta}} + \lambda_{end}\mathcal{L}_{end} \tag{2}$$

where partial losses are defined for the ground-truth states $G_{1:k+1}$, with $G_i = [x_i^g, y_i^g, v_{xi}^g, v_{yi}^g, \dot{\theta}_i^g]$, $i \in [1..k]$, and $G_{k+1} = [val_{end}, val_{end}, val_{end}, val_{end}, val_{end}]$, as

$$\begin{cases} \mathcal{L}_{pos} = \dfrac{1}{k}\displaystyle\sum_{i=1}^{k} \sqrt{(x_i - x_i^g)^2 + (y_i - y_i^g)^2} \\ \mathcal{L}_{spd} = \dfrac{1}{k}\displaystyle\sum_{i=1}^{k} \sqrt{(v_{xi} - v_{xi}^g)^2 + (v_{yi} - v_{yi}^g)^2} \\ \mathcal{L}_{\dot{\theta}} = \dfrac{1}{k}\displaystyle\sum_{i=1}^{k} |\dot{\theta}_i - \dot{\theta}_i^g| \\ \mathcal{L}_{end} = \|O_{k+1} - G_{k+1}\|_2 \end{cases} \tag{3}$$

III. Experiments

A. Dataset

The training and validation datasets are constructed from simulations. They consist in generating measurements from a sensor that attempts to measure the characteristics of the objects present in specific spatial and temporal observation windows. The input data is composed of these measurements, while the ground truth, i.e. the Transformer expected output, is composed of the characteristics of the objects we try to predict. We distinguish 2 training datasets: the first one without clutter and the second one including clutter.

The following units, such as time, positions, speeds, turn rates, noises, and error metrics, are arbitrary units. Indeed, they can be considered as real values that have been rescaled for speeding up the learning process.

The training datasets are composed of n_{train} air situation instances, where an instance describes up to $n_{targets}$ objects. Each object has a constant speed norm and a constant turn rate. However, turn rate transitions are possible in a measurement time window, i.e. an instantaneous turn rate change drawn from the same initial turn rate distribution, in order to provide tracking capability for highly maneuvering targets and trajectory changes. Since disabling transition on the extreme parts of the trajectory improves the learning process, a transition can occur only at a time $t_{trans} \in [t_{trans,min}, t_{trans,max}]$. Initial positions are uniformly drawn in $[-pos_{max}, pos_{max}]$ on all axes, speeds in $[v_{min}, v_{max}]$, initial orientations in $[-\pi, \pi]$, and turn rates in $[-\dot{\theta}_{max}, \dot{\theta}_{max}]$. The value of pos_{max} is determined so that each object is located within a fixed limit range $range_{max}$ at all times, considering the maximum speed v_{max} and the maximum duration T of the time window.

Sensor measurements are generated by adding a zero mean Gaussian noise $\mathcal{N}(0, \sigma^2)$ to the actual positions of the objects at each fixed timestep Δt in a time window of duration T. It is assumed that there are no false detections for the first training dataset. In the second training dataset, at each timestep, clutter is generated by uniformly drawing and adding up to $n_{clutter}$ positions located in a spatial squared window of length $2\,dist_{clutter}$ centered on the object real position. In addition, additional clutter is uniformly drawn with up to $n_{clutter}$ positions located in the acceptable space, i.e. having the x-axis and y-axis positions in $[-range_{max}, range_{max}]$. This enables correlated and uncorrelated false detections, as well as increasing the robustness to various clutter locations. Concerning sequence generation and formatting, the start value, end value, and padding value are set relatively to $range_{max}$, respectively $val_{start} = -1.5\,range_{max}$, $val_{end} = 1.5\,range_{max}$ and $val_{pad} = 2\,range_{max}$.

In this work, $n_{train} = 320K$, $n_{targets} = 5$, $t_{trans,min}$ and $t_{trans,max}$ are respectively equal to 0.12 and 0.28, $v_{min} = 0.4$ and $v_{max} = 2.0$, $\dot{\theta}_{max}$ is taken equal to 7.85 which represents a half turn in 10 timesteps, with $\Delta t = 0.04$. The maximum duration of a time window is $T = 0.36$, $range_{max} = 4.0$, $pos_{max} = 3.28$, $val_{start} = -6.0$, $val_{end} = 6.0$, and $val_{pad} = 8.0$. Concerning noise and clutter parameters, $\sigma = 0.0133$, $n_{clutter} = 2$, and $dist_{clutter} = 0.2$.

B. Transformer Tracker

The Transformer-based state estimation method is described in section II. The parameters selected for the experiments are an embedded latent dimension $d_{model} = 64$, hidden dimension of FFN $d_{hidden} = 256$, FFN activation function $= GELU$, number of attention heads $n_{heads} = 4$, no dropout, number of encoder blocks $N = 4$, number of decoder blocks $M = 4$.

The selected training parameters for the Transformer-based method are a batch size of 512, with 50 epochs, a gradient clipping of 1.0, the Adam optimizer [16] with no weight decay and $(\beta_1, \beta_2) = (0.9, 0.99)$, a OneCycleLR scheduler [17] with sinusoidal annealing strategy: an initial learning rate of 5e-8, a final learning rate of 1e-6, a maximum learning rate of 8e-4,

TABLE I
ABSOLUTE ERRORS ON POSITIONS PREDICTED 1 TIMESTEP FORWARD USING KALMAN FILTER, EKF, AND THE TRANSFORMER TRACKER.

Tracking Method	Straight Line		Constant Turn	
	Mean	Median	Mean	Median
Kalman Filter	0.0180	0.0155	0.0386	0.0327
EKF	**0.0175**	**0.0148**	0.0243	0.0221
Transformer Tracker	0.0228	0.0206	**0.0226**	**0.0200**

and warm-up proportion of 0.2. The selected architecture has a total of 465,349 parameters.

The loss parameters in eq. 2 are: $\lambda_{pos} = 1.0$, $\lambda_{spd} = 0.5$, $\lambda_{\dot{\theta}} = 0.25$, $\lambda_{end} = 0.1$.

C. Results

The results presented in Table I are the mean and median absolute errors of the estimated position at one time step after the last measurement of an object making either a constant turn or going in a straight line. The estimations are made with the proposed method, a Kalman Filter with a constant velocity model, and an Extended Kalman Filter (EKF) with a constant turn model. Both Kalman Filter and EKF have their noise matrices chosen such as using the exact measurement noise level σ. First, this shows the Kalman Filter and the Transformer-based tracker are performing worse than the EKF in a straight line. In this case, the Kalman Filter kinematics model matches the real object kinematics, while the EKF and the Transformer Tracker are more prone to deviate from ground truth because of noise. Depending on the noise, the more probable trajectory given the measurements could be a constant turn rather than a straight line, which is allowed by the EKF and the Transformer Tracker kinematics models. However, the Kalman Filter and EKF are tuned for better overall performances on constant turn, which makes the EKF still better than the Kalman Filter in the straight line case. In addition, the Transformer Tracker is performing worse than the other filters, which may be explained by the initialisation of the Kalman Filter and the EKF, which is based on the ground truth, i.e. an "oracle". In practice, the ground truth is unknown but the results are chosen to use this initialisation scheme to compare the Transformer Tracker with strong baselines. Moreover, in a constant turn case, the Transformer Tracker is performing better than the EKF and the Kalman Filter. In this case, the non-linear kinematics of the object makes the accuracy of the Kalman Filter way worse than EKF, which has a non-linear kinematic model. Finally, in the case of a kinematics transition, such as a sharp change in the turn rate, the difference in accuracy between the Transformer Tracker and EKF is greater, demonstrating the better reactivity of our method in the event of kinematic change. This can be observed in Fig. 2, which illustrates an example of a trajectory of an object performing two successive constant-turn.

The mean absolute errors on the next timestep position seems to be independant on the position of the object, as shown in Fig. 3 and Table II. The differences in the errors are due to aleatoric uncertainty of the sampling, decreasing when the number of samples increases. Thus, our method shows consistent results when varying position variables.

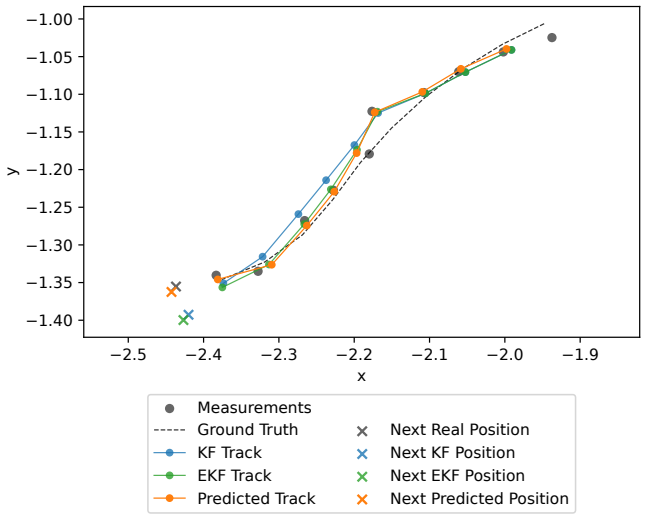


Fig. 2. Trajectory of a target performing two successive constant turn, with the Kalman filter, EKF, and the Transformer Tracker corresponding predictions. Gray dotted line and dots respectively represent the object trajectory and corresponding sensor measurements, colored dots represent the predicted positions at each time step, and colored x's represent positions predicted 1 timestep forward.

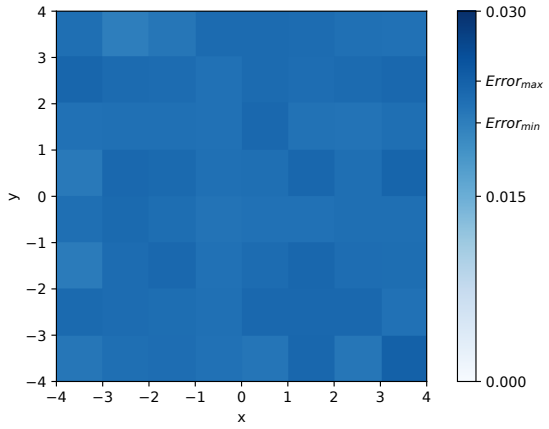


Fig. 3. Mean absolute errors on position predicted 1 timestep forward using the Transformer Tracker, depending on target position. $Error_{min}$ and $Error_{max}$ respectively represent the minimum and maximum error in the position grid.

This mean absolute errors on the next timestep position slightly increases when the speed of the object is increasing, as shown in Table III. It also seems to be consistent when the turn rate is changing, as shown in Table IV, although there is a slight increase for the highest turn rate category. The error associated with turn rates near zero is influenced by samples containing only 2 measurements, where the turn rate is set to zero to enhance the learning process. As demonstrated later, the mean error for samples with only 2 measurements is

TABLE II
MEAN ABSOLUTE ERRORS ON POSITION PREDICTED 1 TIMESTEP FORWARD USING THE TRANSFORMER TRACKER, DEPENDING ON TARGET RANGE r FROM WINDOW CENTER POSITION.

r	[0.0, 1.0]	[1.0, 2.0]	[2.0, 3.0]	[3.0, 4.0]	[4.0, 5.0]	[5.0, 6.0]
Error	0.0225	0.0230	0.0229	0.0228	0.0229	0.0222

TABLE III
MEAN ABSOLUTE ERRORS ON POSITION PREDICTED 1 TIMESTEP FORWARD USING THE TRANSFORMER TRACKER, DEPENDING ON TARGET SPEED v.

v	[0.4, 0.75]	[0.75, 1.0]	[1.0, 1.25]	[1.25, 1.5]	[1.5, 1.75]	[1.75, 2.0]
Error	0.0211	0.0222	0.0227	0.0234	0.0236	0.0241

TABLE IV
MEAN ABSOLUTE ERRORS ON POSITION PREDICTED 1 TIMESTEP FORWARD USING THE TRANSFORMER TRACKER, DEPENDING ON TARGET TURN RATE $\dot{\theta}$.

$\dot{\theta}$	[0.0, 1.3]	[1.3, 2.6]	[2.6, 3.9]	[3.9, 5.2]	[5.2, 6.5]	[6.5, 7.9]
Error	0.0264*	0.0216	0.0216	0.0214	0.0213	0.0222

* 0.0217 when discarding cases with only 2 measurements.

TABLE V
MEAN ABSOLUTE ERRORS ON POSITION PREDICTED 1 TIMESTEP FORWARD USING THE TRANSFORMER TRACKER, DEPENDING ON TIME t_{trans} WHERE TURN RATE TRANSITION OCCUR.

t_{trans}	no transition	[0.12, 0.16]	[0.16, 0.2]	[0.2, 0.24]	[0.24, 0.28]
Error	0.0228	0.0227	0.0228	0.0230	0.0231

higher than that for samples with more than 2 measurements. Consequently, these samples are mainly concentrated in the lower turn rate category, which overall increases the mean error. When discarding the samples with only 2 measurements, the mean error of the lowest turn rate category reduces to the error level of the other categories.

The impact of a turn rate transition on the mean absolute error of the following timestep position is presented in Table V. It shows that the error remains consistent whether or not there is a transition, and if there is a transition wherever it occurs. A marginal increase in the error can be observed when the transition occurs lately in the trajectory. In this scenario, the decreasing number of measurements available to estimate the new turn rate is reduced compared to the total measurements. Moreover, late transitions only appear in the case of trajectories with the highest number of measurements, and as seen in Fig. 5, also those with the best accuracy, which may then have an impact on the prediction accuracy.

As mentioned previously, Fig. 5 illustrates the mean absolute error of the following timestep position prediction of the Transformer Tracker, and it shows that increasing the number of measurements globally enhances the prediction accuracy. However, we still observe a slight decrease in accuracy for higher number of measurements, which is due to transitions that can occur more lately. In addition, the data generation process decreases the proportion of trajectories with no transition when increasing the number of measurements, further amplifying the phenomenon. The results of the proposed method are compared with the mean absolute error of a linear regression in the case of a zero turn rate. The linear regression expected mean error is described in (4). This shows the impact of the space dimension D of the problem, the noise level σ, the prediction time t_{new}, which increase the expected mean absolute error when increased, while increasing the number of measurements m decreases this error.

$$\sqrt{2}\,\frac{\Gamma\left(\frac{D+1}{2}\right)}{\Gamma\left(\frac{D}{2}\right)}\,\sigma\sqrt{\frac{1}{m}+\frac{(t_{new}-\bar{t})^2}{\sum_{i=1}^{m}(t_i-\bar{t})^2}} \tag{4}$$

where $D = 2$ is the space dimension, and t_{new} the time at which the prediction is made, here it is one timestep after the last measurement.

The linear regression error effectively decreases as the number of measurements increases, and it is globally lower than the Transformer Tracker error. This may be due to the more complex non-linear regression, with more degrees of freedom, that the Transformer Tracker has to perform in the case of a turn or during a transition. In addition, the error of the Transformer Tracker is lower than the linear regression for only 2 measurements. After further investigations, this is explained by the knowledge of our system, acquired during training, of the object's possible kinematics. In particular, this reduces regression errors due to highly noisy samples, which can be identified as highly unlikely, so that their processing can be adapted accordingly, improving overall accuracy. This difference fades as the number of measurements increases, i.e. as the effect of aleatoric uncertainty decreases.

The Transformer Tracker trained with up to 2 clutter measurements per timestep shows robustness and generalization capabilities, as shown in Fig. 6. An example trajectory is also illustrated in Fig. 4. The proposed method succeeds in scaling its accuracy relative to clutter, even up to 10 clutter measurements per timestep. In addition, it maintains its enhanced accuracy compared to Kalman Filter and EKF with the use of a Probabilistic Data Association Filter (PDAF) [18] in the presence of various levels of clutter. Finally, as already mentioned, the initialisation of Kalman Filter, EKF, and PDAF is based on the ground truth, which gives additional information that are unknown in practice. This highlights the performance of our method that still surpasses the Kalman Filter and EKF with PDAF.

The Transformer Tracker simultaneously performs data association and state estimation, making its inference time dependent only on the number of measurements and predicted states, resulting in quadratic complexity. This allows for parallelization and GPU acceleration, as shown in Table VI, where the inference times for single-object trajectories of the Transformer Tracker are comparable to those of EKF with PDAF. However, on a CPU, our method is generally slower

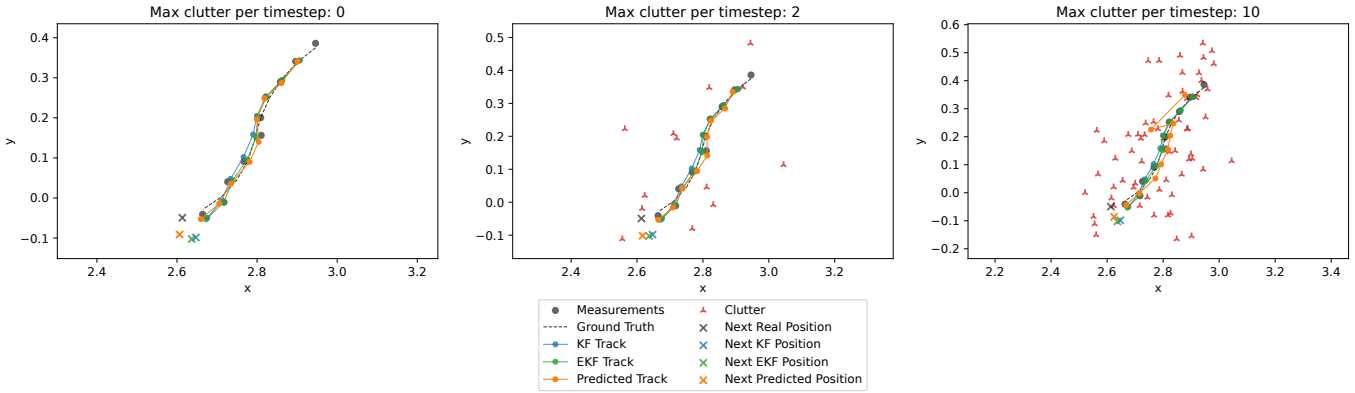


Fig. 4. Trajectory of a target performing two successive smooth turns, with the Kalman Filter with PDAF, EKF with PDAF, and the Transformer Tracker corresponding predictions. Colored dots represent the predicted positions at each time step, orange line and x corresponds to the Transformer Tracker predicted track and position predicted 1 timestep forward.

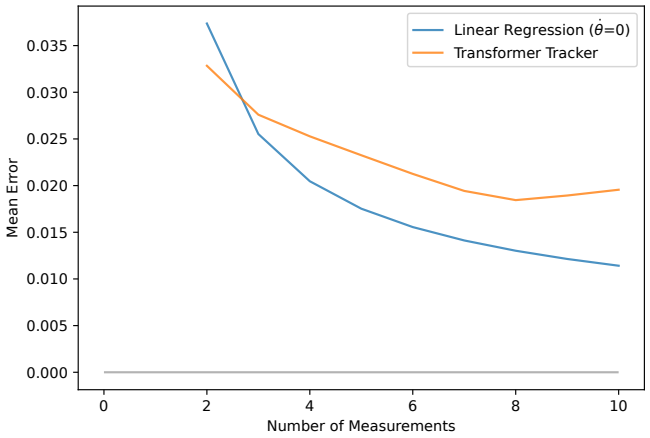


Fig. 5. Mean absolute errors on position predicted 1 timestep forward using the Transformer Tracker and a linear regression, depending on number of measurements. For the linear regression, only straight line trajectories ($\dot{\theta} = 0$) are considered.

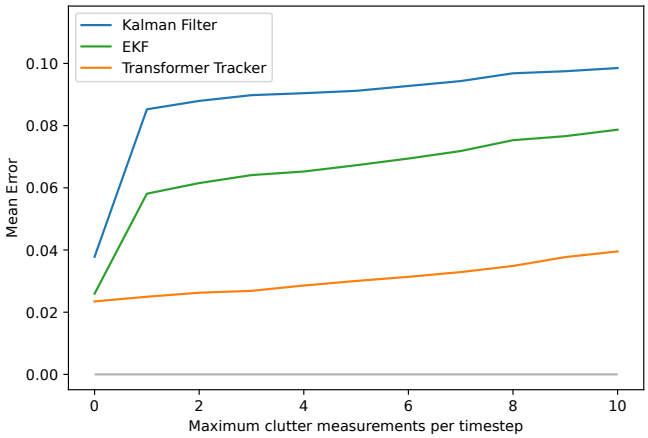


Fig. 6. Mean absolute errors on positions predicted 1 timestep forward using Kalman Filter with PDAF, EKF with PDAF, and the Transformer Tracker, depending on the maximum number of clutter measurements per timestep.

than EKF with PDAF, and becomes impractical with a large number of measurements. Additionally, when using EKF with PDAF, inference time increases with longer time windows due to more EKF updates and PDAF steps. While methods like Joint-PDAF offer finer associations, they suffer from factorial complexity, which our method overcomes with its quadratic scaling.

The Transformer-based tracker multi-target capability is shown in Fig. 7. Moreover, the generalisation on the number of objects is investigated in Fig. 8. More details about this behavior are given in the Appendix. It shows that the Transformer Tracker succeeds in predicting the correct number of objects on the training range, i.e. up to 5 objects, and the performance start decreasing after this threshold, to the point where there are almost no good predictions at 8 objects and above. This inability to generalize on the number of objects may be because the Transformer overfits on the maximum number of measurements which are proportional to the number of objects here, or/and overfits on the number of predicted states which correspond to the number of objects and the end token. This behavior matches the Transformers behavior for NLP applications [19].

TABLE VI
MEAN INFERENCE TIMES FOR SINGLE-TARGET TRAJECTORIES, DEPENDING ON NUMBER OF MEASUREMENTS, TRAJECTORY DURATION, AND DEVICE, USING THE EKF WITH PDAF AND THE TRANSFORMER TRACKER.

Tracking Method	Number of measurements			
	10	100	1,000	5,000
EKF with PDAF ($T = 0.36$)[1]	9.6 ms	12.0 ms	33.4 ms	121 ms
EKF with PDAF ($T = 3.96$)[1]	—	93.8 ms	118 ms	218 ms
Transformer Tracker[1]	12.2 ms	15.9 ms	230 ms	5710 ms
Transformer Tracker (GPU)[2]	16.3 ms	16.4 ms	21.4 ms	141 ms

[1] One core of an Intel i7-9850H CPU.
[2] Nvidia GeForce GTX 1080 8GB GPU.

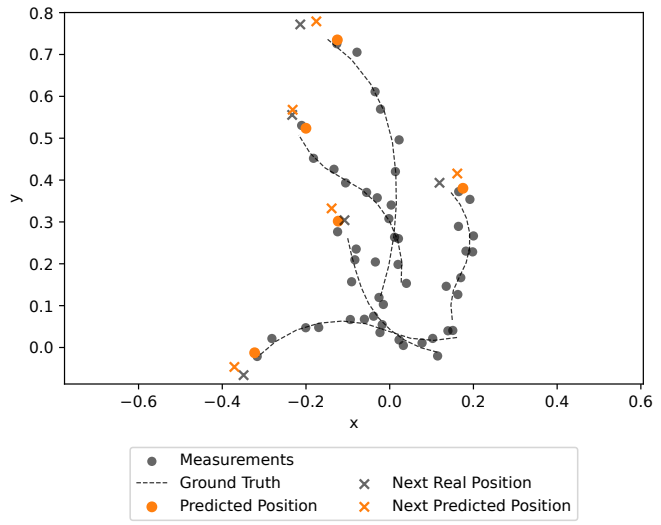


Fig. 7. Trajectory of 5 objects of various kinematics, with the Transformer Tracker corresponding predictions.

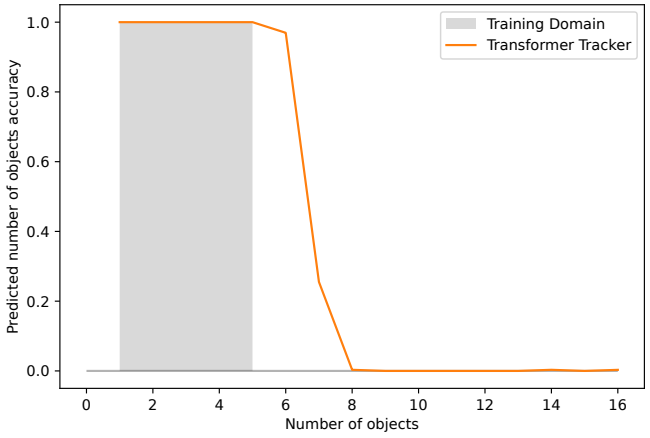


Fig. 8. Number of objects prediction accuracy of the Transformer Tracker, when it is trained with up to 5 objects in an air picture.

Finally, we investigate how the information is used by the Transformer Tracker to make the prediction. We use the attention rollout method [20] to output how the input tokens, i.e. the measurements, and the already estimated states are propagating the information to new state estimates, as shown in Fig. 9. The attention rollout propagates the attention weights to determine a unique attention map that represents how much the Transformer is paying attention to its encoder and decoder inputs, i.e. the measurements and previously predicted states. This is applied to an example trajectory, without clutter. For visibility purposes, the measurements are ordered in ascending time for each object, which are arranged in ascending order of position on the x-axis, to match the output order. Thus, the first measurements correspond to the first target, the next measurements to the second, and so on. For each object, i.e. its corresponding state, the Transformer Tracker seems to mainly

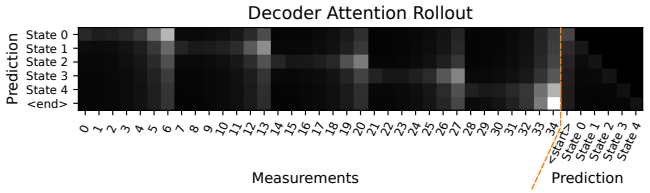


Fig. 9. Transformer Tracker decoder attention rollout for measurements and corresponding predictions of an air picture of duration $6\Delta t$ with 5 objects of various kinematics. Each row corresponds to a state estimation, where the intensity of its cells represents "how much" the measurements and the previous predictions are involved in the state estimation, i.e. how much the Transformer is paying attention to them during the prediction for this specific state estimation.

pay attention to measurements of this object, with a particular emphasis on the most recent ones.

IV. Conclusions and Future Work

In this work, we performed an exploratory analysis of the behavior of a Transformer applied to multi-target tracking under various kinematics parameters, clutter density, and number of objects. The evaluation was conducted in a simulated environment and aimed to assess the accuracy and consistency of the Transformer-based tracker method. The method was compared with the Kalman Filter and Extended Kalman Filter with adapted process and noise covariance matrices. It showed better state estimation in the general case, coming from better reactivity and knowledge building from learning data. This last behavior proves the Transformer Tracker yield better prediction capability than more general models. However, the results show a lack of generalization capability on the number of objects. Finally, the attention mechanism provides preliminary insight into the system's explainability. The reported findings provide valuable insights into the efficiency and adaptability of Transformer-based tracking methods in dynamic object motion estimation scenarios, performing object detection and state estimation simultaneously.

As a consequence, interesting future research directions include testing the Transformer-based tracking method for more realistic and complex target tracking scenarios. In particular, objects with kinematics of different order of magnitude such as speed or maneuverability could represent a great challenge for the Transformer Tracker. The lack of generalization on the number of objects raises valuable directions for research that can also be relevant to other fields, such as in natural language processing. Additional exploration could be conducted relating to multi-target tracking capability such as in crowded air situations, crossing trajectories, swarming, as well as using multiple sensors. Track management also needs to be investigated and tested for dynamic simulations, such as asynchronous track creation and deletion. Finally, the attention mechanism could be further exploited, in particular by transposing techniques used in natural language processing and computer vision for more convincing explainability and explicability results.

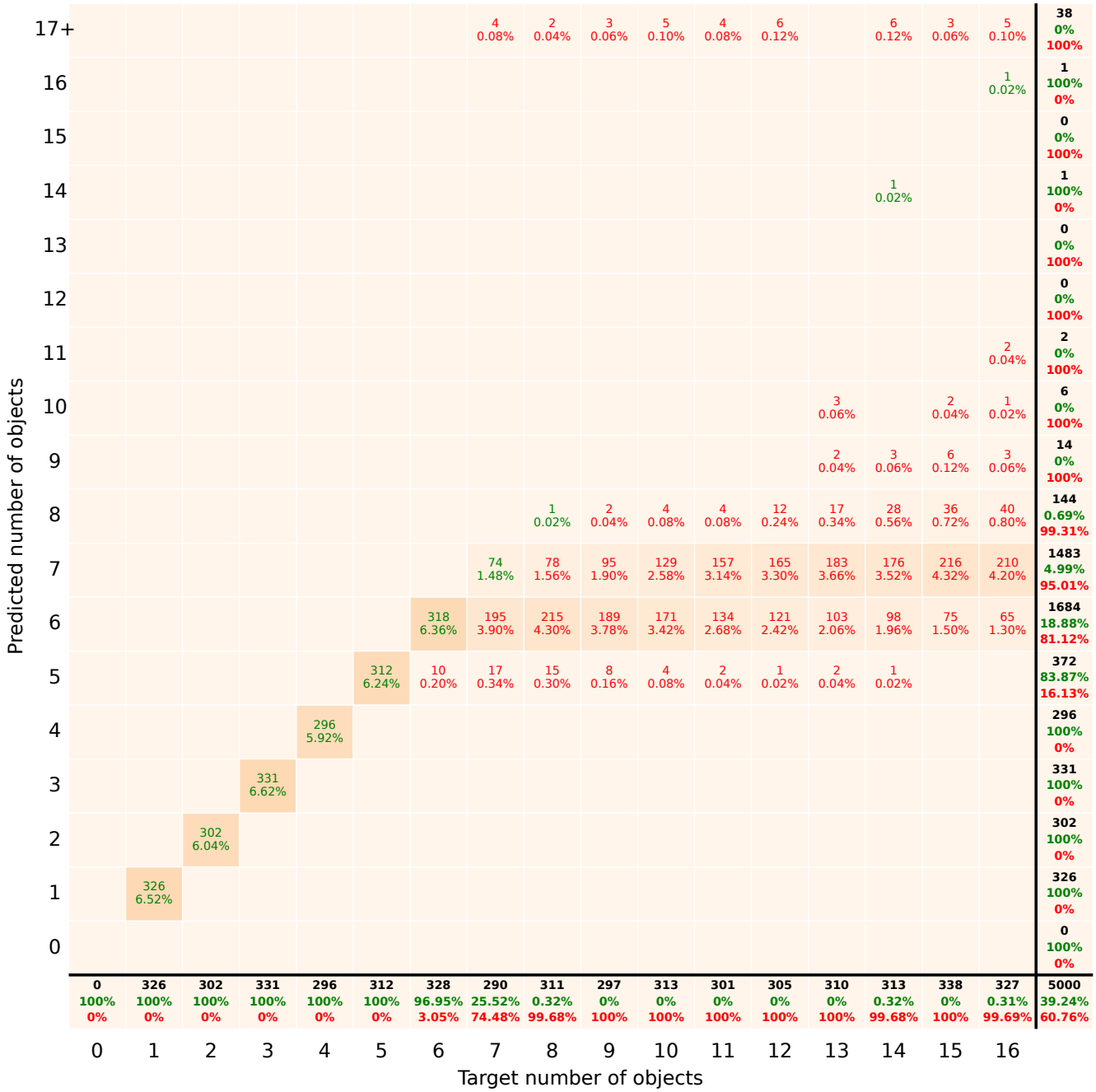


Fig. 10. Confusion matrix of the number of objects, for a Transformer Tracker trained with up to 5 objects in an air picture. In the confusion matrix cells, the first number is the number of samples for a given number of objects (column) and a given predicted number of objects (line), and the percentage is the proportion of the test samples they represent. These numbers are displayed in green when the prediction is correct and in red if not. In the last line, for each column representing a given number of objects, black numbers represent how many samples have the corresponding number of objects, green numbers represent the percentage of correct predicted number of objects, and red numbers represent the percentage of wrong predicted number of objects. In the last column, for each line representing a given predicted number of objects, black numbers represent how many samples have the corresponding predicted number of objects, green numbers represent the percentage of correct predicted number of objects, and red numbers represent the percentage of wrong predicted number of objects.

REFERENCES

[1] P. Steininger, *Les Fondamentaux de la puissance aérienne moderne*. L'Harmattan, 2020.

[2] Y. Bar-Shalom and X.-R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*. Yaakov Bar-Shalom, 1995.

[3] L. Rakai, H. Song, S. Sun, W. Zhang, and Y. Yang, "Data association in multiple object tracking: A survey of recent techniques," *Expert Systems with Applications*, vol. 192, p. 116300, 2022.

[4] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 03 1960.

[5] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, (Cambridge, MA, USA), p. 2692–2700, MIT Press, 2015.

[6] X. Bresson and T. Laurent, "The transformer network for the traveling salesman problem," 2021.

[7] J. Pinto, G. Hess, W. Ljungbergh, Y. Xia, L. Svensson, and H. Wymeersch, "Next generation multitarget trackers: Random finite set methods vs transformer-based deep learning," 2021.

[8] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, Feb. 2017.

[9] K. Yoon, D. Y. Kim, Y.-C. Yoon, and M. Jeon, "Data association for multi-object tracking via deep neural networks," *Sensors*, vol. 19, no. 3, 2019.

[10] H. Liu, H. Zhang, and C. Mertz, "Deepda: Lstm-based deep data association network for multi-targets tracking in clutter," in *2019 22th International Conference on Information Fusion (FUSION)*, pp. 1–8, 2019.

[11] Y. Yao, I. Smal, I. Grigoriev, A. Akhmanova, and E. Meijering, "Deep-learning method for data association in particle tracking," *Bioinformatics (Oxford, England)*, vol. 36, 07 2020.

[12] S. Jouaber, S. Bonnabel, S. Velasco-Forero, and M. Pilté, "Nnakf: A neural network adapted kalman filter for target tracking," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4075–4079, 2021.

[13] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. G. van Sloun, and Y. C. Eldar, "Kalmannet: Neural network aided kalman filtering for partially known dynamics," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1532–1547, 2022.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 6000–6010, Curran Associates Inc., 2017.

[15] V. Sonntag, J.-M. Le Caillec, A. Peres, and S. Devaud, "Transformer-based state estimation for tracking: Maneuvering target and multi- target capabilities," in *2024 IEEE Radar Conference (RadarConf24)*, pp. 1–6, 2024.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[17] L. N. Smith and N. Topin, "Super-convergence: very fast training of neural networks using large learning rates," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications* (T. Pham, ed.), vol. 11006, p. 1100612, International Society for Optics and Photonics, SPIE, 2019.

[18] Y. Bar-Shalom and E. Tse, "Tracking in a cluttered environment with probabilistic data association," *Automatica*, vol. 11, no. 5, pp. 451–460, 1975.

[19] C. Anil, Y. Wu, A. Andreassen, A. Lewkowycz, V. Misra, V. Ramasesh, A. Slone, G. Gur-Ari, E. Dyer, and B. Neyshabur, "Exploring length generalization in large language models," in *Advances in Neural Information Processing Systems* (S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds.), vol. 35, pp. 38546–38556, Curran Associates, Inc., 2022.

[20] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, eds.), (Online), pp. 4190–4197, Association for Computational Linguistics, July 2020.

Methodology for explainable, consistent, and generalizable Reinforcement Learning drone control

Robinson Denève
NEODE SYSTEMS
Paris, France
robinson.deneve@ neode-systems.com

Paul Chaudron
MBDA
Paris, France
paul.chaudron@mbda-systems.com

Axel Puig
NEODE SYSTEMS
Paris, France
axel.puig@neode-systems.com

Alexandre Kotenkoff
MBDA
Paris, France
alexandre.kotenkoff@mbda-systems.com

Mathias Formoso
MBDA
Paris, France
mathias.formoso@mbda-systems.com

***Abstract* — Future aircraft systems must adapt to the unknowns of their environment. During a mission, a drone swarm must adapt its flight formation. Each drone must reach its set position as fast as possible while keeping his front sensor in the same direction as the swarm. Moreover, the computation must be done on board and is called at a high frequency. The control algorithm of the drone must ensure complete reliability of the aircraft system.**

We developed a Deep Reinforcement Learning based control algorithm that outperforms baseline algorithms. Using neural networks in critical systems has many flaws that we were able to overcome thanks to a precise methodology:

- **Explainability: addressed through global and local analyses. The use of the discretized neural network allows a drone operator to validate the decision-making process. The drone operator does not need to be an AI expert.**
- **Consistency: addressed with a supervisory algorithm. It ensures convergence to the set position while using the trained Neural Network only when it leads to better performances. It uses an allocation algorithm and safeguards.**
- **Generalization: addressed with optimal training scenarios. Adaptation capabilities checked by testing on test scenarios.**

Keywords — Deep Reinforcement Learning, Reliability, Explainability, Consistency, Generalization, Swarm, UAV, Control, Critical System

I. INTRODUCTION

Reinforcement Learning (RL) is a machine learning method that helps an agent optimize its actions within a given environment to maximize its cumulative reward. The agent learns through trials and errors by interacting with its environment. RL has gained considerable interest in the early 21st century, especially in automation [1]. More recently, deep neural networks have been employed to address complex nonlinear challenges, such as excelling at Atari games [2] and mastering the board game of Go [3].

Deep Reinforcement Learning (DRL) is a subfield of Reinforcement Learning (RL) that combines the principles of RL with deep neural networks. In traditional RL, the agent's policy, which maps observations to actions, is typically represented using tabular methods or simple function approximators. DRL allows the agent to learn more complex and abstract representations of its environment, enabling it to tackle intricate problems and high dimensional continuous spaces. For instance, DRL has been used in [4] for assets swarm coordination for collaborative combat and has shown promising results. Control of drone fleets is a challenging task that can greatly benefit from the power of DRL technology.

However, using deep neural networks comes with drawbacks, primarily due to their intricate and nonlinear opaque architectures [5, 6]. Additionally, the Reinforcement Learning training process brings distinct challenges.

Explainability: The decision-making logic of a neural network is inherently difficult to interpret. Although some research suggests methods for analysis—like examining neuron activation or the relevance of individual features, and conducting semantic assessments of Deep Neural Networks (DNN) [6]—these approaches are labor-intensive and require

deep expertise in Deep Learning. Additionally, it is possible to perform both global and local analyses to dissect a neural network's decision-making process. Employing a surrogate model with a simpler architecture can also provide valuable insights, facilitating easier analysis and explanation of the neural network's behavior [7].

Consistency: The inconsistency of deep neural networks is a notable concern; minor input variations can lead to significant errors [8]. This vulnerability extends to networks trained via reinforcement learning for tasks requiring continuous control, pertinent to our study [10]. To mitigate this issue, methods such as data augmentation and introducing perturbations during training are commonly employed. Additionally, using adversarial networks that deliberately disrupt inputs and stability training techniques can further enhance the robustness of these systems [9]. As input disturbance are likely in real world applications, raw Deep Neural Networks cannot be used in critical systems.

Generalization: The ability of neural networks trained with Reinforcement Learning algorithms to adapt to new environments is limited. Introducing a test environment can help evaluate a network's generalization abilities [11]. Methods like data augmentation and L2 regularization, decrease overfitting but do not assure comprehensive enough generalization [12, 13]. More sophisticated strategies, such as Meta Reinforcement Learning, increase adaptation capabilities but necessitate retraining [15].

This article proposes a methodology that enables the integration of Deep Reinforcement Learning (DRL) algorithms in critical systems while addressing those issues:

- To the best of our knowledge, our pioneering approach is the first to promote the discretization of neural networks, although adopting more explainable models has been previously suggested [7]. This strategy not only enhances the consistency and explainability of the model but does so through a method that is straightforward to implement.
- We define a set of training scenarios designed to confirm our model's ability to generalize across diverse and complex trajectories.
- We introduce various methods to analyze neural network behavior and decision-making processes using local and global analysis [7].
- A supervisory algorithm is incorporated, using baseline algorithms to maximize performance while ensuring the drone accomplishes his objective thanks to an allocation algorithm and safeguards.

II. Framework and Environment

A. Description of the Use Case

During operations, a swarm of drones must dynamically adjust its formation depending on its environment. For example, changes may be required due to technical malfunctions or if a drone needs to leave the swarm to scout ahead. Each drone is equipped with a front-facing sensor or camera. It is essential for the drone to keep his sensor aligned with the swarm's overall direction in order to effectively anticipate obstacles and keep watching targeted areas of the environment.

As soon as a drone has received the new swarm flight formation, it must relocate as quickly as possible to its set position. Also, it must face the same direction as the rest of the swarm, despite the drones executing complex maneuvers. All computational processes must be conducted onboard and are executed at high frequencies.

Fig.1 Illustration of a possible initialization and the expected drone behavior

As stated earlier, the drone must comply with several constraints:

- Keep the same overall direction as the swarm. We write d_{swarm} and d_{drone} the direction of each asset. The direction is defined by $d_{\text{asset}} = \frac{speed_{\text{asset}}}{\|speed_{\text{asset}}\|}$.

  We must have $d_{\text{swarm}} \cdot d_{\text{drone}} \geq 0$
- Join the set position with a distance < 1500m and a speed difference < 5%

Once it is close enough from its set position, another algorithm takes over.

Absolute reliability is required in the control algorithm.

B. Environment and model

For this study, we used a fixed-wing UAV model that is limited in speed and acceleration. This work could be adapted to any drone model.

In the given simulation, the position of each drone within the swarm is determined by two variables:

- The central position of the swarm, which is shared among all drones.
- The specific position of the drone within the swarm, relative to the swarm's center.

For the current study, we consider the horizontal movement of drones, disregarding vertical speed control. We treat the management of altitude and horizontal speed as separate problems, as altitude control is relatively straightforward and does not necessitate complex algorithms. Furthermore, we employ the third dimension and flocking algorithms to prevent collisions between drones by organizing them in tiers [14].

Thus, our primary focus lies in controlling the horizontal speed of the drones.

C. Reinforcement Learning for UAV control

We developed a drone swarm control system using a Deep Reinforcement Learning. The RL problem is defined as a Partially Observable Markov Decision Process (POMDP). A POMDP consists of: a set of possible states S, a set of actions A, the probability of transitioning from one state to another P, a reward function R and a set of observations O, which equals to S if the process is fully observable.

At each time step t, the agent receives an observation $o_t \in O$ from the environment, which is based on the current state $s_t \in S$. The agent then chooses an action $a_t \in A$ based on its policy π, which maps observations to a probability distribution over actions. The environment then applies the action and the current state to generate the next state s_{t+1} and a reward $r_t \sim R(s_t, a_t)$ from the reward function.

The goal of the agent is to find a policy that maximizes its sum of expected discounted rewards over time:

$$\pi = \underset{\pi}{\operatorname{argmax}} J(\pi)$$

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t(s_t, a_t) \right]$$

where $\gamma \in [0, 1]$ denotes the discount factor.

1) *Initialisation of training episodes*

During the training process, the environment is initialised with random initial conditions :

- **Drone position**: the drone spawns randomly around the swarm's center.
- **Drone heading**: The initial heading of the drone is random.
- **Set position**: The drone has to reach a random set position within the swarm.
- **Swarm trajectory**: The swarm may follow either a random turn with a random radius or move in a straight line with a slight curve.

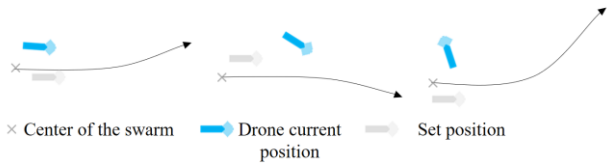


Fig. 2 - Illustration of random training initializations

2) *Episode termination*

During the training process, we have to decide when an episode ends. Gymnasium library introduced two variables to check if an episode is finished: truncated and terminated [16]. Truncation allows RL algorithm to manage episode time limits when the agent does not have access to a time-linked observation.

Our approach involves considering an episode as terminated when the UAV reaches its set position. Conversely, an episode is considered truncated when the time limit is reached or if the drone gets too far from the set position.

3) *Observation and action space*

The observation vector consists of the normalized relative position, distance of the set position and the heading of the swarm, all bounded between -1 and 1. The distance value undergoes normalization with an increasing bijection from $[0, \infty]$ to $[0,1]$, preserving all essential information.

The action encompasses the heading and speed magnitude of the drone. We use the following functions:

$$f_{\text{preprocess}}: \mathbb{R}^2 \times \mathbb{R} \rightarrow [-1,1]^5$$

$$NN: [-1,1]^5 \rightarrow [-1,1]^2$$

Consequently, the action is determined by:

$$\text{action}(\text{observation}) = NN\left(f_{\text{preprocess}}(\text{observation})\right) \quad \textit{Eq.1}$$

D. *Reward system*

RL algorithm try to optimise an actor neural network in order to maximise a reward function. This function shall score an action based on its expected utility in achieving a specific goal, thus guiding the actor towards the most effective behaviours.

We design a simple reward function based on two elements to evaluate the drone performance:

- A direction reward: whenever the model doesn't follow the same direction as the swarm it gets a huge penalty.
- Distance reward: at each time step, the agent receives a reward. The closer the drone gets from its set points, the greater the reward is.

E. *Reinforcement training algorithm*

In our study, we employed the Gymnasium environment and Stable Baselines3 Python library for reinforcement learning. We opted for Proximal Policy Optimization (PPO) and Soft Actor-Critic (SAC) [17] algorithms due to their current state-of-the-art status in the field. Although PPO demonstrated faster training times, it did not achieve the same level of performance after extended hours of training.

III. RELIABLE REINFORCEMENT LEARNING METHODOLOGY.

We trained a reinforcement learning (RL) model using Soft Actor-Critic (SAC) for two million steps on an eight-CPU workstation. Optimal performance was achieved after one million steps and 4.5 hours. As stated in the introduction, using reinforcement learning brings some drawbacks:

- **Consistency**: The RL model may have an erratic behavior in response to certain observations.

- **Generalization**: RL algorithms often underperform in scenarios that differ from the training environment.
- **Explainability**: The actions of the RL model may not be interpretable enough to establish the trust of drone operators. The model needs to be acceptable for them.

These issues render RL impractical for critical systems, where unpredictability causes significant risks for the system. We propose a certification protocol that performs several analyses to address these concerns.

A. *Global Analysis, comparison with other algorithms*

We evaluated the performance of our trained reinforcement learning algorithm against more traditional baseline algorithms on specific scenarios. These scenarios were carefully selected to mimic the conditions the missile pack is likely to encounter in the intended use case. The primary objective of this comparison is to gain a global understanding of the advantages brought about by employing RL algorithms.

- **Pursuit Algorithm (PURSUIT):** The classical pursuit algorithm directs an asset to intercept a moving target by aiming at its current position. It has been modified with a predictive component that aims at a point slightly ahead of the target's location based on its velocity. This modification enhances stability. Additionally, the algorithm is designed to accelerate when it falls behind the set position [18, 19].

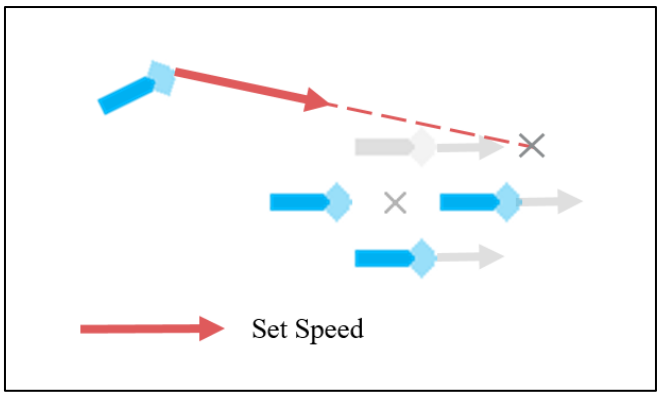


Fig. 3 Illustration of the PURSUIT Algorithm functioning

- **Proportional controller (PROPORTIONAL):** The proportional control algorithm calculates the required set speed of the drone using the following equation:

$$\text{Set Speed} = \text{Swarm Speed} + \text{Correction Speed}$$

$$\text{With Correction Speed} = \gamma \text{Position Target}$$

Position Targeted is the relative position of the set position in the drone reference frame and γ is a positive constant such that $\gamma \ll 1$.

Viewed from the swarm's perspective, the drone's velocity relative to its target is defined as γ Position Targeted. The drone approaches its set position by effectively reducing the relative distance. The drone's direction closely matches that of the swarm, so it complies with the direction constraint.

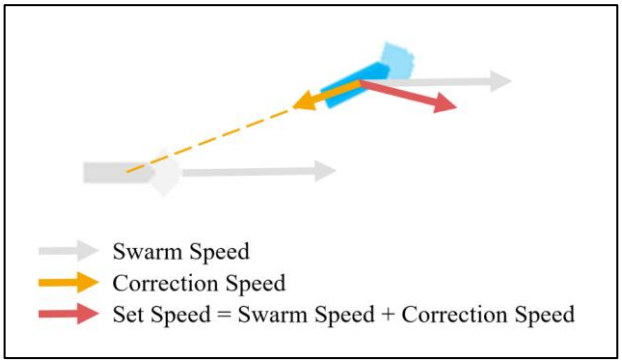


Fig. 4 Illustration of the PROP Algorithm functioning

1) *Benchmark scenarios*

In order to have a comprehensive understanding of the upsides and downsides of the reinforcement learning agent. We tested our model on 500 test episodes. Those episodes are variation of the training episodes previously described. When evaluating the models, we considered two key criteria:

- **Approach speed**: This metric represents the average velocity at which the drone approaches its set position, as measured in the swarm's reference frame, during an entire episode.
- **Compliance with the direction constraint**: This criteria ensures the drone keeps its heading aligned with the one of the swarm.

a) *Catch Up Scenario.*

We assessed the algorithms' capabilities in a scenario where the drone's initial position is situated behind the swarm's center. The episode finishes when the drone reaches a distance of x meters from the swarm. The drone has to catch up the swarm.

In our experiments, both classical algorithms demonstrated superior performance compared to the reinforcement learning model, despite all algorithms complying with the direction constraint. Among the classical algorithms, the pursuit algorithm showed a slightly better approach speed than the proportional controller.

	RL	PURSUIT	PROPORTIONAL
Average approach speed (m/s)	54	**57**	56

b) *Close up Scenario*

One other scenario was implemented and tested. The drone's initial position was randomly placed between 0 and 1500 meters away from the swarm. The episode concluded when the drone reached a distance of 5 meters from the swarm.

Once again, both classical algorithms (PURSUIT and PROPORTIONAL) perform better than RL. Indeed, both of them were able to join the set point with a precision of 5m meters whereas RL model was only able to oscillate around the set points with a range of 50 meters.

PROPORTIONAL performed slightly better and was able to reach the set point with a precision of 2 meters and it was faster than the PURSUIT.

c) Ahead spawn scenario

Eventually, we tested every algorithms on the ahead spawn scenario. The drone spawns ahead of its set position. It ends once it is 1500 meters away from its set position.

Our results revealed that RL outperformed both algorithms by executing innovative and complex maneuvers. PURSUIT failed to maintain the directional constraint, while PROP managed only to decelerate while maintaining the same heading as the swarm. In contrast, RL performed zigzags allowing it to reach its set position faster as shown in Fig. 5.

The model learned to perform a zigzag maneuver to travel a longer distance to let the swarm catch it up.

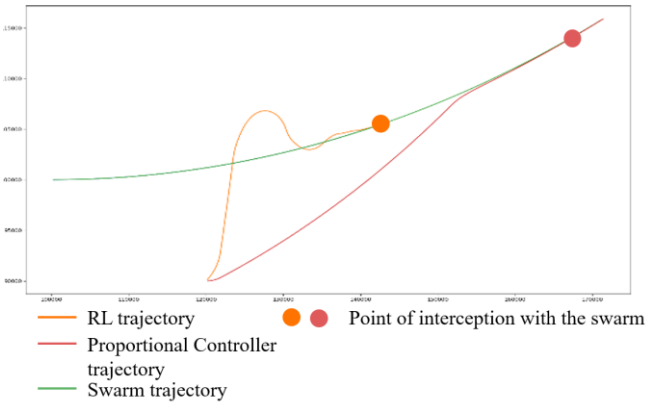


Fig. 5 - Trajectory of the trained RL model and of PROPORTIONAL.

	RL	PURSUIT	PROPORTIONAL
Comments	Zigzag maneuver	Unable to respect the direction constraint	Slow maneuver
Average approach speed (m/s)	**117**	×	68

2) Conclusion

A comprehensive analysis of the RL algorithm's behaviour in the aforementioned scenarios provides valuable insights into its strengths and limitations. Given the complexities and constraints associated with RL, it is crucial to restrict its application to situations where its advantages are the most significant.

We propose implementing a supervisory algorithm (SUPERVISED RL) that uses the optimal algorithm to employ based on the drone position relative to its set position. Using the results of the global analysis, we select the most appropriate algorithm to ensure the best possible performance depending on the drone position.

The different algorithms are used according to the drone's relative position, following the allocation pattern presented in Fig. 6.

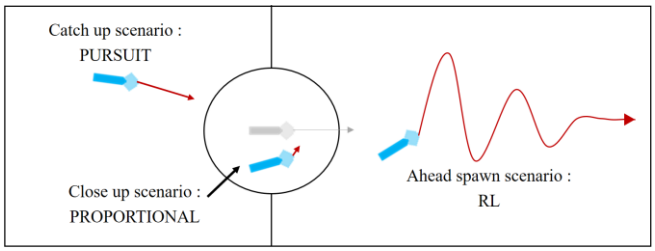


Fig. 6 - Allocation of each algorithm made by the SUPERVISED RL algorithm depending on the drone relative position.

B. Local Analysis

To ensure the reliability of our reinforcement learning model, it is essential to analyze what precisely it is doing and ensure that there is not any discontinuity or aberration that the global analyses would have missed.

We analyzed the behavior of the RL on high-level scenarios. In particular, we know that a zigzag maneuver is carried out to optimize the catch-up time. Yet we do not know exactly how this maneuver is performed and what specific actions are chosen by the neural network. Moreover, his behavior must be validated by a drone pilot/operator to ensure that the actions are consistent and not risk failing in real life environment. Indeed, not all the technical constraints were taken into account during the development of the simulation and we need to make sure that there are no backdoor in the neural networks. Finally, we must ensure that there are no aberrations in the decision-making process of the neural network.

We are going to analyze the Neural Networks outputs from a set of observations. Inputs of the Neural Network are from $\mathbb{R}^3$: relative position of the set point ($\mathbb{R}^2$) and its heading ($\mathbb{R}$). As we cannot analyses every value of this continuous space, we are going to discretize it.

We consider the three-dimensional set $[-X_{\max}, X_{\max}] \times [-Y_{\max}, Y_{\max}] \times [-\pi, \pi]$ discretized into a grid by a constant vector $\Delta\text{step} = (\Delta x,\ \Delta y,\ \Delta\theta)$.

As the output of the Neural Network is only 2 scalars, drone operators have only two values to analyze.

1) Visualisation of the NN outputs

We developed a custom visualization tool for analyzing the neural network's decision-making process.

For each discretized heading, we generated a chart depicting the network's decision based on the drone's relative position to its set points. The drone is oriented with the chosen discretized heading. In cell [0, 0], we present the decision made when the drone is on its set points. In cell [0, 1000], we

illustrate the decision made when the drone is 1000 meters ahead of its set point.

a) Speed Norm Analysis

We used a heat map (Fig. 7) to visualize the speed magnitude ordered. Darker cells indicate slower drone speeds. Our analysis revealed that when the drone is ahead of the swarm position, it predominantly uses the slowest speed available. In contrast, when the drone is behind its set position, it prioritizes the highest speed to catch up. The chosen speed when the drone is on its side is more intricate and depends on the drone's orientation. These observed behaviors are consistent with the drone effectively reaching its set position. Observations are symmetrical, yet actions are expected to be symmetrical; the asymmetry in the decision process therefore reveals an inconsistency when the drone operates beyond a given range.

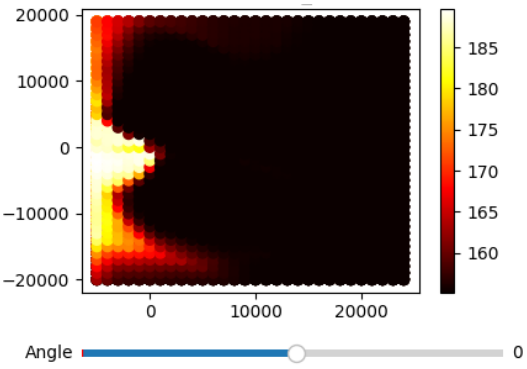


Fig. 7 Heat map of the chosen speed when the drone is going in the same direction as the swarm. [0,0] is the set position.

b) Heading Analysis

We used a 2D field of arrows to visualize the heading chosen by the Neural Network. In those chart, the set point is going straight from left to right. The drone's orientation is determined by the chosen discretized heading, indicated by the grey arrow. The relative heading chosen by the Neural Network is also denoted by an arrow.

This visualization enabled us to approve the discretized neural network's behavior, as we did not detect any discontinuities. However, we identified inconsistencies outside the network's domain of training, as shown in Figure 8. Specifically, when y exceeds 20km, the network chooses to maintain an almost straight course rather than moving closer to its set position.

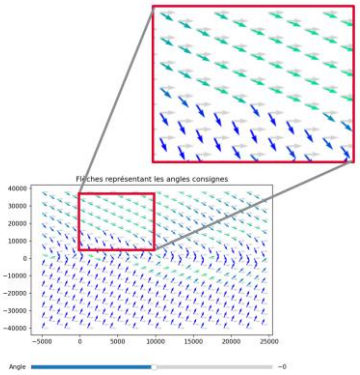


Fig. 8. Arrow field of the chosen heading when the drone is going in the same direction as the swarm. [0,0] is the set position. The red rectangle highlights an area where the Neural Network doesn't choose the optimal heading. When $y > 20\,000$ it chooses to go straight (green arrow) instead of getting closer to the set position (blue arrow)

c) Conclusion

Thanks to those two interactive charts, we were able to have a deep understanding of the model outputs. It highlighted the fact that the Neural Network is consistent but it has one flaw. The Neural Network is unable to generalize when out of a given bound. To address this issue, we updated the allocation map of SUPERVISED RL. When out of bound, the proportional controller is going to overtake the control of the drone. Otherwise, the Neural Network was validated by a drone operator when it operates within the specified range.

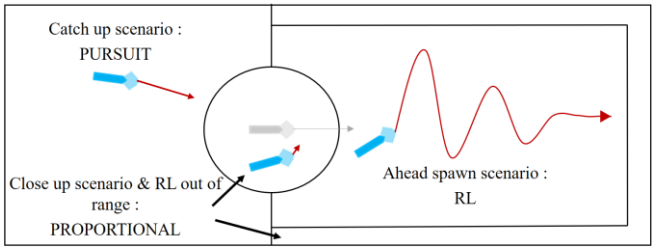


Fig. 8 Updated SUPERVISED RL algorithm's allocation given the RL flaws.

d) Neural Network discretization

Even though we have conducted an in depth analysis of the Neural Network on a discrete space. We are unable yet to confirm that there no aberration or discontinuities when using the Neural Network on a continuous space of observation. Indeed, the step Δstep used to discretize the observation is large in order to limit the number of point to analyze manually. To solve this issue, we are going to use a grid instead of the raw neural network function.

We are considering the 3 dimension grid Grid_{NN} of shape (N_1, N_2, N_3) such that

$$\text{Grid}_{NN}[i,j,k] = NN\left(f_{preprocess}\left(\text{observation}_{i,j,k}\right)\right)$$

With $\text{observation}_{i,j,k} = (i \times \Delta x,\ j \times \Delta y,\ k \times \Delta\theta)$

We won't use the formula Eq.1 to calculate an action given an observation . Instead, we will use Grid_{NN} with the following formula:

For a given observation vector, we consider $\widehat{\text{observation}}_{i,j,k}$ which represents its rounded value by Δstep . Each element is rounded according to the corresponding value in the Δstep vector. i, j, k are the quotients obtained by dividing each value of the observation vector by Δstep.

We will consider:

$$\text{action}_{\text{discrete}}(\text{observation}) = \text{Grid}_{NN}[i,j,k] = \text{action}\left(\widehat{\text{observation}}_{i,j,k}\right)$$

e) Discretized model performances

We tested the discretized reinforcement learning model (DISCRETE RL) performances on the Spawn Ahead Scenario defined earlier to assess any potential loss in performance compared to the original continuous Neural Network (CONT. RL).

DISCRETE RL is able performs better than the raw neural network. It respects the direction constraint and is faster to meet its set position. This improvement can be attributed to the discretization of the observation as the drone takes the same decision across multiple time steps making it more consistent.

	CONT. RL	DISCRETE RL
Average approach speed (m/s)	117	**120**

C. Ensuring convergence with safeguards

We propose another method to ensure that our model is always able to meet his set position while benefiting from the RL faster convergence speed.

If the input dimension of the observation is too large, a manual local analysis would be too time consuming. Moreover, we want to add extra safety features to our algorithmic chain as we are working with critical system. The system reliability cannot rely only on the drone operator analysis as human mistakes are possible.

To do so, we are going to implement safeguards to our algorithmic chain to ensure convergence on the set position regardless of the reinforcement learning decisions.

a) Safeguards

We added two safeguards to guarantee convergence to the set position while complying with the direction constraint:

- **Direction Safeguard**: If the drone is close from breaking the direction constraint, the proportional controller temporarily override the RL model until it is more aligned with the swarm's direction.
- **Speed Safeguard**: the drone's speed is set to minimum when it is in front of its set position and to maximum when it is lagging behind.

We also changed the allocation pattern of SUPERVISED RL. The bottom of the zone originally allocated to RL is switch to PROPORTIONAL in order to form a cone. This cone shape forces the drone to meet the set position when it is on its sides.

Moreover, SUPERVISED RL is also going to use the Safeguards when using RL algorithm.

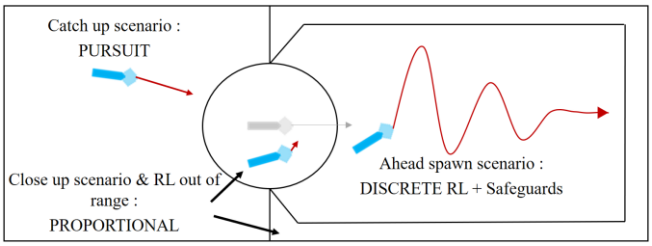


Fig. 9 Updated SUPERVISED RL algorithm's allocation to ensure convergence on the set position

b) Results

	DISCRETE RL	Random + Safeguards	DISCRETE RL + Safeguards	PROPORTIONAL
Worst case time to meet the set position (s)	186	526	189	868

Safeguards allows the drone to regain its set position no matter which decision is taken. We tested an algorithm choosing random heading over 2000 iterations of the Spawn ahead scenario. It met his set positions on 100% of the episodes thanks to the Safeguards algorithm. It was still slower than the other algorithms.

During the worst case of those 2000 iterations, Random + Safeguards took 526 seconds to meet his set position which is comparable with the PROPORTIONAL algorithm performance.

We trained a Reinforcement Learning model with the safeguards. The training was 10% faster as the model dictates only the heading.

We had the following performances across 500 episodes of the Spawn ahead scenario.

	CONT. RL	PROPORTIONAL	Discrete RL	Discrete RL + Safeguards	Random + Safeguards
Average approach speed (m/s)	117	68	**120**	107	61

The "Discrete RL + Safeguards" algorithm is unable to be as fast as the other RL models due to non-optimal speed constraints, particularly when the drone is on the side of its set position. However, it ensures that the drone will reach it. Moreover, it outperforms PROPOTIONAL, all while complying with the direction constraint.

Thanks to those precise analyses, the SUPERVISED RL algorithm uses the allocation pattern described in Fig. 10. It uses Discrete RL to have an explainable algorithm. It also uses Safeguards to ensure task completion and compliance with the direction constraint.

D. Ensuring the generalisation capabilities of RL

In real life environment, drones perform complex and unpredictable maneuvers. We couldn't train our model on every possible trajectory. Yet, our model was able to generalize and meet a swarm in carrying out any trajectory as we trained on the correct subset of scenario.

A full trajectory is a complex movement. But we can break it down into simpler maneuvers and train our Reinforcement Learning model on those simpler maneuvers.

As a trajectory is made out of straight lines and turns, we trained our model on scenarios where the swarms performed a straight line with a slight curb or a turn with a random turn radius. To assess the generalization capabilities of the reinforcement learning model, we trained and tested it across two distinct environments. The model was trained on trajectory's segments and tested on full trajectories.

We compared the proportional controller against the SUPERVISED RL model with all the improvement that have been implemented:

- **Discrete RL**: to ensure that we an explainable Neural Network.
- **Safeguards** so that the drone comply with the constraints.
- **Allocation patterns**: RL is only used when it brings better performances.

Those two algorithms are able to respect the direction constraints while ensuring the drone to meet his set position. They are also explainable thanks to in-depth analyses. We tested the two different algorithms on 500 episodes. The initialization was the same as the training scenario but the swarm performs a much more complex trajectory. It performs several random turns instead of just one. We ended the simulation once the drone is 1500 meters away from his set position as the SUPERVISED RL will also use the proportional controller at this point.

We found out the SUPERVISED RL algorithm is faster at joining the set position. Using the Reinforcement Learning only where it brings better performances led to high performance algorithm that complies with the use case constraints.

	PROPORTIONAL	SUPERVISED RL
Average approach speed (m/s)	77	**85**

IV. CONCLUSION

Our approach enhances the reliability of decision algorithm based on reinforcement learning and AI. It successfully addresses the major challenges associated with reinforcement learning: generalization, consistency, and explainability. This enables us to leverage the superior performance of reinforcement learning in critical systems. However, our discretization methodology may not be as effective for tasks involving high-dimensional observation inputs as the observation space might be too large to conduct a local analysis. Moreover, integrating a supervision algorithm and safeguards to ensure task completion is not straightforward across all use cases.

ACKNOWLEDGMENT

The authors would like to thank the engineers who were involved in this project for the quality of their contributions.

REFERENCES

[1] Bagnell, J. A., & Schneider, J. G. (2001, May). Autonomous helicopter control using reinforcement learning policy search methods. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)* (Vol. 2, pp. 1615-1620). IEEE.

[2] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

[3] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... & Hassabis, D. (2017). Mastering the game of go without human knowledge. nature, 550(7676), 354-359.

[4] Bois, J., Puig, A., Rullière, L., Teboul, Y., Ossola, M., Kotenkoff, A., & Formoso, M. (2022, November). Heterogeneous swarming for collaborative combat using Multi-agent Deep Reinforcement Learning. In Conference on Artificial Intelligence for Defense.

[5] Samek, W., Wiegand, T., & Müller, K. R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*.

[6] Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., & Zhu, J. (2019). Explainable AI: A brief survey on history, research areas, approaches and challenges. In Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II 8 (pp. 563-574). Springer International Publishing.

[7] Puiutta, E., & Veith, E. M. (2020, August). Explainable reinforcement learning: A survey. In International cross-domain conference for machine learning and knowledge extraction (pp. 77-95). Cham: Springer International Publishing.

[8] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.

[9] Zheng, S., Song, Y., Leung, T., & Goodfellow, I. (2016). Improving the robustness of deep neural networks via stability training. In Proceedings of the ieee conference on computer vision and pattern recognition (pp. 4480-4488).

[10] Weng, T. W., Dvijotham, K. D., Uesato, J., Xiao, K., Gowal, S., Stanforth, R., & Kohli, P. (2019, September). Toward evaluating robustness of deep reinforcement learning with continuous control. In International Conference on Learning Representations.

[11] Cobbe, K., Klimov, O., Hesse, C., Kim, T., & Schulman, J. (2019, May). Quantifying generalization in reinforcement learning. In International conference on machine learning (pp. 1282-1289). PMLR.

[12] Wang, K., Kang, B., Shao, J., & Feng, J. (2020). Improving generalization in reinforcement learning with mixture regularization. Advances in Neural Information Processing Systems, 33, 7968-7978.

[13] Lee, K., Lee, K., Shin, J., & Lee, H. (2019). Network randomization: A simple technique for generalization in deep reinforcement learning. arXiv preprint arXiv:1910.05396.

[14] Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: Algorithms and theory. IEEE Transactions on automatic control, 51(3), 401-420.

[15] Mandi, Z., Abbeel, P., & James, S. (2022). On the effectiveness of fine-tuning versus meta-reinforcement learning. *arXiv preprint arXiv:2206.03271*.

[16] Farama Foundation. (2023). Handling time limits. Retrieved from https://gymnasium.farama.org/tutorials/gymnasium_basics/handling_time_limits/

[17] Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018, July). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In International conference on machine learning (pp. 1861-1870). PMLR.

[18] Scharf, L. L., Harthill, W. P., & Moose, P. H. (1969). A comparison of expected flight times for intercept and pure pursuit missiles. IEEE Transactions on Aerospace and Electronic Systems, (4), 672-673.

[19] Coulter, R. C. (1992). Implementation of the pure pursuit path tracking algorithm (pp. 92-01). Carnegie Mellon University, The Robotics Institute.

V. ANNEX

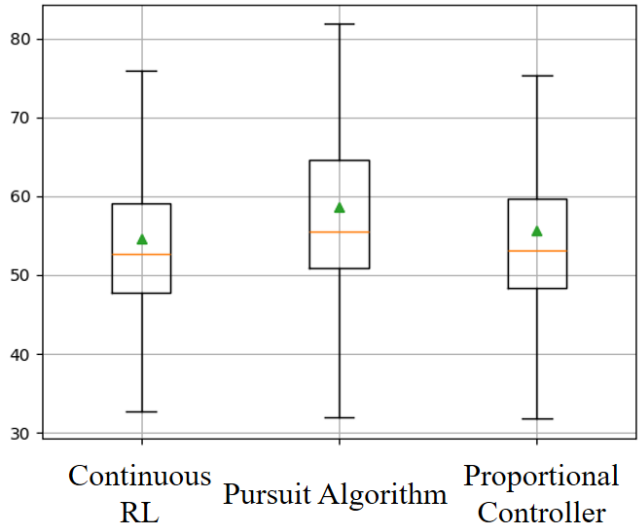


Figure 10 Boxplot of the average approach speed of the different algorithms across 500 catch up episode.

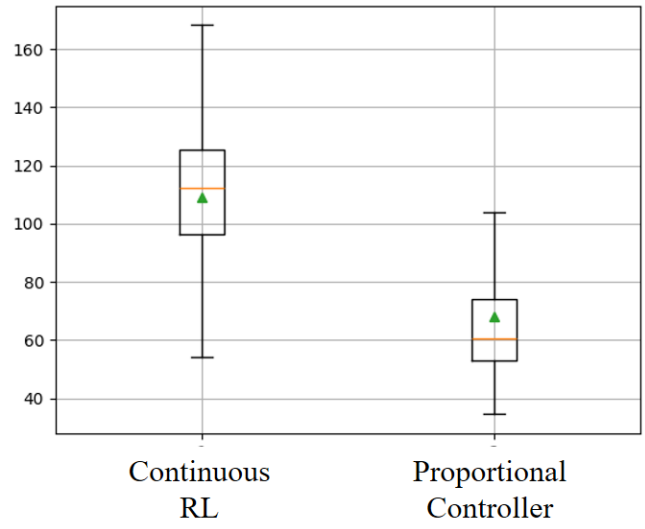


Figure 11 Boxplot of the average approach speed of the different algorithms across 500 Spawn ahead episodes

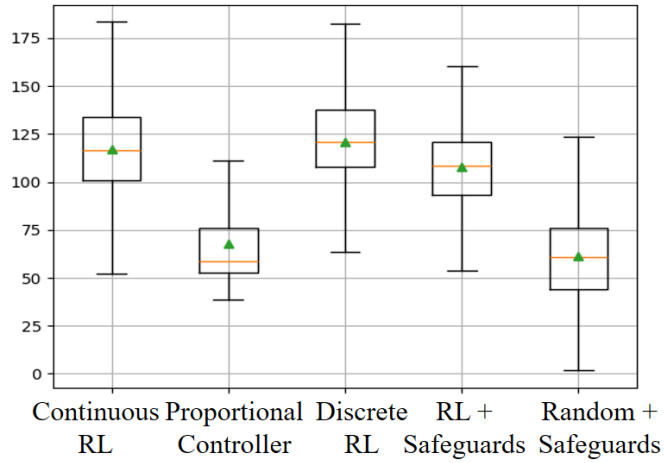


Figure 12 Boxplot of the average approach speed of the different algorithms across 500 Spawn ahead episodes

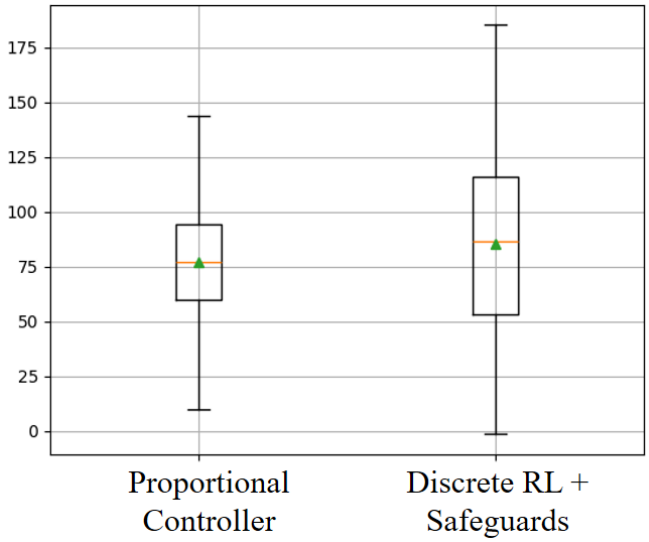


Figure 13 Boxplot of the average approach speed of the different algorithms across 500 full trajectory episodes.

Guidage de drone pour la triangulation à N-vues basé sur l'apprentissage par renforcement multi-agents

Timothée Gavin*†‡, Murat Bronz† and Simon Lacroix‡
*IAS, Thales LAS, Rungis, France
†Dynamic Systems, OPTIM, Fédération ENAC ISAE-SUPAERO ONERA, Université de Toulouse, Toulouse, France
‡LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
timothee.gavin@thalesgroup.fr , murat.bronz@enac.fr , simon.lacroix@laas.fr

***Abstract*—Cet article présente une nouvelle approche pour le contrôle d'une flotte de drones qui peuvent suivre la position d'une cible volante à l'aide de caméras omnidirectionnelles embarquées. Les drones utilisent l'apprentissage par renforcement multi-agents (MARL) pour apprendre des stratégies décentralisées qui optimisent leur formation et leur mouvement autour de la cible, en minimisant l'incertitude sur la position triangulée. Nous définissons une fonction de récompense qui encourage les suiveurs à minimiser la trace de la matrice de covariance de la position triangulée, dérivée d'un modèle analytique de propagation de l'incertitude. Nous utilisons Multi-Agent PPO (MAPPO), une extension de la méthode Proximal Policy Optimization (PPO) au domaine multi-agents, pour entraîner les modèles à l'aide de cette fonction de récompense commune qui favorise une bonne configuration et permet d'éviter les collisions. Nous validons notre approche en simulation et en vol réel, démontrant son efficacité et son potentiel dans l'amélioration de la coordination multi-drones autonomes pour un suivi de cible précis.**

***Index Terms*—Apprentissage par renforcement, multi-agents, PPO, triangulation**

I. Introduction

Dans cet article, nous cherchons à faire les premiers pas vers une nouvelle approche pour le contrôle d'une flotte de drones qui peuvent suivre l'emplacement d'une ou de plusieurs cibles volantes à l'aide de caméras embarquées. L'algorithme de guidage doit optimiser le placement des drones afin de minimiser l'incertitude de la triangulation des cibles suivies, même lorsqu'elles se déplacent avec des trajectoires imprévisibles.

L'approche proposée utilise l'apprentissage par renforcement multi-agent (ou Multi-Agent Reinforcement Learning (MARL) en anglais) pour positionner de manière optimale plusieurs drones afin de trianguler la position d'une cible unique à l'aide de caméras omnidirectionnelles embarquées. Nous utilisons Multi-Agent PPO (MAPPO) [1], une extension de l'algorithme Proximal Policy Optimization (PPO) [2] au domaine multi-agent, pour entraîner des agents à suivre des stratégies décentralisées en utilisant une fonction de récompense commune spécialement définie qui encourage une bonne formation autour de la cible afin de minimiser l'incertitude dans la position triangulée. Les modèles entraînés sont validés en simulation et le comportement résultant est démontré dans des expériences en vol réel.

Nous étudions le MARL par rapport aux méthodes d'optimisation traditionnelles car nous espérons de meilleures performances dans la résolution des problèmes de triangulation impliquant un grand nombre de drones. Les méthodes d'optimisation traditionnelles, même en utilisant des heuristiques, sont lentes et coûteuses en temps de calcul pour les problèmes à grande échelle, ce qui n'est pas adapté à un algorithme de guidage embarqué. Bien que l'entrainement d'un agent RL soit une opération coûteuse en temps de calcul, il peut rapidement générer des solutions en ligne.

L'article est organisé comme suit. Après avoir passé en revue les travaux connexes pertinents sur le suivi des drones utilisant des drones dans la Section II, et présenté le contexte existant en matière de MARL et de triangulation dans la Section III, nous présentons notre approche de guidage de drone basé sur le MARL appliqué à la triangulation à N-vues dans la Section IV. Nous détaillons la configuration des simulations et leurs résultats dans la Section V. Enfin, nous décrivons les expériences en vol réel dans la Section VI.

II. Travaux Associés

La triangulation est le processus d'estimation de la position 3-D d'un point à partir de ses projections sur deux ou plusieurs images prises à partir de points de vue différents. Cette opération peut être réalisée en calculant l'intersection des rayons de projection associés à chaque image, et en minimisant l'erreur de reprojection. Cependant, la triangulation est sensible aux erreurs d'étalonnage de la caméra, ainsi qu'au bruit et aux valeurs aberrantes dans les points de l'image. Une technique courante pour optimiser le placement des caméras et l'estimation de la position de la cible dans la triangulation à N-vues est *l'ajustement de faisceau* [3]. Il s'agit d'une méthode d'optimisation des moindres carrés non linéaire qui affine simultanément les paramètres intrinsèques et extrinsèques de la caméra, en minimisant la somme des carrés des erreurs de reprojection sur tous les points d'image et toutes les vues. Cette méthode est largement utilisée pour la recontruction 3D de scènes à partir d'images calibrées prises

par différentes caméras. Cependant, le coût de calcul et les besoins en mémoire de l'ajustement des faisceaux augmentent de façon superlinéaire avec le nombre de caméras [4].

Le suivi est le processus d'estimation de la position et de l'orientation d'un objet en mouvement dans le temps à partir d'une séquence d'images. Des techniques telles que les filtres de Kalman et la régression polynomiale non linéaire sont couramment utilisées pour le suivi des drones [4]. La majorité des recherches sur les robots mobiles coopératifs pour l'observation de cibles mobiles se concentrent sur des cibles terrestres se déplaçant dans un plan 2-D [5]–[7]. Certaines études ont proposé l'utilisation de caméras embarquées pour le suivi en temps réel et la localisation en 3-D de plusieurs drones [8]. Cependant, le suivi de drones à l'aide d'une caméra embarquée sur un autre drone reste un domaine relativement peu exploré dans la littérature, car il pose plus de difficultés que le suivi à l'aide de systèmes de caméras fixes.

L'apprentissage par renforcement (Reinforcement Learning (RL) en anglais) est un paradigme d'apprentissage automatique qui permet à un agent d'apprendre de ses propres actions en recevant des récompenses et des punitions. Le RL a été appliqué à divers problèmes impliquant le placement optimal de capteurs et à des algorithmes de guidage pour les drones et les flottes de drones. Le RL a été utilisé pour mesurer la profondeur des scènes intérieures à l'aide de plusieurs caméras, l'agent apprenant à sélectionner les meilleures positions et orientations des caméras afin de minimiser l'erreur d'observation de la profondeur [9]. Alternativement, une approche basée sur le RL pour la poursuite et l'évasion de drones, où l'agent apprend à suivre un drone cible en utilisant des données de capteurs et détecteur d'objet, a été présentée dans [10]. Ces travaux se concentrent principalement sur le RL pour un seul agent, où un agent interagit avec l'environnement de manière indépendante. MARL est une branche de RL qui traite de plusieurs agents qui coopèrent ou sont en concurrence les uns avec les autres dans un environnement partagé. Le MARL a été utilisé pour apprendre la configuration optimale et le déplacement d'une flotte de drones pour la triangulation, en maximisant la couverture et la diversification des vues et en optimisant les communications [7], [11], mais encore une fois, ces études se concentrent sur l'observation de cibles terrestres se déplaçant dans un plan à deux dimensions. À notre connaissance, aucune étude récente n'a abordé le placement optimal d'une flotte de drones utilisant le MARL pour suivre des objets volants à l'aide de caméras embarquées.

III. Méthodes et outils

A. Processus de décision markoviens décentralisés partiellement observables (Dec-POMDP)

Nous considérons une tâche multi-agents entièrement coopérative qui peut être décrite comme un processus de décision markovien décentralisé et partiellement observable [12] défini par $(n, S, A, \Omega, T, O, R, \gamma)$, où n le nombre d'agents, S est l'espace d'état, $A = A_1 \times \ldots \times A_N$ et $\Omega = \Omega_1 \times \ldots \times \Omega_N$ sont l'ensemble des actions conjointes et des observations conjointes avec chaque A_i et Ω_i étant les ensembles d'actions locales et d'observations locales de l'agent i, $T : S \times A \times S \to [0, 1]$ est la fonction de probabilité de transition entre états, $O : S \times A \times \Omega \to [0, 1]$ est la fonction de probabilité d'observation sur les états, $R : S \times A \to \mathbb{R}$ est une fonction de récompense sur les transitions d'états, et enfin $\gamma \in [0, 1]$ est le facteur d'actualisation.

À chaque pas de temps t, chaque agent i choisit une action $a_{i,t} \in A_i$ sur la base de son historique d'observations locales $o_{i,1:t}$, et reçoit une observation locale $o_{i,t+1} \in \Omega_i$ basée sur l'état résultant s_{t+1}. L'action conjointe $a_t = (a_{1,t}, \ldots, a_{N,t})$ détermine la récompense immédiate $r_t = R(s_t, a_t)$ et l'état suivant s_{t+1} selon la probabilité de transition $T(s_t, a_t, s_{t+1}) = \Pr(s_{t+1}|s_t, a_t)$. L'objectif est de trouver une politique $\pi_\theta(a_{i,t}|o_{i,1:t})$ paramétrée par θ qui produise une action $a_{i,t}$ à partir de l'historique d'observations locales $o_{i,1:t}$, qui maximise le gain attendu $J(\theta) = \mathbb{E}[\sum_{t=0}^{T} \gamma^t r_t]$. La fonction de valeur d'état $V^{\pi_\theta}(s) = \mathbb{E}_{\pi_\theta}[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | S_t = s]$ est le gain attendu en partant de l'état s et en suivant la politique π_θ par la suite. La fonction de valeur action-état $Q^{\pi_\theta}(s, a) = \mathbb{E}_{\pi_\theta}[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | S_t = s, A_t = a]$ est le gain attendu en partant de l'état s, en prenant l'action a, et en suivant la politique π par la suite.

B. Algorithmes d'apprentissage par renforcement à un seul agent

La plupart des méthodes d'apprentissage par renforcement peuvent être divisées en deux groupes : les méthodes basées sur la valeur et les méthodes basées sur la politique. Les méthodes basées sur la valeur, telles que Deep Q-Learning (DQN) [13], utilisent des réseaux neuronaux profonds pour estimer les fonctions de valeur et dérivent la politique optimale π^* à partir des fonctions de valeur optimisées en choisissant pour chaque état s l'action qui maximise la valeur de l'action : $\pi^*(s) = \arg\max_a Q^*(s, a)$. Les méthodes basées sur la politique, quant à elles, paramètrent directement la politique $\pi_\theta(a|s)$ en fonction de l'état et de l'action. Les méthodes basées sur le gradient de la politique sont une sous-classe des méthodes basées sur la politique qui mettent à jour les paramètres de la politique θ en suivant le gradient du gain attendu : $\nabla_\theta J(\theta) = E_\pi[\nabla_\theta \log \pi_\theta(a|s) Q^\pi(s, a)]$. L'utilisation du gain total de chaque épisode pour estimer le gradient entraîne une forte variance dans les estimations du gradient. Pour réduire cette variance, les méthodes Actor-Critic ont été introduites [14]. Ces méthodes maintiennent un approximateur explicite séparé (la Critique) pour estimer la fonction de valeur, qui est utilisée comme base de référence pour calculer la fonction Advantage. La fonction Advantage, $A^\pi(s, a) = Q^\pi(s, a) - V\pi(s)$, mesure l'amélioration d'une action a par rapport à l'action moyenne à l'état s dans le cadre de la politique π. Cela permet de réduire la variance des estimations du gradient, ce qui conduit à un apprentissage plus stable. PPO [2] est une évolution des méthodes de gradient de politique qui cherche à mettre à jour la politique d'une manière qui évite les changements importants et abrupts qui pourraient déstabiliser le processus d'apprentissage, garantissant ainsi un apprentissage stable et efficace.

C. Apprentissage par renforcement multi-agents

Les méthodes RL à un seul agent échouent souvent dans des contextes multi-agents en raison du fléau de la dimension et de la non-stationnarité. Récemment, les approches MARL ont abordé ces problèmes avec l'approche CTDE (Centralized Training, Decentralized Execution). En entraînant les agents de manière centralisée afin d'exploiter les informations globales, puis en exécutant les politiques apprises de manière décentralisée, CTDE garantit robustesse et capacité de montée à l'échelle. Plusieurs algorithmes MARL ont été développés dans le cadre du CTDE, y compris des méthodes basées sur la valeur comme QMIX [15] et VDAC [16], et des méthodes de gradient de politique comme COMA [17] et MADDPG [18]. PPO s'est révélée très prometteuse dans le domaine multi-agent: Multi-Agent PPO (MAPPO) [1], une extension de PPO dans le cadre CTDE, utilise un critique centralisé et des acteurs décentralisés et a démontré des performances supérieures dans diverses tâches complexes multi-agents par rapport à d'autres algorithmes MARL de l'état de l'art [1].

D. Triangulation linéaire à N-vues

La triangulation linéaire est une méthode permettant d'estimer les coordonnées 3-D d'un point à partir de ses projections 2-D sur deux ou plusieurs images prises par des caméras différentes. Le principe de la triangulation linéaire est basé sur le modèle du sténopé, qui relie le point 3-D $\mathbf{X}$ et sa projection 2-D $\mathbf{x}$ en coordonnées homogènes par la matrice 3×4 de projection de la caméra $\mathbf{P}$ sous la forme $\mathbf{x} = \mathbf{PX}$. La matrice de projection de la caméra $\mathbf{P}$ englobe à la fois les paramètres intrinsèques (tels que la distance focale et le centre optique) et les paramètres extrinsèques (rotation et la position) d'une caméra.

La triangulation linéaire résout un système linéaire surdéterminé de la forme $\mathbf{AX} = 0$, qui dans le cas d'une triangulation à N-vues, avec n de ces points d'image et matrices de projection donne :

$$\begin{bmatrix} u_1\mathbf{P}_1^{3T} - \mathbf{P}_1^{1T} \\ v_1\mathbf{P}_1^{3T} - \mathbf{P}_1^{2T} \\ u_2\mathbf{P}_2^{3T} - \mathbf{P}_2^{2T} \\ v_2\mathbf{P}_2^{3T} - \mathbf{P}_2^{2T} \\ \vdots \\ u_n\mathbf{P}_n^{3T} - \mathbf{P}_n^{1T} \\ v_n\mathbf{P}_n^{3T} - \mathbf{P}_n^{2T} \end{bmatrix} \begin{bmatrix} \lambda x \\ \lambda y \\ \lambda z \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{1}$$

où $\mathbf{X} = (\lambda x, \lambda y, \lambda z, \lambda)^T$, λ étant un facteur d'échelle inconnu, et u_i, v_i et P_i^{jT} sont les coordonnées de l'image et la $j$ème ligne de la matrice de projection P_i de la $i$ème caméra. Pour trouver $\mathbf{X}$, on trouve une solution X non nulle qui satisfait $\mathbf{AX} = 0$. Dans le cas de mesures bruitées, comme les n rayons définis par les points de projection 2-D ne se croisent pas en un seul point, le système n'a pas de solutions non nulles, et le problème est transformé en un problème de minimisation, généralement résolue avec des techniques des moindres carrés [19].

IV. Détails de l'approche

A. Présentation du scénario

Notre problème consiste en plusieurs drones, appelés ci-après "suiveurs", qui poursuivent un seul drone, la "cible", pour suivre sa position, en utilisant des capteurs embarqués tels que des caméras. Les suiveurs s'efforcent d'organiser la formation de leur flotte de sorte que l'incertitude de la position triangulée soit minimale.

Les suiveurs et les mouvements de la cible sont omnidirectionnels, nous n'avons pas mis en œuvre de modèle de vol de drone dans cette étude. Nous supposons que la cible à une vitesse de vol constante. Les drones volent dans une arène carrée sans aucun obstacle. Ni les suiveurs ni la cible ne peuvent sortir de l'arène : leurs actions sont limitées pour rester à l'intérieur des frontières de l'arène. Les collisions entre suiveurs, ou entre un suiveur et la cible, entraînent un échec. Le fait de s'écraser au sol entraîne également un échec. L'orientation des drones n'est pas prise en compte dans cette étude, on suppose que tous les drones sont alignés avec leur cap parallèle à la direction positive de l'axe X dans le système de coordonnées global.

Nous supposons que les suiveurs peuvent différencier les autres suiveurs de la cible. Nous n'avons pas pris en compte l'observabilité partielle dans cette étude. Nous faisons l'hypothèse forte que chaque suiveur connaît les positions 3D exactes dans le monde, sans bruit, des autres suiveurs de la flotte et de la cible. Cependant, les actions entreprises par les suiveurs sont décentralisées, ce qui signifie qu'ils ne connaissent pas les actions entreprises par les autres membres de la flotte.

La caméra embarquée sur les suiveurs est supposée être omnidirectionnelle, sans champ de vision restreint et non affectée par les occultations. Le centre de la caméra coïncide avec l'emplacement 3D des suiveurs. Nous supposons également que les paramètres intrinsèques de la caméra n'introduisent aucune distorsion dans les mesures. Plutôt que de mesurer l'emplacement de la projection d'un point du monde sur un plan de caméra 2D, la caméra omnidirectionnelle mesure simplement les angles polaires et azimutaux dans un repère sphérique centré sur la position du suiveur. Au vu de ces simplifications, l'incertitude de la triangulation n'est que le résultat de l'incertitude de la mesure le long des angles polaire

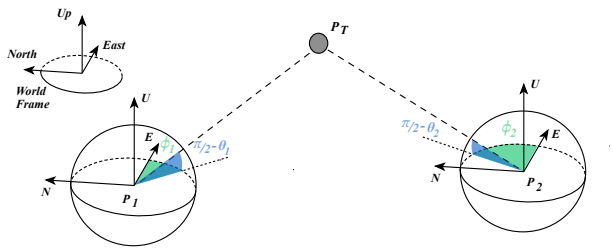


Fig. 1. Représentation des caméras omnidirectionnelles dans un repère sphérique. Les angles polaires θ sont représentés par rapport au plan horizontal pour faciliter la visualisation.

et azimutal dans chaque repère de suiveur. C'est ce qui a conduit à l'élaboration de la fonction de récompense décrite à la Section IV-D.

B. Algorithme d'apprentissage par renforcement multi-agents

Dans cette étude, nous utilisons l'algorithme Multi-Agent Proximal Policy Gradient (MAPPO) en raison de sa simplicité d'implémentation, de sa capacité à opérer dans des espaces d'état et d'action continus et de son efficacité démontrée dans diverses tâches MARL [1]. Nous supposons que tous les agents sont homogènes, ce qui nous permets de partager les modèles entrainés, accélérant ainsi le processus d'apprentissage et maximisant les informations extraites de chaque interaction avec l'environnement. MAPPO fonctionne dans un cadre CTDE, il utilise un critique centralisé et des acteurs décentralisés : alors que tous les agents sont régis par une politique commune et entraînés de manière centralisée, ils agissent de manière indépendante sur la base de leurs observations locales à chaque pas de temps, créant ainsi un système décentralisé. Cette approche est nécessaire car nous n'avons pas abordé le problème de la longueur variable de la représentation de l'état en fonction du nombre de suiveurs dans l'environnement.

C. Représentation de l'état

L'espace d'état, l'espace d'observation et l'espace d'action sont continus. Chaque position 3-D du suiveur en coordonnées cartésiennes globales est encodée dans un vecteur 3-D $\mathbf{p_i}$. La position 3-D de la cible est $\mathbf{p_T}$. Les n suiveurs sont ordonnés de telle sorte que $\mathbf{p_i}$ est toujours la position du $i$ème suiveur. L'état de l'environnement est le nuplet ordonné $\mathbf{s} = (\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_n}, \mathbf{p_T})$ et constitue l'entrée du réseau de neurone critique centralisé utilisé dans l'algorithme MAPPO. Pour chaque suiveur i, son observation de l'environnement est codée dans le nuplet ordonné $\mathbf{o_i} = (\mathbf{p_i}, \mathbf{p_1}, \ldots, \mathbf{p_{i-1}}, \mathbf{p_{i+1}}, \ldots, \mathbf{p_n}, \mathbf{p_T})$ et constitue l'entrée du réseau de neurone acteur décentralisé.

En ce qui concerne l'espace d'action, comme expliqué dans la section IV-A, le mouvement des suiveurs est holonome. À chaque étape, les agents naviguent dans une sphère d'un rayon de 20 cm centrée sur leur emplacement précédent (cette valeur est directement héritée de l'environnement de simulation CrazyRL [20] que nous avons adapté pour mettre en œuvre notre environnement de simulation, voir la section V). Ce rayon est déterminé par la vitesse maximale de l'agent (environ 2m/s) divisée par la fréquence de contrôle (10Hz dans l'environnement de simulation). Pour calculer la position du prochain pas, le réseau acteur produit un point 3-D dans $[-1, 1]^3$, qui est ensuite mis à l'échelle de 20 cm.

D. Structure de la fonction de récompense

À chaque étape, chaque agent reçoit une récompense commune destinée à encourager les suiveurs à atteindre des positions autour de la cible où l'incertitude sur le résultat de l'algorithme de triangulation est minimale.

L'évaluation de l'incertitude du résultat d'un algorithme d'estimation peut se faire selon deux approches. Une approche *a posteriori* implique la mise en œuvre d'une méthode Monte Carlo : l'incertitude de la sortie est statistiquement obtenue à partir d'un échantillonnage aléatoire répété compte tenu d'une distribution d'entrée bruitée. Bien qu'elle soit facile à mettre en œuvre, elle nécessite un nombre important d'exécutions, ce qui peut prendre beaucoup de temps. La seconde méthode est une approche *a priori*, qui implique le calcul de relations analytiques à l'aide d'approximations linéaires pour décrire la façon dont l'incertitude se propage des entrées aux sorties. Cette deuxième approche ne nécessite qu'une seule exécution pour calculer l'incertitude de la sortie. C'est donc l'approche privilégiée pour notre fonction de récompense, puisqu'elle sera calculée à chaque étape d'interaction avec l'environnement.

1) Triangulation avec des caméras omnidirectionnelles: Dans notre cas simplifié avec des caméras omnidirectionnelles, au lieu de mesurer la position de la projection du point sur le plan de la caméra 2-D, nous mesurons l'angle polaire et l'angle azimutal dans le repère sphérique centré sur la position du suiveur. Pour chaque suiveur à l'emplacement $\mathbf{p} = (p_x, p_x, p_z)$ en coordonnées cartésiennes globales, les coordonnées cartésiennes globales d'un point $\mathbf{X} = (x, y, z)$ sont liées à ses coordonnées sphériques locales de la manière suivante :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = r \times \begin{bmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{bmatrix} + \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{2}$$

avec $\theta \in [0, \pi]$ l'angle polaire, et $\phi \in [0, 2\pi[$ l'angle azimutal. La distance radiale $r \in \mathbb{R}^+$, est inconnue dans le problème de triangulation. Définissons le vecteur suivant :

$$\mathbf{d_i} = \begin{bmatrix} d_{x,i} \\ d_{y,i} \\ d_{z,i} \end{bmatrix} = \begin{bmatrix} \sin\theta_i\cos\phi_i \\ \sin\theta_i\sin\phi_i \\ \cos\theta_i \end{bmatrix} \tag{3}$$

Ce vecteur, souvent appelé *cosinus directeurs*, définit la direction du point $\mathbf{X}$ dans le système de coordonnées cartésiennes locales du $i$ème suiveur à l'aide des coordonnées sphériques. Avec l'emplacement du suiveur $\mathbf{p_i}$, elles définissent une droite passant par $\mathbf{X}$. Dans le cas d'une triangulation avec deux caméras omnidirectionnelles, nous avons :

$$\begin{cases} \mathbf{X} = r_1 \times \mathbf{d_1} + \mathbf{p_1} \\ \mathbf{X} = r_2 \times \mathbf{d_2} + \mathbf{p_2} \end{cases} \tag{4}$$

Nous pouvons éliminer les distances radiales inconnues à l'aide d'un produit en croix afin d'obtenir trois équations pour chaque suiveur, très semblables à celles de la triangulation linéaire dans le cas des caméras à sténopé dans (1). Notez que la troisième équation est une composition linéaire des deux autres et qu'elle pourrait être supprimée.

$$\begin{cases} d_{y,i}(z - p_{z,i}) - d_{z,i}(y - p_{y,i}) = 0 \\ d_{z,i}(x - p_{x,i}) - d_{x,i}(z - p_{z,i}) = 0 \\ d_{x,i}(y - p_{y,i}) - d_{y,i}(x - p_{x,i}) = 0 \end{cases} \tag{5}$$

Nous obtenons, pour n suiveurs, un système linéaire surdéterminé de la forme $\mathbf{AX} = \mathbf{b}$, qui peut être résolu en

utilisant une méthode telle que la décomposition en valeurs singulières ou en utilisant des techniques de moindres carrés pour obtenir la valeur de $\mathbf{X}$.

$$\begin{bmatrix} \mathbf{A_1} & 0 & \dots & 0 \\ 0 & \mathbf{A_2} & \dots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \dots & \mathbf{A_n} \end{bmatrix} \begin{bmatrix} \mathbf{X}-\mathbf{p_1} \\ \mathbf{X}-\mathbf{p_2} \\ \vdots \\ \mathbf{X}-\mathbf{p_n} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{6}$$

with

$$\mathbf{A_i} = \begin{bmatrix} 0 & -d_{z,i} & d_{y,i} \\ d_{z,i} & 0 & -d_{x,i} \\ -d_{y,i} & d_{x,i} & 0 \end{bmatrix} \tag{7}$$

2) Propagation des incertitudes: Les équations de triangulation dans (6) ont la forme implicite :

$$f_\rho(\mathbf{X}, \mathbf{v}) = 0 \tag{8}$$

où f_ρ est paramétré par le vecteur de paramètres $\rho = (\mathbf{p_1}, \dots, \mathbf{p_n})$, $\mathbf{X}$ est le point 3-D triangulé en sortie de l'algorithme de triangulation et $\mathbf{v}$ est un vecteur d'entrées bruitées $\mathbf{v} = (\theta_1, \phi_1, \dots, \theta_n, \phi_n)$, régi par un bruit gaussien avec une matrice de covariance $\mathbf{\Sigma_v}$.

Dans ce cas, la matrice de covariance $\mathbf{\Sigma_X}$ de la sortie $\mathbf{X}$ est liée à la matrice de covariance $\mathbf{\Sigma_v}$ par [21], [22] :

$$\mathbf{J_X \Sigma_X J_X}^T = \mathbf{J_v \Sigma_v J_v}^T \tag{9}$$

où $\mathbf{J_X}$ et $\mathbf{J_v}$ sont les matrices jacobiennes des dérivées partielles de f_ρ par rapport, respectivement, à la sortie $\mathbf{X}$ et à l'entrée $\mathbf{v}$. La matrice de covariance de sortie peut alors être exprimée comme une fonction de la matrice de covariance d'entrée (qui est soit connue, soit présumée) par :

$$\mathbf{\Sigma_X} = \mathbf{J_X}^+ \mathbf{J_v} (\mathbf{\Sigma_v J_v}^T)(\mathbf{J_X}^+)^T \tag{10}$$

où $\mathbf{J_X}^+$ est la matrice pseudo-inverse de $\mathbf{J_X}$.

Dans notre cas, les jacobiens peuvent être exprimés analytiquement à l'aide de (6). Le vecteur d'entrée $\mathbf{v}$ peut être décomposé en $\mathbf{v} = (\mathbf{v_1}, \mathbf{v_2}, \dots, \mathbf{v_n})$ avec $\mathbf{v_i} = (\theta_i, \phi_i)$, alors $f_\rho(\mathbf{X}, \mathbf{v})$ peut être écrit :

$$f_\rho(\mathbf{X}, \mathbf{v}) = \begin{bmatrix} f_{\mathbf{p_1}}(\mathbf{X}, \mathbf{v_1}) & \dots & f_{\mathbf{p_n}}(\mathbf{X}, \mathbf{v_n}) \end{bmatrix}^T \tag{11}$$

avec chaque $f_{\mathbf{p_i}}(\mathbf{X}, \mathbf{v_i}) = [\mathbf{A_i} \cdot (\mathbf{X} - \mathbf{p_i})]^T$. D'où les matrices jacobiennes suivantes :

$$\mathbf{J_X} = \begin{bmatrix} \frac{\partial f_{\mathbf{p_1}}}{\partial \mathbf{X}} \\ \frac{\partial f_{\mathbf{p_2}}}{\partial \mathbf{X}} \\ \vdots \\ \frac{\partial f_{\mathbf{p_n}}}{\partial \mathbf{X}} \end{bmatrix} = \begin{bmatrix} \mathbf{A_1} \\ \mathbf{A_2} \\ \vdots \\ \mathbf{A_n} \end{bmatrix} \tag{12}$$

$$\mathbf{J_v} = \begin{bmatrix} \frac{\partial f_{\mathbf{p_1}}}{\partial \mathbf{v_1}} & 0 & \dots & 0 \\ 0 & \frac{\partial f_{\mathbf{p_2}}}{\partial \mathbf{v_2}} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\partial f_{\mathbf{p_n}}}{\partial \mathbf{v_n}} \end{bmatrix} \tag{13}$$

avec

$$\frac{\partial f_{\mathbf{p_i}}}{\partial \mathbf{v_i}} = \begin{bmatrix} \frac{\partial \mathbf{A_i}}{\partial \theta_i} \cdot (\mathbf{X} - \mathbf{p_i}) & \frac{\partial \mathbf{A_i}}{\partial \phi_i} \cdot (\mathbf{X} - \mathbf{p_i}) \end{bmatrix} \tag{14}$$

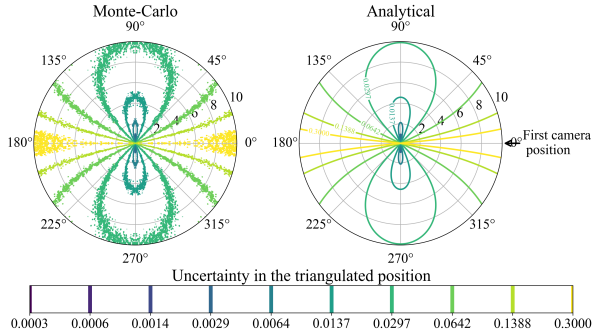


Fig. 2. Incertitude de la sortie triangulée sur l'axe X pour deux caméras dans le plan XY, l'une d'entre elle étant en $(x, y) = (10, 0)$.

3) Comparaison avec l'approche de Monte Carlo: Pour valider le modèle analytique précédent, nous avons comparé l'incertitude obtenue avec l'approche analytique aux résultats d'une approche Monte Carlo. Nous avons placé, en simulation, deux caméras à différents endroits d'une sphère autour d'un point cible. En supposant que les entrées sont soumises à un bruit gaussien, nous avons ajouté un bruit gaussien avec un écart type $(\sigma_{\theta_i}, \sigma_{\phi_i})$ aux mesures réelles et avons utilisé un algorithme de triangulation basé sur (6) et un solveur des moindres carrés pour calculer les positions de sortie estimées. Nous avons ensuite calculé l'incertitude statistique de la position de la cible pour chaque paire d'emplacements de caméra.

La Fig. 2 montre les résultats de la comparaison entre le modèle analytique proposé et les simulations de Monte Carlo pour différentes paires d'emplacements de caméra dans un plan. La première caméra est fixée à 10m de la cible le long de l'axe X, et la seconde caméra est positionnée uniformément dans le cercle autour de la cible sur le plan XY. Nous avons utilisé pour les deux caméras un écart-type d'entrée de $\sigma_{\theta_i} = \sigma_{\phi_i} = 0,003$ rad. Pour chaque paire d'emplacements de caméra, 100 exécutions de Monte Carlo ont été utilisées pour calculer l'incertitude statistique de la sortie. L'incertitude de la coordonnée x de $\mathbf{X}$ en fonction de la distance et de l'angle dans le plan XY de la deuxième caméra est représentée graphiquement. Les autres composantes ne sont pas présentées ici, mais la composante x est la plus illustrative, puisque la première caméra est alignée sur l'axe X. Il est facile de voir que les valeurs de Monte Carlo sont très proches des résultats analytiques.

4) Fonction de récompense: À chaque pas de temps, les suiveurs reçoivent la récompense suivante :

$$r_i = \begin{cases} \frac{1}{\sqrt{\mathrm{Tr}(\mathbf{\Sigma_X})}}, & \text{si } d_{i,target} > d_{threshold} \\ & \text{ou } d_{i,ground} > d_{crash} \\ & \text{ou } \forall j \neq i \ d_{i,j} > d_{crash} \\ -r_{too\_close} & \text{si } d_{crash} < d_{i,target} < d_{threshold} \\ -r_{penalty}, & \text{sinon} \end{cases} \tag{15}$$

Où $\mathrm{Tr}(\mathbf{\Sigma_X})$ désigne la trace de la matrice de covariance $\mathbf{\Sigma_X}$, c'est-à-dire la variance de la distance du vecteur $\mathbf{X}$

par rapport à sa moyenne. Cela encourage les suiveurs à atteindre des positions autour de la cible où l'incertitude sur le résultat de l'algorithme de triangulation est minimale. Il s'agit d'une récompense commune calculée à partir de la position de chaque suiveur, ce qui encourage la collaboration.

Si l'un des suiveurs entre en collision avec la cible, le sol ou un autre suiveur au cours d'un pas de temps donné (sur la base d'un seuil de distance d_{crash}), le suiveur reçoit une pénalité individuelle $(-r_{penalty})$. Cela encourage chaque suiveur à éviter les collisions.

Additionnellement, une faible pénalité individuelle $(-r_{too\_close} << -r_{penalty})$ est infligée lorsqu'un suiveur s'approche trop prêt de la cible (sur la base d'un seuil de distance $d_{threshold}$), pour encourager les suiveurs à se tenir à une distance de sécurité respectable de la cible. Sans cette pénalité, les suiveurs seraient encouragés à s'approcher au plus prêt de la cible pour réduire l'incertitude triangulée, jusqu'à frôler la collision. Cet ajout peut aussi être justifié par le fait que, dans la pratique, un drone qui suit une cible à l'aide d'une caméra ne pourrait pas le faire s'il était trop proche, en raison du champ de vision limité des caméras.

V. Expériences en simulation

Nous avons implémenté notre scénario en simulation en adaptant les environnements de simulation de CrazyRL [20] basés sur l'API standard de la Farama Foundation pour les environnements MARL, PettingZoo [23]. CrazyRL est une bibliothèque MARL Python qui fournit des environnements de simulation et des outils pour faire du MARL avec les drones Crazyflie 2.1, commercialisés par Bitcraze AB. Nous avons utilisé cette bibliothèque pour tester ultérieurement nos modèles RL entraînés lors de vols réels, comme présenté dans VI. Les environnements d'entrainement de CrazyRL sont très rapides, ce qui est nécessaire pour pouvoir entrainer nos agents dans un temps raisonnable, mais au détriment de la complexité, car le modèle dynamique du drone n'est pas pris en compte, comme spécifié dans IV-A. Les environnements de simulation existants de CrazyRL ont été fortement personnalisés pour s'adapter à la description de notre scénario, à notre nouvelle fonction de récompense et à l'ajout de fonctionnalités de "domain randomization" (les positions de départ de la cible et des suiveurs ont été rendus aléatoires à chaque exécution, une caractéristique non présente dans l'implémentation originale de CrazyRL mais nécessaire pour obtenir une meilleure généralisation).

Nos environnements de simulation et notre implémentation de MAPPO [1] ont été codés avec JAX [24], [25]. Contrairement aux processus d'apprentissage RL plus classiques, avec cette implémentation le simulateur est aussi porté sur le GPU, ce qui permet d'éliminer les délais introduits par les communications entre CPU et GPU et de tirer pleinement parti des fonctionnalités de parallélisme des GPUs pour accélérer le processus d'apprentissage par plusieurs ordres de grandeur.

Nos politiques sont paramétrées par un perceptron multicouche à deux couches avec 128 unités par couche. Le réseau acteur fait correspondre les observations de l'agent aux vecteurs de moyenne et d'écart-type d'une distribution gaussienne multivariable à laquelle on applique une transformation *tanh*, à partir de laquelle les actions sont échantillonnées dans l'espace continu. La transformation *tanh* est utilisée pour contraindre les actions en sortie dans un intervalle fini [26].

[1] fournit des recommandations de bonnes pratiques pour le choix des hyperparamètres pour l'entrainement des réseaux de neurones profonds à l'aide de MAPPO. Conformément à ces recommandations, les hyperparamètres pertinents utilisés pour entraîner nos modèles sont résumés dans le tableau I. Tirant parti du fait que nos agents sont homogènes, nous avons utilisé le partage des paramètres pour accélérer l'apprentissage. Nous normalisons les récompenses et les observations. Nous avons entraîné nos modèles sur un ordinateur doté de 128 Go de RAM DDR5, d'un processeur à 16 cœurs cadencés à 5,73 GHz et d'un GPU GeForce RTX 4090 pour 200 millions pas de temps dans l'environnement.

Les résultats présentés dans cet article ont été obtenus avec des agents entraînés contre une cible fixe : la position de la cible est toujours aléatoire à chaque épisode d'entraînement, mais elle ne bouge pas. La récompense cumulée moyenne des épisodes au cours de l'entraînement pour 2 à 4 suiveurs est présentée dans la Fig. 3. Il est facile de voir que dans les trois cas, l'entraînement a convergé. Nous pouvons observer que les gains cumulés finaux augmentent avec le nombre d'agent. Cette différence peut être attribuée à la fonction de récompense qui récompense la minimisation de l'incertitude dans la position triangulée, et cette incertitude est connue pour diminuer avec le nombre de points de vue. La vitesse de convergence diminue avec le nombre d'agent, ce qui est normal puisque le temps d'apprentissage augmente avec la taille du modèle à entraîner, bien que la taille du batch utilisé pour chaque épisode d'apprentissage avec MAPPO augmente également avec le nombre d'agents. Nous pouvons notamment observer que le cas à quatre suiveurs converge beaucoup plus lentement, il n'atteint pas complètement l'asymptote après 200 millions pas de temps.

Pour évaluer les performances des modèles entraînés, nous

TABLE I
Hyperparamètres d'entraînement

Hyperparamètres	Valeur
num training episodes	num training steps / buffer length
batch size	num envs × buffer length × num agents
mini batch size	batch size / num of mini-batches
num parallel envs	128
num training steps	60e6
buffer length	1024
num of mini-batches	1
num of epoch per training	15
actor learning rate	5e-4
critic learning rate	5e-4
clip parameter	0.2
entropy coefficient	0.01
value loss coefficient	0.5
optimizer	Adam
optimizer epsilon	1e-5
weight decay	None

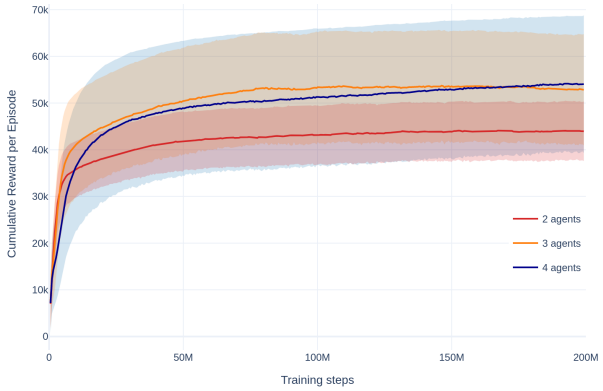


Fig. 3. Récompense moyenne cumulée par épisode au cours de l'entraînement pour différents nombres d'agents

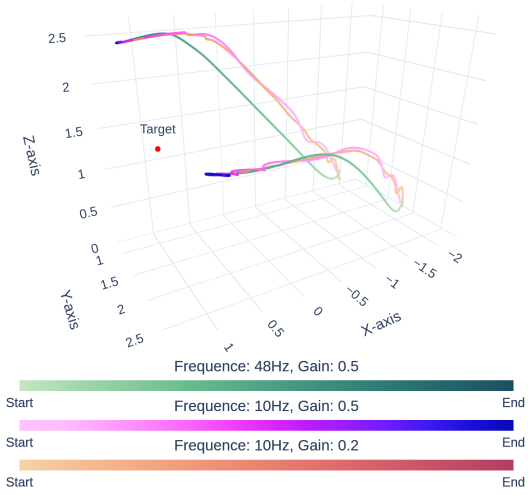


Fig. 4. Comparaison de l'effet de la fréquence de contrôle et du gain sur des trajectoires obtenues lors de vols simulés.

les avons évalués à la convergence dans un environnement réaliste, appelé *dronesim* [27], qui utilise le moteur Bullet Physics [28]. Cette fois, le modèle dynamique des drones est simulé et les drones sont dirigés à l'aide de lois de commande réalistes. Comme souvent dans les simulateurs de drones, le simulateur *dronesim* dispose de contrôleurs de guidage de haut niveau qui acceptent des positions de référence en entrée. Nos modèles calculent en sortie une commande en position autour de la position actuelle du drone dans le repère inertiel, que le contrôleur du drone convertit ensuite en commandes de bas niveau pour contrôler les rotors. Cependant, nos modèles ont été entraînés dans une simulation simpliste, où la dynamique du drone n'est pas prise en compte. En pratique, il est nécessaire d'ajuster la sortie du réseau en la multipliant par un gain, ajusté pour s'adapter à la dynamique du drone et à la fréquence de contrôle, afin de garantir que les commandes soient réalisables par le drone réel. Lorsqu'un contrôleur de drone reçoit une commande en position, il passe par des phases d'accélération, de vitesse maximale, et de décélération pour atteindre la position cible, lié à la dynamique et l'inertie du drone. Un gain élevé éloigne les commandes générées de la position du drone, ce qui encourage le drone à accélérer rapidement jusqu'à sa vitesse maximale. Au contraire, un gain faible rapproche les commandes en position du drone, trop proche et le drone n'accélérera pas jusqu'à sa vitesse maximale, le rendant plus lent. La fréquence de contrôle influe également sur la valeur du gain. Les commandes sont générées en supposant que le drone atteindra la position demandée au pas de temps suivant. Lorsque le gain est élevé et la fréquence faible, le drone peut dépasser la position désirée, produisant des oscillations, surtout en vol stationnaire. Si la fréquence de contrôle est très élevée, ce phénomène est largement amorti. Dans la Fig. 4 nous avons tracé les trajectoires de deux drones autour d'une cible fixe avec différentes valeurs de gain et de fréquence de contrôle. Il apparait que lorsque la fréquence de contrôle est haute, les trajectoires sont plus stables. Au contraire lorsqu'elle est faible, un gain faible stabilise le vol.

La Fig. 5 présente plusieurs trajectoires obtenues dans le simulateur *dronesim* avec un gain correctement ajusté et une fréquence de 10Hz. Dans la figure en haut à gauche, nous pouvons observer comment les suiveurs se placent autour d'une cible fixe et stationnent à des positions fixes à une distance de sécurité de la cible. De plus, même si nous n'avons pas été entraînés à le faire, nous avons évalué notre politique contre une cible en mouvement. Les trois autres figures présentent une cible se déplaçant en cercle à la même vitesse maximale que les suiveurs (2m/s). Nous pouvons observer que, bien qu'entraînée avec des cibles fixes, la politique résultante peut s'adapter à des cibles qui se déplacent lentement. Ce résultat est dû aux propriétés markoviennes inhérentes à l'apprentissage par renforcement, car les actions de contrôle sont calculées sur la base de l'état actuel uniquement et non des données antérieures. Cependant, comme les politiques

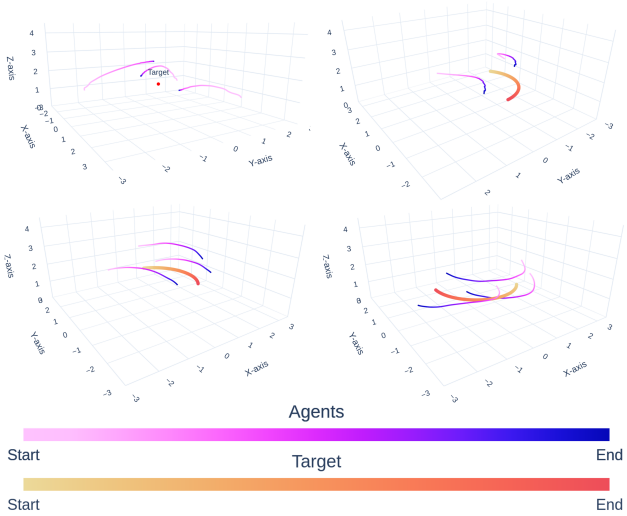


Fig. 5. Trajectoires obtenues lors de vols simulés. La cible est indiquée en gradients de rouge, Les trajectoires des drones sont tracées en gradients de bleu.

n'ont pas été formées avec une cible en mouvement, les suiveurs n'ont pas appris à anticiper les mouvements de la cible, ce qui entraîne un retard dans leur réponse.

VI. Test en vol réel

Pour observer le comportement dans des scénarios réels de la politique apprise, nous avons déployé les scénarios d'observation de cibles fixes utilisant deux suiveurs et trois suiveurs dans des démonstrations réelles. Les expériences ont eu lieu dans l'arène de vol de l'ENAC. Les dimensions de l'espace de vol sont $L \times W \times H = 8 \times 8 \times 4\,\text{m}$ et il est équipé d'un système de capture de mouvement, qui calculent la position et l'orientation des objets volant dans la volière.

Les engins utilisés lors de la démonstration sont des DJI Tello. Nous avons utilisé la bibliothèque Python DJITelloPy, qui dispose de contrôleurs de guidage de haut niveau acceptant des positions de référence en entrée. Par conséquent, comme pour V avec les drones simulés, nous transmettons directement la commande de position calculée, avec un gain de contrôle ajusté pour la dynamique de vol des Tellos et une fréquence de 10 Hz.

Des exemples de trajectoires de vol sont présentés dans les Fig. 6 et Fig. 7. Fig. 6, on remarque immédiatement que les trajectoires obtenues pendant les vols réels sont plus bruitées que les trajectoires simulés. Cela peut provenir du bruit dans les estimations de position des suiveurs, dans celle de la cible et de la dynamique de vol des suiveurs. Cependant, la politique de guidage réussit à atteindre et à se stabiliser au-dessus d'un point cible fixe.

Comme dans la section V, la politique est également testée sur une cible mobile, qui tourne en rond horizontalement au centre de la volière avec un rayon de 1,5 m et à 0,6 m de hauteur. Les trajectoires obtenues sont présentées Fig. 7, et nous pouvons comparer avec les mêmes vols effectués en simulation présentés Fig. 5. Les oscillations observées dans le vol réel sont dues aux perturbations du flux d'air de la

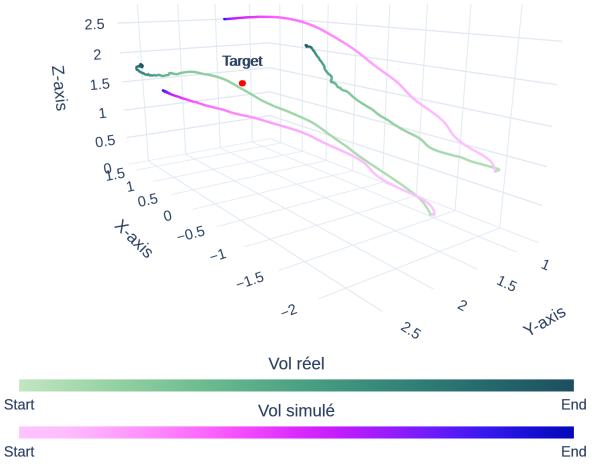


Fig. 6. Comparaison des trajectoires d'un vol réel et d'un vol simulé en direction d'une même cible.

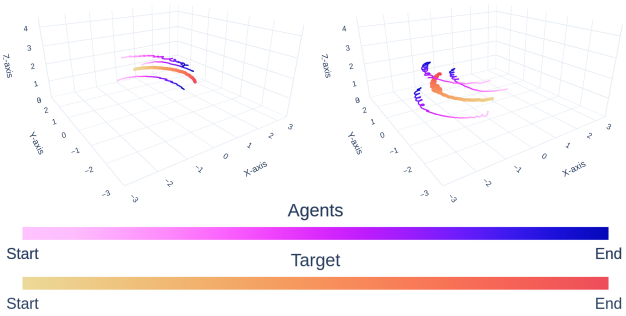


Fig. 7. Exemple de mouvements successifs de suiveurs lorsque la cible se déplace en cercle.

cible par le suiveur qui la survolent. Ces perturbations, qui ne sont pas simulés dans *dronesim*, font osciller la cible et par conséquent les suiveurs qui suivent ses mouvements.

VII. Conclusion

Nous avons introduit une approche de guidage de drones basée sur MARL pour la triangulation à N-vues de cibles volantes à l'aide de caméras omnidirectionnelles embarquées. Nous avons utilisé MAPPO pour apprendre avec succès des politiques décentralisées qui permettent aux drones d'ajuster dynamiquement leurs positions pour trianguler de manière optimale la cible, réduisant ainsi l'incertitude dans l'estimation de la localisation. Nous avons proposé une fonction de récompense basée sur la trace de la matrice de covariance de la position triangulée, calculée à l'aide d'un modèle analytique de propagation de l'incertitude. Une comparaison avec l'approche de Monte Carlo montre un bon accord avec le modèle proposé. En outre, nous avons évalué les politiques entraînées en simulation et les avons testées lors d'expériences en vol réel, montrant qu'elles peuvent gérer différents scénarios et être transférées dans des environnements réalistes. Notre approche est une étape prometteuse vers l'utilisation de MARL pour le suivi de plusieurs drones. Outre des évaluations et des analyses expérimentales plus approfondies, nous évaluerons à l'avenir le comportement du système avec un plus grand nombre d'agents. Plus important encore, nous ne supposerons plus que les positions des suiveurs sont connues de tous les suiveurs, mais seulement partiellement observées par la vision. Nous envisagerons également le cas de cibles multiples.

Remerciements

Les auteurs tiennent à remercier Arnaud SAMAMA, Frédéric BARBARESCO, Jean-Marc TRIN et Pierre-Louis CARTIER de Thales LAS, ainsi que Florence ALIGNE de Thales RT pour leur précieuse contribution à ce sujet.

References

[1] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 611–24 624, Dec. 2022.

[2] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," Tech. Rep., Aug. 2017.

[3] S. Ramalingam, S. K. Lodha, and P. Sturm, "A generic structure-from-motion framework," *Computer Vision and Image Understanding*, vol. 103, no. 3, pp. 218–228, Sep. 2006.

[4] R. A. Zitar, A. Mohsen, A. E. Seghrouchni, F. Barbaresco, and N. A. Al-Dmour, "Intensive Review of Drones Detection and Tracking: Linear Kalman Filter Versus Nonlinear Regression, an Analysis Case," *Archives of Computational Methods in Engineering*, vol. 30, no. 5, pp. 2811–2830, Jun. 2023.

[5] A. Khan, B. Rinner, and A. Cavallaro, "Cooperative Robots to Observe Moving Targets: Review," *IEEE Transactions on Cybernetics*, vol. 48, no. 1, pp. 187–198, Jan. 2018.

[6] B. Bethke, M. Valenti, and J. How, "Cooperative Vision Based Estimation and Tracking Using Multiple UAVs," in *Advances in Cooperative Control and Optimization*, ser. Lecture Notes in Control and Information Sciences, P. M. Pardalos, R. Murphey, D. Grundel, and M. J. Hirsch, Eds. Berlin, Heidelberg: Springer, 2007, pp. 179–189.

[7] W. Zhou, J. Li, Z. Liu, and L. Shen, "Improving multi-target cooperative tracking guidance for UAV swarms using multi-agent reinforcement learning," *Chinese Journal of Aeronautics*, vol. 35, no. 7, pp. 100–112, Jul. 2022.

[8] S. Srigrarom, S. M. Lee, M. Lee, F. Shaohui, and P. Ratsamee, "An Integrated Vision-based Detection-tracking-estimation System for Dynamic Localization of Small Aerial Vehicles," in *2020 5th International Conference on Control and Robotics Engineering (ICCRE)*, Apr. 2020, pp. 152–158.

[9] Y. Chen, M. Tsukada, and H. Esaki, "Reinforcement Learning Based Optimal Camera Placement for Depth Observation of Indoor Scenes," Tech. Rep., Oct. 2021.

[10] M. A. Akhloufi, S. Arola, and A. Bonnet, "Drones Chasing Drones: Reinforcement Learning and Deep Search Area Proposal," *Drones*, vol. 3, no. 3, p. 58, Sep. 2019.

[11] Z. Xia, J. Du, J. Wang, C. Jiang, Y. Ren, G. Li, and Z. Han, "Multi-Agent Reinforcement Learning Aided Intelligent UAV Swarm for Target Tracking," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 931–945, Jan. 2022.

[12] F. A. Oliehoek, C. Amato *et al.*, *A concise introduction to decentralized POMDPs*. Springer, 2016, vol. 1.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[14] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1928–1937.

[15] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning." PMLR, Jul. 2018, pp. 4295–4304.

[16] J. Su, S. Adams, and P. Beling, "Value-Decomposition Multi-Agent Actor-Critics," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 13, pp. 11 352–11 360, May 2021.

[17] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'18/IAAI'18/EAAI'18, vol. 32, no. 1. New Orleans, Louisiana, USA: AAAI Press, Apr. 2018, pp. 2974–2982.

[18] R. Lowe, Y. WU, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[19] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge: Cambridge University Press, 2004.

[20] F. Felten, C. Ledez, P.-Y. Houitte, E.-G. Talbi, and G. Danoy, "Crazyrl: A multi-agent reinforcement learning library for flying crazyflie drones," https://github.com/ffelten/CrazyRL, 2023.

[21] G. Di Leo, C. Liguori, and A. Paolillo, "Propagation of uncertainty through stereo triangulation," in *2010 IEEE Instrumentation & Measurement Technology Conference Proceedings*, May 2010, pp. 12–17.

[22] C. M. G and P. M. Harris, "Software Support for Metrology Best Practice Guide No 6 - uncertainty evaluation." Sep. 2006.

[23] J. Terry, B. Black, N. Grammel, M. Jayakumar, A. Hari, R. Sullivan, L. S. Santos, C. Dieffendahl, C. Horsch, R. Perez-Vicente *et al.*, "Pettingzoo: Gym for multi-agent reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 15 032–15 043, 2021.

[24] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018.

[25] DeepMind, I. Babuschkin, K. Baumli, A. Bell, S. Bhupatiraju, J. Bruce, P. Buchlovsky, D. Budden, T. Cai, A. Clark, I. Danihelka, A. Dedieu, C. Fantacci, J. Godwin, C. Jones, R. Hemsley, T. Hennigan, M. Hessel, S. Hou, S. Kapturowski, T. Keck, I. Kemaev, M. King, M. Kunesch, L. Martens, H. Merzic, V. Mikulik, T. Norman, G. Papamakarios, J. Quan, R. Ring, F. Ruiz, A. Sanchez, L. Sartran, R. Schneider, E. Sezener, S. Spencer, S. Srinivasan, M. Stanojević, W. Stokowiec, L. Wang, G. Zhou, and F. Viola, "The DeepMind JAX Ecosystem," 2020.

[26] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," Tech. Rep., Aug. 2018.

[27] L. T. Fernandez, M. Bronz, T. Lefebvre, and N. Bartoli, "Multi-Vehicle Simulation Framework for Heterogeneous Unconventional MAVs," in *IMAV 2023*, AACHEN, Germany, Sep. 2023.

[28] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

Couplage entre un apprentissage par renforcement profond et une machine à états : approche théorique

Idriss Abdallah*†, Laurent Ciarletta†, Patrick Hénaff†, Jonathan Champagne* and Matthieu Bonavent*
*Naval Group
Gassin, France
†LORIA, CNRS, Université de Lorraine
Nancy, France
Mail : idriss.abdallah@loria.fr

***Abstract*—Dans le cadre de la mise au point de torpilles chez Naval group, les algorithmes de contrôles embarqués sont développés en simulation numérique. Le simulateur intègre plusieurs modèles pour évaluer les performances dans une mise en situation la plus proche possible de la réalité. Parmi tous ces modèles, on distingue le modèle opérateur dont le rôle est de simuler la communication entre le lanceur et la torpille. Cependant, les modèles opérateurs mis en place jusque là grâce à des méthodes symboliques, dont notamment des machines à états, n'ont pas un niveau de représentativité satisfaisants.**

D'un autre côté, l'apprentissage par renforcement profond a récemment montré de très bonnes capacités à résoudre des problèmes de décision séquentielle complexes. Il semble donc, a priori, capable de répondre à cette problématique industrielle. Cependant, cette approche souffre d'une inefficience de l'utilisation de ses données et d'un manque d'interprétabilité dû à l'utilisation des réseaux de neurones.

Nous proposons ici une approche théorique visant à étudier le couplage entre une machine à états et l'apprentissage par renforcement profond afin de tirer profit des connaissances métiers sur l'environnement opérationnel pour pallier certaines difficultés de l'apprentissage par renforcement profond et ainsi obtenir un modèle opérateur à la fois représentatif et explicable.

***Index Terms*—Apprentissage par renforcement, Apprentissage par renforcement profond, Machine à états, Connaissance extérieure, Interprétabilité**

I. Contexte industriel

Naval Group est un acteur international dans le domaine du naval de défense qui conçoit et produit une grande diversité de produits, dont les plus notables sont les bâtiments de surface et les sous-marins. C'est un systémier intégrateur présent sur l'ensemble du cycle de vie de ses produits. Pour l'armement de ces derniers, il s'occupe notamment du développement de torpilles.

Dans le cadre de la mise au point de torpilles, un effort important est porté sur le développement d'algorithmes embarqués de guidage et de prise de décision permettant d'atteindre une cible. Dans le cas étudié, une liaison bi-directionnelle entre le lanceur (i.e. bâtiment mettant en oeuvre l'arme) et la torpille permet une communication entre ces deux acteurs :

- Le logiciel embarqué de la torpille remonte différentes informations du contexte opérationnel à l'opérateur (cinématique, détections acoustiques, ...).
- L'opérateur a la possibilité d'influer sur les décisions de l'intelligence embarquée de la torpille.

En plus des informations remontées par la torpille, l'opérateur dispose également des informations données par les capteurs du lanceur. Le schéma 1 illustre les différents éléments et leurs interactions dans la situation tactique entretenue.

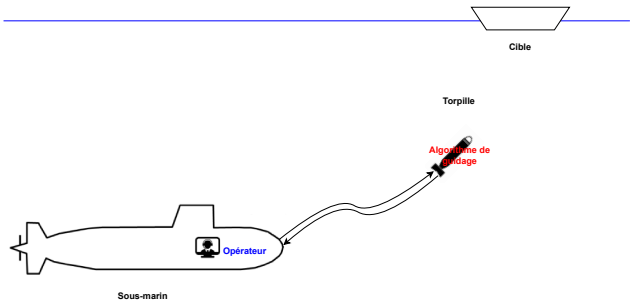


Fig. 1: Contexte industriel

Afin de développer et d'évaluer ces algorithmes embarqués, un simulateur numérique permet de simuler les différents éléments représentés dans le schéma 2. L'environnement doit être aussi représentatif que possible afin d'évaluer de façon fidèle les performances des algorithmes. Il est donc, entre autres, nécessaire de créer un modèle de l'opérateur permettant de recréer les interactions possibles avec l'algorithme au cours d'un tir.

Dans cette optique, plusieurs modèles opérateurs ont été mis au point à partir de méthodes symboliques, mais tous ces modèles offrent une représentativité assez faible tant au niveau du comportement que des performances. Cela est notamment induit par la complexité de la tâche à résoudre et le fait qu'il n'existe pas de modèle applicable simplement contrairement à des modèles physiques. Il est néanmoins intéressant de noter que, par cette voie de modélisation classique, le meilleur modèle opérateur a été créé à partir d'une machine à états.

En accord avec le contexte industriel et l'état de l'art dans le domaine de la décision séquentielle, une approche à base d'apprentissage par renforcement profond a été retenue pour créer un nouveau modèle opérateur. Bien que cette

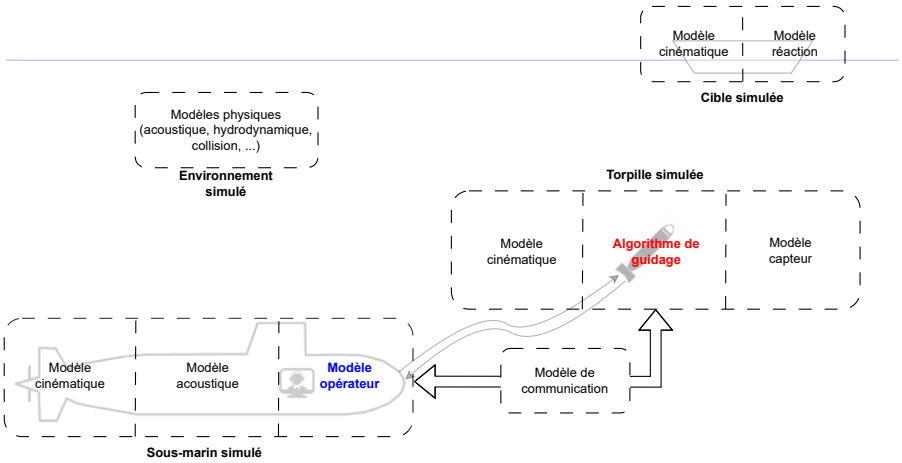


Fig. 2: Environnement de simulation du contexte industriel

méthode semble adéquate pour la mise en place d'un modèle opérateur performant, elle comporte des défauts dont les plus notables sont l'efficacité de l'utilisation des données, ainsi qu'un manque d'interprétabilité du modèle dû à l'utilisation des réseaux de neurones. Or, ces deux aspects sont cruciaux. L'augmentation de l'efficacité de l'utilisation de données permet de réduire les ressources calculatoires nécessaires à l'apprentissage. L'interprétabilité du modèle, quant à elle, est essentielle dans la validation de son utilisation dans un simulateur certifié. Pour pallier à ces difficultés, l'utilisation de la machine à états déjà existante est proposée dans une approche d'apprentissage couplée pour une accélération de la convergence de l'apprentissage vers un modèle performant et pour augmenter l'interprétabilité du modèle final.

Afin d'illustrer les approches proposées, l'environnement classique d'apprentissage par renforcement du Lunar Lander a été choisi afin d'avoir un substitut non sensible au simulateur industriel.

II. État de l'art

L'apprentissage par renforcement (*Reinforcement Learning* ou RL) est une approche d'apprentissage automatique qui utilise les actions d'un agent dans un environnement pour apprendre un comportement maximisant une récompense obtenue au cours de ses interactions [1]. Cette approche couplée à la capacité d'approximation de fonctions des réseaux de neurones a permis l'apparition du *Deep Reinforcement Learning* (DRL) qui est à la base de plusieurs avancées récentes dans le domaine du contrôle séquentiel [2] [3].

Le succès vient notamment des approches dites "*Model-Free*" qui n'ont pas besoin d'avoir un modèle de transition de l'environnement, mais seulement d'une fonction de récompense. L'utilisation de fonction de récompense est une approche expressive pour définir un problème [4] bien qu'elle soit en pratique difficile à mettre en place, car son design influe fortement sur les capacités d'apprentissage des algorithmes [5]. Cette liberté vient également au prix de la nécessité d'un nombre important d'interactions pour obtenir un comportement optimal, notamment dû au dilemme exploration-utilisation [1]. De plus, ce couplage entre RL et *Deep Learning* entraîne plusieurs défauts inhérents aux réseaux de neurones : explicabilité [6], stabilité et reproductibilité de l'entraînement [7]. Ainsi, il reste encore plusieurs points cruciaux à améliorer afin que le DRL soit applicable à une plus grande variété de problèmes [8].

Une des principales préoccupations de la communauté scientifique du DRL est l'amélioration de la rapidité et de la capacité de convergence des algorithmes. Pour cela, il y a des approches agnostiques de l'environnement à la base de plusieurs concepts clés. Ces méthodes agissent sur des aspects différents des algorithmes [1]. Par exemple, les approches *On-Policy/Off-Policy*, les approches de types *Value-Actor*, *Policy Gradient*, *Actor-Critic*, le rejeu d'expérience [9], une optimisation de l'entropie [10] [11] ou l'ajout de motivation intrinsèque [12]. D'autres approches se concentrent sur l'ajout de connaissances extérieures sur le problème afin de faciliter la convergence vers une politique adéquate. Le format, la quantité et l'utilisation de ces connaissances sont très variables, on peut trouver parmi les approches les plus classiques l'Imitation Learning [13], le *Curriculum Learning* [14] ou bien le *Transfer Learning* [15].

Plusieurs approches fusionnent le DRL avec l'utilisation d'une machine à états finie (MAE) ou un automate déterministe fini. Une première approche est d'intégrer cette MAE dans la structure du problème de DRL en agissant sur l'espace d'observation ou/et d'action avec une approche hiérarchique, avec un macro et un micro contrôleur où l'un des deux est géré par une MAE et l'autre appris par DRL [16] [17]. D'autres approches couplent la MAE et le DRL au niveau algorithmique, c'est le cas des *Reward Machines* en ajoutant une structure de MAE sur la fonction de récompense, ce qui permet notamment d'augmenter l'espace d'observation et d'ajouter un mécanisme de rejeu d'expérience basé sur les états [18]. De plus, une MAE peut directement être apprise

afin d'aider la convergence comme pour les *Subgoal Automata* [19] ou [20]. Ainsi, le couplage d'une machine à états et d'un algorithme de DRL peut améliorer la capacité, la stabilité et l'interprétabilité [21] de l'apprentissage.

Dans cet article, nous nous intéressons à une approche de couplage algorithmique entre un algorithme de DRL et une MAE. Cette dernière représente des connaissances a priori sur l'environnement pour augmenter l'interprétabilité du processus d'apprentissage et potentiellement l'améliorer en permettant d'incorporer des heuristiques exploratoires externes.

Nous présenterons dans un premier temps les définitions nécessaires pour le DRL, les MAE ainsi que pour l'environnement (section III) et les outils utilisés pour les études menées (section IV). Dans un deuxième temps, nous définirons plusieurs approches théoriques pour le couplage d'une MAE avec un algorithme d'apprentissage par renforcement (section V). Puis, nous utiliserons ce formalisme afin de recontextualiser plusieurs méthodes de l'état de l'art (section VI). Par la suite, nous présenterons plusieurs possibilités de couplage identifiées (section VII). Finalement, nous conclurons sur les perspectives envisagées dans le contexte industriel qu'apporte le couplage proposé entre une MAE et un algorithme de DRL.

III. Définitions

A. Apprentissage par renforcement

L'apprentissage par renforcement est un domaine de l'apprentissage automatique où un agent interagit avec un environnement par une action, ce qui induit une transition de l'état de l'environnement. Il reçoit alors une récompense sous forme d'un scalaire qui évalue la transition effectuée. Le but de l'apprentissage par renforcement est d'apprendre grâce à des interactions avec l'environnement, le comportement qui maximise la récompense obtenue au cours de ses interactions.

Le formalisme mathématique utilisé classiquement pour représenter un tel problème est le processus de décision markovien, abrégé MDP pour *Markov Decision Process*. On se placera par la suite dans le cas d'un MDP déterministe. Il est défini par le tuple $\langle \mathcal{S}, \mathcal{A}, \mathbf{T}, \mathbf{R}, \rho_0 \rangle$ avec :

- $\mathcal{S}$, l'ensemble des états possibles.
- $\mathcal{A}$, l'ensemble des actions à la disposition de l'agent.
- $\mathbf{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, une fonction de transition qui régit l'évolution de l'environnement.
- $\mathbf{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, la fonction de récompense.
- ρ_0, la distribution de l'état initial sur l'ensemble des états

On définit également le gain cumulé réduit, $G_t = \sum_k \gamma^k r_{t+k+1}$, où r_t désigne la récompense reçue à l'instant t, qui permet d'estimer la récompense obtenue sur un certain horizon temporel en fonction d'un facteur de réduction γ. Une fonction de politique π représente alors une fonction de prise de décision qui permet à partir d'une observation, de choisir une action. La résolution du problème est la recherche de π^*, la politique optimale qui maximise le gain cumulé réduit.

B. Machine à états

Dans la suite, nous nous placerons dans le cadre d'une machine à états finie (MAE) définie par le tuple $\langle \mathcal{Q}, \mathcal{U}, \delta, q_0 \rangle$ un ensemble d'états possibles, $\mathcal{Q}$, un ensemble d'observable, $\mathcal{U}$, une fonction de transition, $\delta : \mathcal{Q} \times \mathcal{U} \to \mathcal{Q}$ ainsi qu'un état initial $q_o \in \mathcal{Q}$.

C. Environnement

Le type d'environnement étudié est une simulation où l'agent n'a accès qu'à une observation dégradée de l'environnement réel dans lequel il évolue. D'un côté, l'observation partielle à disposition du modèle opérateur est l'ensemble constitué des données fournies par ses propres capteurs ainsi que des informations remontées périodiquement par la liaison de communication. D'un autre côté, l'observation globale (i.e. réelle) est la position de la cible. Cette dichotomie est largement présente dans le cas de simulation de contrôle d'un véhicule ou bien de personnages dans un monde fictif où les données comme la position des obstacles ou de l'objectif ne sont observables qu'en fonction de la position de l'agent via des capteurs ou un point de vue [25] [26].

Ce type de problématique peut alors être posé sous la forme d'un processus de décision markovien partiellement observable (POMDP). Il se définit par un tuple $\langle \mathcal{S}, \mathcal{A}, \mathbf{T}, \mathbf{R}, \rho_0, \Omega, \mathrm{O} \rangle$ avec :

- $\langle \mathcal{S}, \mathcal{A}, \mathbf{T}, \mathbf{R}, \rho_0 \rangle$ est un MDP défini tel que précédemment.
- $\mathcal{S}$, l'espace d'observation globale.
- Ω, l'espace d'observation partielle.
- $\mathrm{O} : \mathcal{S} \to \Omega$ permet d'obtenir l'observation partielle à partir de l'observation globale.

Cependant, le formalisme du POMDP induit classiquement l'ajout d'une probabilité sur l'espace des états, communément appelé vecteur de croyance, car ce formalisme s'applique le plus souvent à des problèmes non-markoviens. Dans notre cas, le formalisme permet de traduire une plus grande complexité d'extraction de caractéristiques pertinentes pour la prise de décision à partir de l'observation partielle plutôt qu'à partir de l'observation globale.

Le problème à résoudre est donc de trouver la politique optimale choisissant une action à partir de Ω.

Dans le cas de la définition d'une MAE, l'espace $\mathcal{Q}$ peut être soit $\mathcal{S}$ ou Ω. Dans cette étude, les deux possibilités seront considérées. Les différences sur les possibilités d'utilisation et les intérêts de ces deux approches seront explicitées par la suite.

Pour la suite, nous utiliserons l'environnement Lunar Lander [23] afin d'illustrer simplement les approches présentées. L'espace $\mathcal{S}$ sera le vecteur d'observation physique et l'espace Ω sera construit à partir du rendu graphique de l'environnement. Afin de réduire l'aspect non-markovien, l'observation sera composée d'une concaténation comme proposée dans [2].

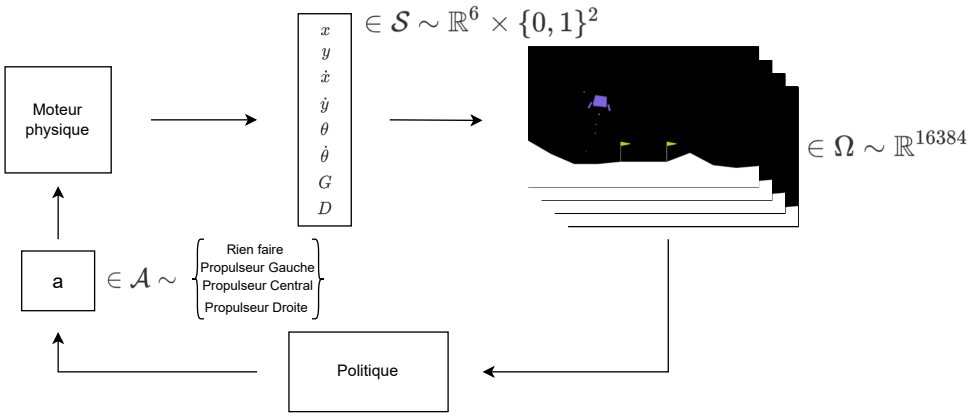


Fig. 3: Relation entre les différents espaces d'observation considérés pour l'environnement Lunar Lander

IV. Outils

A. Environnement

Le problème du Lunar Lander est issu de [23]. C'est un environnement de contrôle discret 2D dont le but est de faire atterrir un vaisseau spatial sur la Lune entre deux drapeaux. Le vecteur d'observation, appartenant donc à $\mathcal{S}$, est de dimension 8 et est constitué de :

- La position : x, y
- La vitesse : $\dot{x}$, $\dot{y}$
- L'angle de roulis : θ
- La vitesse de roulis : $\dot{\theta}$
- Le contact des pieds au sol (deux booléens) : G, D

Son espace d'action est un choix discret entre les 4 actions correspondantes à l'activation d'un de ses 3 réacteurs ou bien de ne rien faire.

Sa fonction de récompense l'encourage à se poser de manière la plus stable possible tout en minimisant l'utilisation de ses réacteurs. Après chaque pas de temps, une récompense est donnée. La récompense totale d'un épisode est la somme de toutes les récompenses obtenues au cours d'un épisode.

Pour chaque pas de temps, la récompense est :

- augmentée/réduite plus le vaisseau est proche/loin de la plate-forme d'atterrissage.
- augmentée/réduite plus le vaisseau est lent/rapide.
- résuite plus le vaisseau à un angle important par rapport à la verticale.
- augmentée de 10 points pour chaque pied qui est en contact avec le sol.
- réduite de 0.03 pour chaque pas de temps où un moteur latéral est utilisé.
- réduite de 0.3 pour chaque pas de temps où le moteur central est utilisé.

A la fin de l'épisode, un récompse finale de -100 ou +100 est donnée en fonction de la réussite ou non de l'atterrissage.

Un épisode est considérée comme étant réussie à partir d'un récompense totale d'au moins 200 points [24] . Les épisodes durant plus de 1000 pas de temps sont tronqués.

Une seconde version du problème est introduite où l'observation est constituée à partir du rendu graphique de l'environnement, en concaténant 4 images consécutives réduites en taille et mise en niveau de gris, ce qui permet de construire les observations issues d'Ω. Le schéma 3 explicite le placement des espaces d'observations $\mathcal{S}$ et Ω dans le cas du Lunar Lander.

B. Machine à états utilisée

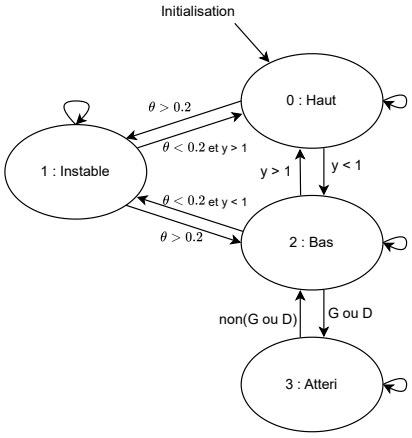


Fig. 4: États de la MAE avec leurs conditions de transition

La MAE utilisée pour la suite a pour but de découper l'espace d'état avec une heuristique simple. L'objectif de ce découpage est d'avoir des états où il est attendu que la politique soit différente afin de pouvoir déceler des difficultés d'apprentissage sur différents ensembles d'états. Les états sont :

- Etat 0 : Haut. Le vaisseau a un angle très faible et est dans la partie haute de l'environnement.
- Etat 1 : Instable. Le vaisseau a un angle non négligeable.

- Etat 2 : Bas. Le vaisseau a un angle très faible et est dans la partie basse de l'environnement.
- Etat 3 : Atterri. L'un des deux pieds du vaisseau touche le sol.

Le schéma 4 représente la MAE ainsi que les conditions de transitions entre ces différents états.

V. Mise en place de la MAE

Nous décomposerons simplement un algorithme de DRL comme étant la succession de deux phases. Une première phase exploratoire de récolte d'expérience et une seconde phase d'apprentissage utilisant ces expériences afin de mettre à jour l'état de la politique qui servira pour créer une nouvelle politique exploratoire pour une nouvelle phase exploratoire. Cette représentation est illustrée par la figure 5. Avec ce découpage simple, nous pourrons étudier plusieurs possibilités de placements de MAE.

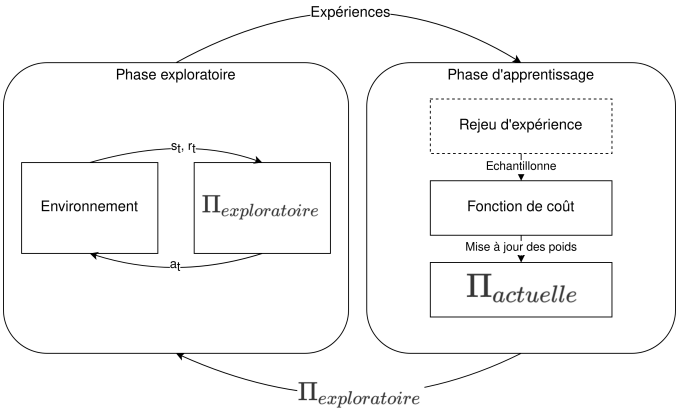


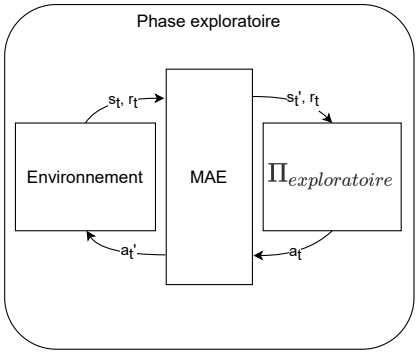
Fig. 5: Schéma générique d'apprentissage par renforcement

A. Définition du couplage de la MAE au sein de l'algorithme de DRL

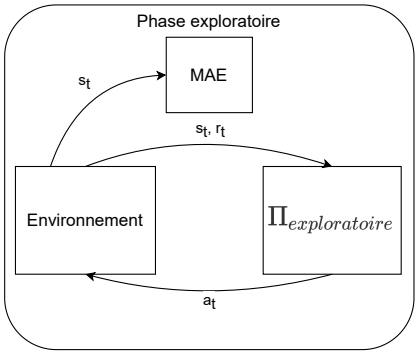
Une première question est si la fonction de transition de la MAE est utilisée lors de la phase d'exploration ou bien lors de la phase d'apprentissage. En effet, les deux placements n'ont pas accès aux mêmes informations.

Une première façon consiste à utiliser la MAE lors de phase d'exploration. Dans ce cas, on s'appuie sur la MAE à chaque interaction avec l'environnement. Elle peut alors être placée en "série", c'est-à-dire qu'elle permet de définir l'espace d'observation et/ou l'espace d'action de la politique apprise par DRL comme décrit par la figure 6a. Elle peut également être placée en parallèle de la politique, c'est à dire qu'elle utilise le même vecteur d'observation que la politique comme décrite par la figure 6b.

Lorsque la MAE est utilisée lors de la phase d'apprentissage, les informations à sa disposition sont l'ensemble des transitions en mémoire et des métriques d'apprentissage calculées.



(a) MAE placée en série



(b) MAE placée en parallèle

Fig. 6: Placement schématique d'une MAE en phase exploratoire

B. Modification algorithmiques permises par ces différents usages de la MAE

Les modifications algorithmiques induites par l'utilisation d'une MAE peuvent être soit en phase exploratoire, soit en phase d'apprentissage.

Pour les modifications en phase exploratoire, celles qui modifient l'espace d'entrée de la politique et/ou le choix des actions, doivent être gardées pour la solution finale. Dans le cas d'une MAE utilisée dans la phase exploratoire, le placement en série implique naturellement ce type de modifications. Pour une MAE placée en parallèle, cela dépend de si elle est utilisée pour augmenter l'espace d'observation et/ou pour changer activement l'action choisie. Ainsi, une MAE utilisant $\mathcal{S}$ et modifiant l'espace d'observation ou d'action ne peut être utilisée comme une solution finale à un problème de décision séquentielle, car cela revient à utiliser des informations non accessibles.

Pour les modifications en phase d'apprentissage, l'utilisation de la MAE a une influence sur la manière algorithmique de mettre à jour les poids du réseau de neurones et/ou sur la politique exploratoire utilisée pour la phase exploratoire. Une différence notable par rapport aux modifications en phase exploratoire est qu'elles n'induisent pas de modifications sur

la définition de la politique finale.

VI. Reconsidération de l'état de l'art au regard des définitions précédentes

A. Fonctionnement et modification en phase exploratoire

Les approches mêlant une MAE avec le DRL de cette façon modifient la problématique initiale de l'environnement en utilisant une MAE pour la définition de l'espace d'observation et/ou d'action de la partie apprise par DRL.

Ainsi, plusieurs approches placent la MAE en série avec une architecture utilisant un macro et un micro contrôleur afin de cibler l'apprentissage sur une partie considérée comme complexe tout en gardant un contrôle et une compréhension sur le modèle final. Par exemple, dans [17], la fonction de politique est apprise pour un seul état de la MAE et garde une définition symbolique pour les fonctions de transition et pour la politique appliquée dans les autres états. Dans [16], la politique apprend directement en utilisant l'état du système mais son espace d'action agit sur une MAE construite à partir d'un contrôleur Proportionnel Dérivé qui transmet des actions plus bas niveau aux actionneurs.

Une utilisation d'une MAE placée en parallèle est proposée par [18]. Tout d'abord, la fonction de récompense est une *Reward Machine* et l'observation est augmentée par l'état courant de la MAE qui est utilisée en parallèle des interactions de la politique afin d'obtenir la récompense. Un résultat intéressant prouvé par [18] est que cette utilisation peut rendre un problème, initialement non markovien, markovien grâce aux informations fournies par l'état courant de la MAE.

B. Fonctionnement en phase exploratoire et modification de la phase d'apprentissage

Ce type d'approche utilise une MAE lors de la phase d'exploration et tire profit des informations données par la MAE sur les transitions afin d'améliorer l'apprentissage.

Les algorithmes les plus notables sont le HRM et le QRM issus de [18] qui profitent tous les deux de la structure de MAE pour augmenter artificiellement le nombre de transitions rencontrées en rejouant chaque transition avec tous les états possibles de la MAE. De plus, le HRM maintient une politique différente pour chacun des états mais comme cela a été soulignée, cela implique que la MAE doit être gardée pour la politique finale afin de choisir quelle politique doit être utilisée.

Plusieurs travaux se sont penchés sur le fait d'apprendre automatiquement une MAE ou un automate dans ce contexte [19] mais ces approches perdent alors l'intérêt d'utiliser une MAE créée par un humain qui est plus facilement interprétable.

C. Fonctionnement en phase d'apprentissage et modification de la phase exploratoire

Les méthodes de l'état de l'art se rapprochant le plus de ce concept sont issues du *Curriculum Learning* mais n'utilisent pas la structure de MAE, mais plutôt celle de graphe [14]. En effet, plusieurs approches mettent en place un ensemble de tâches et font évoluer la tâche courante à partir de laquelle les interactions sont générées en phase exploratoire en fonction de l'état d'avancement de la politique dans son processus d'apprentissage.

Placement			Modification	Article
Exploratoire	Parallèle	Ω	Exploratoire	[18] [19]
			Apprentissage	[18] [19]
		$\mathcal{S}$	Exploratoire	
			Apprentissage	
	Série	Ω	Exploratoire	[16] [17]
			Apprentissage	
Apprentissage			Exploratoire	[14]
			Apprentissage	

TABLE I: Tableau récapitulatif des méthodes présentées avec l'approche proposée

VII. Perspectives

Cette section propose des premières pistes de couplage entre une MAE et un algorithme de DRL appliquées à l'environnement *Lunar Lander*. Ces études ont comme but d'évaluer des gains en performances et en explicabilité afin d'avoir une approche permettant une application finale exploitable dans notre contexte industriel.

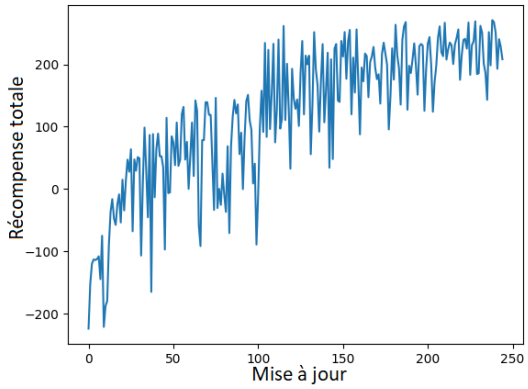
A. Nouvelles métriques d'évaluation de l'état courant de la politique

Plusieurs approches se basent sur des métriques de l'état courant de la politique afin de pouvoir mettre en place des rétroactions dans le processus d'apprentissage, par exemple, l'utilisation de l'entropie pour [10]. Ainsi, ces métriques permettent d'ajuster le mécanisme d'apprentissage en prenant en compte l'état courant de la politique afin d'ajuster la rétroaction.

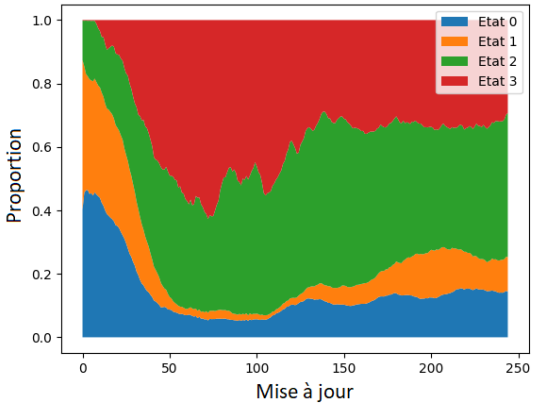
L'utilisation d'une MAE en phase exploratoire de façon parallèle peut permettre de classifier chacune des observations à un état issu de la fonction de transition. Cet ajout d'information peut permettre de créer de nouvelles métriques permettant d'évaluer l'état courant de la politique. Afin d'illustrer l'approche proposée, un entraînement utilisant PPO [10] sur l'environnement Lunar Lander a été fait, en utilisant la MAE de la section IV-B en parallèle lors de la phase exploratoire. Cela permet d'avoir, pour chacune des transitions de l'environnement, la valeur de l'état dans lequel la MAE est au cours de l'épisode sans modification algorithmique de l'apprentissage.

Un premier type de métrique est d'utiliser la proportion de chacun des états comme une métrique. La figure 7 montre la proportion moyenne de chacun des états rencontrés par la politique par épisode au cours d'un apprentissage. Ainsi, ce type de métrique peut permettre d'avoir des éléments de compréhension sur l'état courant de la politique au cours de l'apprentissage. En effet, on voit ici que le début de l'apprentissage se concentre sur la phase aérienne car la proportion de l'état 3 est très faible, puis le concept d'atterrissage

devient beaucoup plus présent avec une forte proportion d'état 3. Finalement, on observe une phase où les proportions des états convergent vers des valeurs stables avec notamment une récompense totale moyenne qui évolue beaucoup moins.



(a) Récompense moyenne par épisode



(b) Proportion moyenne par épisode de chaque état

Fig. 7: Apprentissage sur l'environnement Lunar Lander avec PPO en utilisant une MAE en parallèle à partir de $\mathcal{S}$ pour classifier les différents états

Un second type de métriques est d'utiliser des métriques déjà existantes, mais de les évaluer par état afin d'avoir une évaluation de l'état de la politique pour chacun des états. La figure 8 montre l'écart à la valeur moyenne de l'entropie de l'acteur par état cours de l'entraînement. On peut voir ici une différence notable entre les différents états et plus particulièrement de l'état 3 par rapport aux autres états donc l'évaluation de métriques par état semble intéressante pour déceler des différences au cours de l'apprentissage face à des transitions venant d'états différents.

De façon similaire à l'algorithme HRM issu de [18], il peut être intéressant de différencier le traitement algorithmique en fonction des états grâce à ces différentes métriques afin de profiter de la structure de la MAE pour diviser la complexité du problème global en sous-problèmes plus simple.

B. Utilisation d'une MAE Politique

Dans un cas où une MAE existe déjà pour répondre au problème de décision séquentiel visé, comme dans notre cas

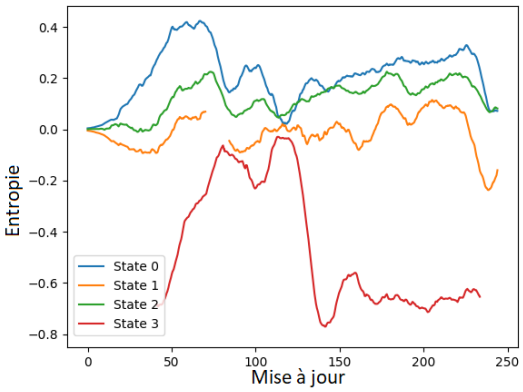


Fig. 8: Écart à la valeur moyenne de l'entropie de l'acteur par état

industriel, il peut être intéressant d'étudier la capacité de tirer profit de la structure de MAE tout en ayant accès à une politique pour chacun de ces états. Cependant, il est bon de noter, qu'on se place dans un contexte où la politique fournie par la MAE est sous-optimale car sinon, l'intérêt d'appliquer une méthode d'apprentissage par renforcement est inexistant.

On définit ici une machine à états politique (MAE-π) par une machine de Mealy définie par un tuple $\langle \mathcal{Q}, \mathcal{U}, \delta, q_0, \mathcal{A}, \pi \rangle$ avec $\langle \mathcal{Q}, \mathcal{U}, \delta, q_0 \rangle$ une MAE, $\mathcal{A}$ l'ensemble des actions possibles et $\pi : \mathcal{Q} \times \mathcal{U} \to \mathcal{A}$, une fonction de sortie, qui permet donc d'avoir une politique, dépendant de l'état courant de la MAE et de l'observable.

Dans la littérature, on trouve différentes méthodes pouvant utiliser des MAE politiques. En effet, il est possible d'utiliser les approches de *Transfer Learning* basées sur l'utilisation d'une politique extérieure (notamment le *Policy Transfer* ou le *Learning from demonstration*), bien qu'elles ne tirent pas profit de la structure de la MAE et qu'elles doivent être capables de tirer profit d'une politique imparfaite. Il peut donc être intéressant d'explorer l'utilisation de ces approches, mais en tirant profit de la structure de MAE, ce qui permettrait d'avoir un traitement différent et potentiellement plus adéquat pour chacun des états.

Les MAE politiques semblent intéressantes pour une utilisation en parallèle avec des modifications dans la phase exploratoire, car cela permet d'introduire une action alternative à celle fournie par la politique optimisée, ce qui peut permettre de guider l'exploration lors de l'apprentissage grâce à une heuristique métier si certains états sont complexes. De plus, il est également possible de restreindre l'utilisation de la politique issue du DRL à seulement certains états de façon similaire à [17].

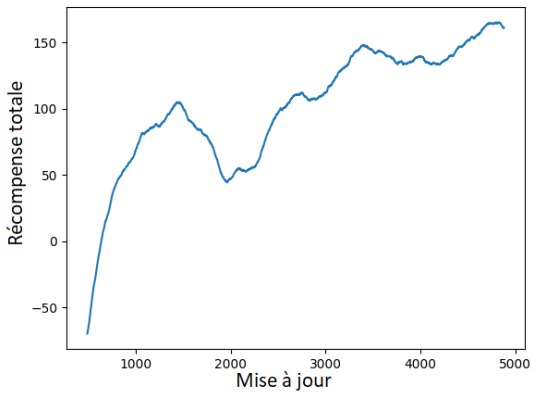
C. Utilisation de $\mathcal{S}$ comme espace d'entrée de la MAE

Une MAE utilisant directement Ω peut être complexe à mettre en place lorsque l'observation est difficile à traiter, notamment avec des données déstructurées (images, nuage de points lidars, sonars). Cependant, lorsque l'espace $\mathcal{S}$ existe et qu'il est possible grâce au simulateur d'y avoir accès, une

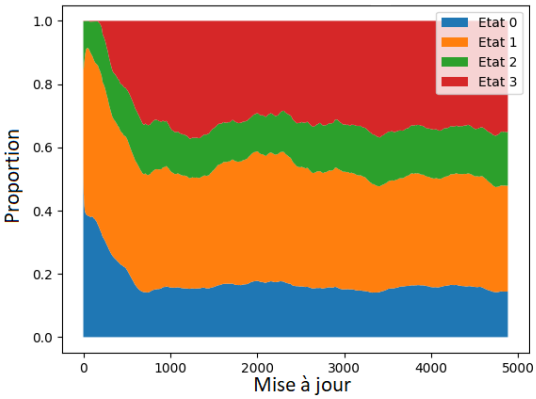
approche pourrait être d'utiliser directement $\mathcal{S}$ comme espace d'entrée de la MAE. En reprenant l'approche et les termes de [26] qui décompose le problème de la décision séquentielle en un problème de compréhension de l'observation et un problème de choix de l'action, dans notre cas, cette approche permet de nous affranchir du problème de compréhension de l'observation pour la création de la MAE. Ainsi, l'utilisation d'Ω semble intéressante dans les environnements où le problème de compréhension de l'observation est complexe.

Cependant, cela implique une utilisation de la MAE sans nécessité de l'utiliser pour l'inférence de la politique finale. En effet, les informations issues de $\mathcal{S}$ ne sont pas accessibles dans le cas d'une résolution réelle. Toutes les approches modifiant la phase d'apprentissage ou bien agissant seulement sur la politique exploratoire peuvent être envisagées, comme celles présentées dans les sections VII-A et VII-B. Cela permet de simplifier la création de la MAE et donc potentiellement d'augmenter son efficacité.

Afin d'illustrer ce concept, un apprentissage à partir de Ω tout en utilisant la MAE de la section IV-B en parallèle à partir de l'espace $\mathcal{S}$ a été fait à l'aide de PPO, ce qui implique aucun changement algorithmique dans le processus d'apprentissage. La figure 9 montre l'évolution de la récompense moyenne ainsi que de la proportion d'état par épisode. Tout d'abord, on note que la modification de l'observation complexifie l'apprentissage avec un nombre d'interactions nécessaires plus grand et une performance plus faible. L'ajout de la MAE permet notamment de voir que la proportion de l'état 1 (instable) est beaucoup plus grande que celle observée pour l'apprentissage à partir de $\mathcal{S}$ (figure 7) ce qui donne des informations sur les difficultés rencontrées au cours de l'apprentissage pour obtenir une politique capable de stabiliser l'angle de roulis du vaisseau à partir de Ω.



(a) Récompense totale moyenne par épisode



(b) Proportion moyenne par épisode de chaque état

Fig. 9: Apprentissage sur l'environnement Lunar Lander utilisant PPO à partir d'Ω en utilisant une MAE en parallèle à partir de $\mathcal{S}$ pour classifier les différents états

Conclusion

L'apprentissage par renforcement profond est une approche très intéressante pour la résolution de problème de décision séquentielle comme celui de la mise en place d'un modèle opérateur en simulation numérique. Cependant, cette approche présente des lacunes sur l'efficience de l'utilisation des données et l'explicabilité du modèle final. Dans un contexte industriel, et en particulier celui de la défense nationale, la nécessité d'obtenir un résultat explicable avec un coût calculatoire raisonnable ce qui est très critique dans la stratégie d'investissement matériel et logiciel.

Les MAE sont déjà fortement présentes pour résoudre ce type de problème, car elles montrent de bonnes performances tout en restant explicables. En outre, l'utilisation couplée d'une MAE avec le DRL est une approche efficace et déjà largement utilisée dans la littérature pour ajouter des connaissances extérieures afin d'aider le processus d'apprentissage. En effet, cela permet d'accélérer la convergence et/ou d'augmenter l'interprétabilité du modèle final soit par des restrictions de l'espace d'action ou bien par des processus d'apprentissage plus compréhensibles.

Nous avons donc défini un cadre théorique pour l'incorporation d'une MAE au sein d'un processus d'apprentissage par renforcement. Ainsi, ces définitions permettent d'identifier les contraintes et les gains potentiels en fonction du placement du couplage et nous les avons utilisées pour replacer plusieurs approches de l'état de l'art. En outre, cela a permis d'identifier des couplages algorithmiques intéressants, et non explorés jusqu'à présent. Tout d'abord, la mise en place de nouvelles métriques utilisant la structure d'une MAE afin d'avoir plus d'informations sur l'état courant de la politique au cours de l'apprentissage. En outre, l'utilisation d'une MAE politique semble également être une approche intéressante pour tirer profit de connaissances a priori sur le comportement souhaité. Finalement, l'utilisation d'information supplémentaire non accessible pour le modèle final peut permettre de simplifier et d'améliorer la mise en place de MAE et donc d'améliorer les gains d'explicabilité et de performance d'un couplage. Ces différentes approches proposées sont en cours de développement sur l'environnement Lunar Lander, dans un but de résolution de la problématique industrielle exposée initialement.

REFERENCES

[1] R. S. Sutton and A. G. Barto, “Reinforcement Learning : An Introduction, ”, Second Edition. Cambridge, Massachussetts : The MIT Press, 2018, isbn : 978-0-262-03924-6.

[2] V. Mnih, K. Kavukcuoglu, D. Silver et al. “Human-level control through deep reinforcement learning, ” Nature, 518 (7540):529–533, 2015

[3] D. Silver, J. Schrittwieser, K. Simonyan et al., “Mastering the game of Go without human knowledge, ” Nature, t. 550, no 7676, p. 354-359, 2017.

[4] D. Silver, S. Singh, D. Precup R. S. Sutton, ”Reward is enough, “ Artificial Intelligence, t. 299, p. 103 535, 2021, issn : 00043702.

[5] A. Gupta, A. Pacchiano, Y. Zhai, S. Kakade and S. Levine, “Unpacking Reward Shaping : Understanding the Benefits of Reward Engineering on Sample Complexity, ” in Advances in Neural Information Processing Systems, 2022, p. 15 281-15 295.

[6] T. Zahavy, N. Ben-Zrihem and S. Mannor, “ Graying the black box : Understanding DQNs, ” in Proceedings of The 33rd International Conference on Machine Learning, New York, New York, USA : PMLR, 2016, p. 1899-1908.

[7] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup and D. Meger, “ Deep Reinforcement Learning That Matters, ” Proceedings of the AAAI Conference on Artificial Intelligence, t. 32, no 1, 29 avr. 2018, issn : 2374-3468, 2159-5399.

[8] G. Dulac-Arnold, N. Levine, D. J. Mankowitz et al. “ An empirical investigation of the challenges of real-world reinforcement learning, ” arXiv : 2003.11881, preprint 2020.

[9] W. Fedus, P. Ramachandran, R. Agarwal et al., “ Revisiting Fundamentals of Experience Replay, ” arXiv : 2007.06700, preprint 2020.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford and O. Klimov. « Proximal Policy Optimization Algorithms. » arXiv : 1707.06347.

[11] T. Haarnoja, A. Zhou, P. Abbeel and S. Levine. “ Soft Actor-Critic : Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor, ” arXiv : 1801.01290.

[12] A. Aubret, L. Matignon and S. Hassas. “ A survey on intrinsic motivation in reinforcement learning, ” arXiv : 1908.06976.

[13] . Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel and J. Peters, “ An Algorithmic Perspective on Imitation Learning, , ” Foundations and Trends in Robotics, t. 7,no 1-2, p. 1-179, 2018, issn : 1935-8253, 1935-8261.

[14] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. Taylor and P. Stone, “ Curriculum Learning for Reinforcement Learning Domains : A Framework and Survey, ” Journal of Machine Learning Research, 2021.

[15] Z. Zhu, K. Lin, A. K. Jain and J. Zhou. “ Transfer Learning in Deep Reinforcement Learning :A Survey, ” arXiv : 2009.07888.

[16] G.-C. Kang and Y. Lee, « Finite State Machine-Based Motion-Free Learning of Biped Walking, » IEEE Access, t. 9, p. 20 662-20 672, 2021, issn : 2169-3536.

[17] S. Hwang, K. Lee, H. Jeon and D. Kum, “ Autonomous Vehicle Cut-In Algorithm for Lane-Merging Scenarios via Policy-Based Reinforcement Learning Nested Within Finite-State Machine, ” IEEE Transactions on Intelligent Transportation Systems, t. 23, no 10, p. 17 594-17 606, oct. 2022, issn : 1524-9050, 1558-0016.

[18] R. T. Icarte, T. Q. Klassen, R. Valenzano and S. A. McIlraith, “ Using Reward Machines for High-Level Task Specificationand Decomposition in Reinforcement Learning, ” Proceedings of the 35 th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

[19] D. Furelos-Blanco, M. Law, A. Russo, K. Broda and A. Jonsson, “ Induction of Subgoal Automata for Reinforcement Learning, ” Proceedings of the AAAI Conference on Artificial Intelligence, t. 34, no 04, p. 3890-3897, 2020, issn : 2374-3468, 2159-5399

[20] R. T. Icarte, R. Valenzano, E. Waldie, M. P. Castro, T. Q. Klassen and S. A. McIlraith, “ Learning Reward Machines for Partially Observable Reinforcement Learning, ” in Advances in Neural Information Processing Systems, 2019.

[21] C. Glanois, P. Weng, M. Zimmer et al. “ A Survey on Interpretable Reinforcement Learning, ” arXiv : 2112.13112.

[22] J. E. Hopcroft, R. Motwani and J. D. Ullman, “ Introduction to Automata Theory, Languages, and Computation, ” 3rd ed. Boston : Pearson/Addison Wesley, 2007, 535 p., isbn : 978-0-321-45536-9 978-0-321-46225-1 978-0-321-45537-6.

[23] G. Brockman, V. Cheung, L. Pettersson et al. « OpenAI Gym. » arXiv : 1606.01540.

[24] https://gymnasium.farama.org/environments/box2d/lunar_lander/#lunar-lander

[25] B. Baker, I. Kanitscheider, T. Markov et al. “ Emergent Tool Use From Multi-Agent Autocurricula, ” arXiv : 1909.07528.

[26] D. Chen, B. Zhou, V. Koltun and P. Krähenbühl, “ Learning by Cheating, ” Proceedings of Machine Learning Research, t. 100, p. 66-75, 2020.

An investigation of knowledge distillation methods for underwater image semantic segmentation

Gabriel Guéganno* †, Ayoub Karine§, Thibault Napoléon†, Franck Florin* and Ayman Alfalou‡
* Thales DMS, 525 Route des Dolines, 06560 Valbonne, France.
† LabISEN, Vision-AD, ISEN Yncréa Ouest, 20 rue Cuirassé Bretagne, 29200 Brest, France.
‡ LabISEN, LSL, ISEN Yncréa Ouest, 33 Quater Chemin du Champ de Manoeuvre, 44470 Carquefou, France.
§ Université Paris Cité, LIPADE, F-75006 Paris, France.
gabriel.gueganno@isen-ouest.yncrea.fr

***Abstract*—Autonomous Underwater Vehicles (AUV) are key in seabed warfare applications. Seabed observation can be realized with an embedded camera processed with Artificial Intelligence (AI) that performs real-time semantic segmentation of the underwater images; this requires high performances and light-weight neural networks. However, AUV embedded computing capacity is limited and we cannot consider using a large network to obtain high performances. Knowledge distillation (KD) is a method for teaching a lightweight network with a large pre-trained supervisor network. This makes KD an ideal solution for training a network both highly efficient and capable of operating in real time embedded on a AUV. In this context, we evaluate different semantic segmentation network architectures, initially designed for urban datasets such as Cityscapes, over an underwater dataset: the semantic Segmentation of Underwater IMagery (SUIM) dataset. In addition, we evaluate the capacity of several KD methods to transfer knowledge in the underwater image domain with the SUIM dataset.**

Experiments demonstrate that the change of domain from urban scenes to underwater scenes achieves good results, both for semantic segmentation and KD.

***Index Terms*—knowledge distillation, semantic segmentation, autonomous underwater vehicle**

I. Introduction

The underwater domain holds a strategic importance for defense applications. In particular, the surveillance of the seabed is crucial for maintaining national security as it enables the detection of suspicious objects like mines or other threats. In this context, AUV play an important role in monitoring this environment, ensuring that security forces can respond swiftly and effectively to any underwater threat. However, those AUV need to be equipped correctly to move underwater without difficulty, and to detect threats.

Vision in underwater context can be performed using AI techniques applied to images/videos acquired by underwater acquisition systems. The sonar imagery approach can be very efficient as it can work from long distance to the seabed and with a good resolution [1]. Synthetic aperture sonar can be used to get high-resolution images of the sea bed. An alternative solution is to use an underwater optical imaging system to get to know the environment around the AUV [2]. Optical images in an underwater context have some drawbacks, specifically the effective distance, but they allow gathering real-time vision of the environment in front of the AUV, and some works have managed to improve underwater performances of optical devices [3].

With prior treatment, the AUV can use the video information to navigate. This treatment can be performed by deep learning with semantic segmentation algorithms. However, this raises a question about real time usability. Deep learning is sometimes not well suited to this application. From this perspective, it is of paramount importance to use methods of compression to obtain light-weight and real time neural networks to compute the semantic segmentation of the environment of the AUV. In addition, it goes together with the notion of ecodesign. A compressed network will require less energy per inference and it can therefore reduce the negative environmental impact of the AUV.

Semantic segmentation, which aims to predict a class to every pixel of an image, is an important topic in autonomous navigation. Autonomous driving and AUV navigation require indeed the knowledge of the environment to move correctly and semantic segmentation provides this type of information. Semantic segmentation has made great strides thanks to advances in Deep Learning, mainly in the urban and aerial domains [4], with databases such as ADE20K [5] or Cityscapes [6]. More occasionally, a few adaptations have also been made in the underwater domain [2], [7].

With the evolution of research in semantic segmentation, two families of neural networks have emerged: one based on the Fully Convolutional Network (FCN) architecture [8] and another based on the Vision Transformer (ViT) architecture [9]. Among the architectures that took the FCN as a base, some of them achieved very good performances such as PSPNet [10] or DeepLab [11]. On the side of the ViT-based architecture, we can cite the Segformer architecture [12] that achieved stunning performances on the semantic segmentation task. However, regardless of the architecture used, those good performances were obtained by wide architectures with many parameters. This limits the possibilities of using these architectures for real-time purposes. The computing capability of autonomous AUV is greatly limited, thus using the best performing semantic segmentation architectures is not an option. In addition, large neural network inference on images after being trained is energy consuming, and this must also be taken into account to embed a neural network in an AUV with limited energy

capacity.

However, one can surpass those limitations by various methods of neural network compression: pruning, quantization [13]. Those methods aim to reduce the size of the neural networks while limiting performance loss. A part of the research in this domain is dedicated to reducing the number of floating-point operations and the size of the models. One can do it either by designing light-weight models, by trying to remove useless parameters from wide architectures with network pruning or by using parameters quantization to reduce the size and the inference time of a neural network. These methods will be efficient in terms of computation time but often reduce the performance of the resulting compressed neural networks. Another field of research tries to overcome this issue: KD [14]. The main objective of KD is to train on a specific task a light-weight neural network, called the student network, under the supervision of a larger and more efficient neural network, called the teacher network. By doing so, one raises the performances of the student network without any augmentation of its size and number of parameters.

In this paper, we propose an investigation about the use of common semantic segmentation architectures and KD methods in an underwater context. To the best of our knowledge, KD were never exploited to train lightweight neural networks in an underwater context. Our investigation therefore focuses on the feasibility of using it in this domain. We focus on the following points:

- We evaluate the performances of three semantic segmentation architectures, designed for urban scene, on an underwater database SUIM [7]. With this part of the work we highlight the necessity to use KD for the training of light-weight neural network.
- We evaluate three KD methods, specifically, two conceived for FCN based architectures and one conceived for ViT based architecture.

In Part II, we present related work in the fields of semantic segmentation and KD. In Parts III and IV, we introduce some generalities of semantic segmentation and KD and describe the different methods considered for the investigation. Finally, we present and analyze the results obtained with these methods on the SUIM database in Part V.

II. Related Work

A. Semantic segmentation

Semantic segmentation is among the most important fields of computer vision with the constant development of autonomous driving and aerial surveillance. The principle behind semantic segmentation is the auto-encoder: an architecture composed of an encoder and a decoder. Among the most basic architectures of semantic segmentation, we can find the Fully Convolutional Network (FCN) [8] that only use convolutions, pooling and skip connections. PSPNet [10], which is a FCN, introduces the pyramid pooling that is a method to focus on the context by gathering information from different scales. The exclusive use of the convolution can cause problems related to receptive fields. When using conventional convolutions, the collection of information between two distant areas of an image can prove difficult due to the low receptive fields of the convolution. An answer to this issue is the atrous/dilated convolution used for example in DeepLab [11] and Dilation [15], which are both FCN. This type of convolution allows aggregating information from a wider zone without using spatial pooling. It has to be noted that all these methods are based on convolutional neural networks such as ResNet [16] or VGG [17] if one needs good performances, or MobileNet if one wants low latency computation.

Another way to tackle the receptive field issue is using a self-attention-based architecture, and specifically a ViT based architecture [9]. ViT uses the self-attention to gather information of the whole images given as input to compute the feature maps, even in the first layers of the network. ViT was originally designed for tasks as detection or classification, but recent works used this architecture for semantic segmentation. A solution adopted by the SEgmentation TRansformer (SETR) architecture [18] to use ViT for this task is to keep the encoder and add a simple decoder. Other architectures like Swin Transformer or Segformer [12] complexify the Transformer encoder by splitting it in several Transformer block. As for classical CNN encoders, each Transformer block computes a feature map with decreasing spatial size and increasing number of channels as one goes deeper in the encoder. To go further, both architectures propose several encoders with different sizes. Other methods propose using the Transformer as a decoder to compute the semantic segmentation. Mask2Former [19] for example uses a variation of the Transformer with masked self-attention to compute the semantic segmentation.

However, both convolution-based and Transformer-based architectures are often quite cumbersome. This implies the need to reduce their size so that they can be embedded in AUV.

B. Knowledge distillation

The basic concept of KD was introduced with the classification task by Hinton *et al.* [14], by comparing class probabilities of teacher and student models. The main objective of KD is to find a trade-off between the performance of the student model after the distillation and its size. According to Yang *et al.* [20], we can separate KD in three different types: response-based, feature-based and relation-based KD. Firstly, response-based KD corresponds to the distillation of information between the outputs of teacher and student models. Secondly, feature-based KD goes deeper in the networks by distilling knowledge directly between feature maps of teacher and student models. Finally, relation-based KD is a bit more complex as it distills knowledge between different samples of the training dataset or even different layers of the teacher and student models. With the success of KD to obtain performing lightweight classification models, this approach was also applied to the semantic segmentation task.

Among the first work done in the domain of FCN distillation, Xie *et al.* [21] propose to distill two types of knowledge

from the teacher network to the student networks simultaneously: first, the zero-order knowledge, the pixel probabilities, and secondly, the first-order knowledge, the sum of differences between a pixel and its eight neighbors. He *et al.* [22] use the knowledge from affinity maps to distill long-range information from the teacher's features to the student, it is feature-based KD. To tackle inconsistency caused by the difference in terms of shape between the teacher's and the student's features, a pre-trained auto-encoder is used to transform the teacher's features. Lui *et al.* [23] propose structural KD in two different ways. First, a pairwise distillation which aims to transfer pairwise pixel similarity to the student network, and secondly, a holistic distillation that uses a generative adversarial learning approach to align the segmentation map computed by the student network on the one of the teacher network. The Cross Image Relational Knowledge Distillation method (CIRKD) proposed by Yang *et al.* [24] is a relation-based KD that distillates two types of knowledge pixel-to-pixel information and pixel-to-region information. The distillation is performed either on mini-batches or on a memory bank composed of information extracted from the teacher network during the previous steps of the training. With the Channel-Spatial Knowledge Distillation method (CSKD), Karine *et al.* [25] used the similarity between the channels and the pixels of the intermediate feature maps to transfer knowledge from the teacher network to the student network.

As ViT-based architectures are more recent than FCN-based architectures, fewer methods have been implemented. Liu *et al.* propose with TransKD [26] a way to make profit of the specificities of the ViT architecture, and more specifically the Segformer [12] architecture. In the encoding part of the architecture, we find two types of information, the patch embeddings, that are fed into the different Transformer blocks, and the corresponding output feature maps. The distillation is performed between the patch embeddings and the feature maps of the student network and the teacher network.

III. Considered Semantic Segmentation Architectures

A. Principle

The objective of semantic segmentation is to assign to each pixel of an image a class prediction. The networks used for this task are generally composed of two parts. Firstly, an encoder that extracts from the input image a feature map $f \in \mathbb{R}^{d \times w \times h}$, where w, h and d denote respectively the height, the width and the number of channels of the feature map. Secondly, a decoder that computes a logit prediction $z \in \mathbb{R}^{C \times W \times H}$ from the feature map, where W and H denote the height and the width of the image and C denote the number of classes considered for the prediction. Finally, the semantic segmentation of the input image $y \in \mathbb{R}^{C \times W \times H}$ is obtained pixel by pixel by applying successively a softmax function, to get the probability distribution corresponding to the pixel, and an argmax function, to get the class prediction of the pixel.

For a classical semantic segmentation training, the loss used to compare each pixel of the logit prediction of the network with the corresponding ground truth is the cross-entropy measurement:

$$\mathcal{L}_{CE} = -\frac{1}{HW} \sum_{i}^{HW} CE\left(\sigma\left(z_i\right), y_i^{GT}\right) \tag{1}$$

Here, y_i^{GT} denotes the ground truth label of the i-th pixel of the image, z_i denotes the i-th pixel of the network logit prediction, σ denotes the softmax function and CE denotes the cross-entropy loss.

B. Considered methods

1) FCN: The first FCN-based semantic segmentation architecture considered is PSPNet. This semantic segmentation architecture capture global contextual information based on a pyramid pooling module. This module extracts features at multiple spatial scales, enabling the network to understand the overall structure of the scene while retaining the finest details. PSPNet combines these multi-scale features via a pyramid parsing module, which improves the representation of the overall context in the final feature map.

The second FCN-based semantic segmentation architecture considered in this investigation is DeepLabV3. This architecture uses atrous/dilated convolutions to extract features at different levels of resolution while maintaining high spatial resolution. DeepLabV3 also incorporates an improved version of the spatial pyramid pooling technique, called Atrous Spatial Pyramid Pooling (ASPP), to efficiently capture contextual information at multiple scales. By combining these features, DeepLabV3 improves segmentation performance, while mitigating the problem of loss of fine detail in segmentation objects.

DeepLabV3 and PSPNet have proven their effectiveness in the field of semantic segmentation by achieving state-of-the-art performances across various benchmark. Given their widespread adoption and established performance, these architectures provide a strong basis for further developments in knowledge distillation.

For both architectures, feature extraction is performed using a ResNet18, ResNet101 or MobileNetV2 convolutional neural network.

2) ViT: The ViT-based semantic segmentation architecture considered in this investigation is Segformer. Segformer is a semantic segmentation architecture that combines the strengths of FCN-based and ViT-based architectures. It uses a hierarchical structure with a multistage feature extraction process, where each stage applies a mix of convolutional layers and Transformer encoders. This design allows Segformer to efficiently capture both local and global contextual information. There are several versions of Segformer, depending on the size of the part dedicated to feature extraction. Unlike traditional ViT-based architectures, some versions of Segformer are lightweight and computationally efficient, making it suitable for real-time applications and knowledge distillation.

IV. Considered Knowledge Distillation Methods

A. Principle

When using KD to train a semantic segmentation network, two losses are generally considered. The first loss is the cross-entropy loss, presented in equation (1), used on the student network logit prediction. The second loss compares information between the teacher network and the student network. According to the method, this KD loss $\mathcal{L}_{KD}$ can be divided into several sub-losses, and we will see that it is the case for the different methods discussed in this paper. Generally speaking, the global loss $\mathcal{L}_{global}$ to train student can be formulated as follows:

$$\mathcal{L}_{global} = \mathcal{L}_{CE} + \lambda_{KD} \times \mathcal{L}_{KD} \tag{2}$$

Here, λ_{KD} denotes a coefficient that weights KD loss against the cross-entropy loss.

Inspired by Hinton *et al.*, a basic way to formulate the KD loss for semantic segmentation is to use the Kullback-Leibler divergence to compare the logit predictions of the teacher z^T and the student z^S. This loss can be formulated as follows:

$$\mathcal{L}_{KD} = \frac{1}{HW} \sum_{i}^{HW} KL\left(\sigma\left(\frac{z_i^S}{T}\right), \sigma\left(\frac{z_i^T}{T}\right)\right) \tag{3}$$

Here, KL denotes the Kullback-Leibler divergence and T denotes a temperature. With this basic loss function, the knowledge is distilled from the teacher output to the student output. Thus, KD is response based in this case.

B. Considered methods

In this section, the precise description of the different KD methods considered for this investigation has been placed in the appendices A, B and C.

1) Distillation from FCN: The first FCN-based method we consider for this investigation is CIRKD, proposed by Yang *et al.* [24]. This KD method is relation- and feature-based. It means that information from different images is used for the distillation during one step of the training, and particularly the feature maps extracted from the encoder of the semantic segmentation network are used. Three different distillations are conducted during the training of a student network with CIRKD, in addition to the classical KD, presented in equation (3).

The details of the method are presented in Appendix A. The complete distillation loss of the CIRKD method if formulated as follows:

$$\begin{aligned}\mathcal{L}_{CIRKD} =& \mathcal{L}_{CE} + \mathcal{L}_{KD} + \lambda_1 \mathcal{L}_{CIRKD}^{p2p-batch} \\ &+ \lambda_2 \mathcal{L}_{CIRKD}^{p2p-memory} + \lambda_3 \mathcal{L}_{CIRKD}^{p2r-memory}\end{aligned} \tag{4}$$

Here, $\mathcal{L}_{CIRKD}^{p2p-batch}$, $\mathcal{L}_{CIRKD}^{p2p-memory}$ and $\mathcal{L}_{CIRKD}^{p2r-memory}$ denote the three distillation losses proposed by the method. The losses $\mathcal{L}_{CE}$ and $\mathcal{L}_{KD}$ are the ones presented in equations (1) and (3) respectively. Finally, λ_1, λ_2 and λ_3 are weights set to 1, 0.1 and 0.1 respectively, as proposed in the original paper.

The second FCN-based method we consider for this investigation is CSKD, proposed by Karine *et al.* [25]. This method is feature- and response-based: information from the output of the teacher network and from intermediate calculations are used to distill information to the student. Two different distillations are conducted in addition to the classical KD, presented in equation (3). For each distillation one uses a dedicated module to catch information of feature maps, a Channel self-Attention Module (CAM) and a Position self-Attention Module (PAM).

The details of the method are presented in Appendix B. The complete distillation loss of the CSKD method if formulated as follows:

$$\begin{aligned}\mathcal{L}_{CSKD} =& \mathcal{L}_{CE} + \lambda_{KD} \mathcal{L}_{KD} \\ &+ \lambda_{PAM} \mathcal{L}_{CSKD}^{PAM} + \lambda_{CAM} \mathcal{L}_{CSKD}^{CAM}\end{aligned} \tag{5}$$

Here, $\mathcal{L}_{CSKD}^{PAM}$ and $\mathcal{L}_{CSKD}^{CAM}$ denote the two distillation losses proposed by the method. The losses $\mathcal{L}_{CE}$ and $\mathcal{L}_{KD}$ are the ones presented in equations (1) and (3) respectively. Finally, λ_{PAM} and λ_{CAM} are weights for the PAM and CAM losses set to 0.8 and 0.3 respectively, as proposed in the original paper.

2) Distillation from ViT: The ViT-based method we consider for this investigation is TransKD, proposed by Liu *et al.* [26]. This method is specific to the Segformer semantic segmentation architecture. As this architecture is based on the ViT architecture, two types of features can be found in the encoder: classical feature maps and patch embeddings. The method proposes one distillation for each type of feature. More specifically three variations of the patch embedding distillation are proposed.

The details of the method are presented in Appendix C. The complete distillation loss of the TransKD method if formulated as follows:

$$\mathcal{L}_{TransKD} = \mathcal{L}_{CE} + \mathcal{L}_{TransKD}^{PE} + \mathcal{L}_{TransKD}^{F} \tag{6}$$

Here, $\mathcal{L}_{TransKD}^{PE}$ and $\mathcal{L}_{TransKD}^{F}$ denote the two distillation losses proposed by the method. Depending on the variation of TransKD, $\mathcal{L}_{TransKD}^{PE}$ takes different forms. The loss $\mathcal{L}_{CE}$ is the one presented in equation (1).

V. Experimentation

A. Dataset

The underwater dataset we use for the investigation is SUIM [7]. This dataset is composed of 1635 underwater images annotated for semantic segmentation among 8 different classes. As proposed by the authors of the dataset, images are partitioned in two sub-datasets, 1525 are dedicated to the training phase and 110 are dedicated to the validation and testing phases. The size of images fluctuates from 375×590 to 960×1280 pixels, but most of the images ($\sim$80%) have a size of 480×640 pixels.

B. Experiments Details

1) Semantic Segmentation Architectures: For the different experiments, we use three semantic segmentation architectures. We use a DeepLabV3 and a PSPNet both with a ResNet101 encoder as a teacher network for the CIRKD

and the CSKD methods as they were designed for FCN architectures. The student networks used with these methods are also DeepLabV3 and PSPNet with different encoders, ResNet18 and MobileNetV2. For the experiments performed on the TransKD method, we use as a Teacher a Segformer with encoder MiT-B4 or MiT-B2 as TransKD is designed for this architecture. The student network is a Segformer with a MiT-B0 encoder.

2) Metric: The metric we use to evaluate the different trained networks is the mean intersection over union ($mIoU$). For a given class c, the IoU is formulated as follows:

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{7}$$

Here, A denotes the area assigned to the class by the evaluated network on the prediction of an image and B denotes the area of the same class on the ground truth semantic segmentation of the image. To obtain the $mIoU$ we simply mean the IoU over all the classes.

3) Training Setup: The training setup is the same for all the experiments performed, with the exception of a few points or training for architectures based on FCN or ViT. In terms of data augmentation on the input images, we apply random horizontal flip, random scaling from 0.5 to 2 and crop of size 240×320. A stochastic gradient descent (SGD) with a momentum of 0.9 is used to update the parameters of the networks during training. The total number of iterations is 40000, and the trained network is tested on the validation dataset every 800 iterations. The batch size is set to 16, but experiments were also made with a batch size of 8. The learning rate is where there is a difference between FCN- and ViT-based architectures. For the FCN-based architectures, the initial learning rate is set to 0.02 while it is set to 0.0002 for the ViT-based architectures. In both cases the learning rate is updated at each iteration by multipling it by $(1 - \frac{iter}{Total_{iter}})^{0.9}$, where $iter$ denotes the current iteration and $Total_{iter}$ denotes the total number of iterations.

C. Semantic Segmentation Results

In this section, we compare the results of the different semantic segmentation architectures that we will use either as students or teacher in the following.

In Table I and Table II, we show the results on the dataset SUIM of the different FCN-based and ViT-based semantic segmentation architectures respectively. A first observation we can make is that regardless of the semantic segmentation architecture family, a logical trend can be observed. The more parameters a neural network has, the better its performance. The only exception is the DeepLabV3 associated with a MobileNetV2 encoder that has better performances than the PSPNet associated with a ResNet18 encoder although it has fewer parameters.

If we now compare the FCN- and ViT-based architectures, it clearly appears that the different Segformers outperform the convolutional networks. If we look first at the performances of the Segformer MiT-B0, it outperforms FCN-based architectures with a comparable number of parameters. If we look now at the performances of the two other Segformer architectures, they clearly outperform every other architectures. We particularly note the Segformer MiT-B2, which, with a reasonable number of parameters, achieves high performances.

TABLE I
RESULTS ON SUIM OF SEMANTIC SEGMENTATION ON THE DIFFERENT FCN-BASED ARCHITECTURES. R AND MN DENOTE RESNET AND MOBILENET ENCODERS RESPECTIVELY.

Method	Params (M)	mIoU (%)
PSPNet R101	68.1	69.50
DeepLabV3 R101	61.1	69.66
DeepLabV3 R18	13.6	67.72
PSPNet R18	12.9	64.60
DeepLabV3 MNV2	3.2	66.03

TABLE II
RESULTS ON SUIM OF SEMANTIC SEGMENTATION ON THE DIFFERENT VIT-BASED ARCHITECTURES.

Method	Params (M)	mIoU (%)
Segformer MiTB4	64.0	72.20
Segformer MiTB2	27.4	70.92
Segformer MiTB0	3.7	68.37

However, it is worth noting that for both FCN- and ViT-based architectures, there is a relatively large gap between the performances of the lightest and heaviest neural networks. This justifies the use of KD in this context, to improve the performances of the DeepLabV3 ResNet18, the DeepLabV3 MobileNetV2, the PSPNet ResNet18 and the Seformer MiT-B0.

D. Knowledge Distillation Results

1) Results: In this section, we analyze the results obtained on the SUIM database with the different KD methods presented in Part IV. In Table III and Table IV, we present the results of the different KD methods applied to the neural network studied in the previous part. The Table III regroups the results of the CIRKD and the CSKD methods applied to the FCN-based semantic segmentation neural networks DeepLbV3 and PSPNet, while Table IV regroups the results of the different TransKD methods applied to the ViT-based semantic segmentation neural network Segformer.

It is worth noting that the results obtained with those methods on the Cityscapes dataset are very good, for all pairs of teacher and student CIRKD, CSKD and TransKD methods allow improving the performances of the student.

For the FCN-based architectures, depending on the choice of the student and the teacher, the results can be significantly different. If we first look at the training performed with the DeepLabV3 ResNet101 as a teacher, using CSKD seems to be more pertinent associated with this teacher as every student trained with CSKD and supervised by it outperforms the same student trained with CIRKD. Moreover, for all students,

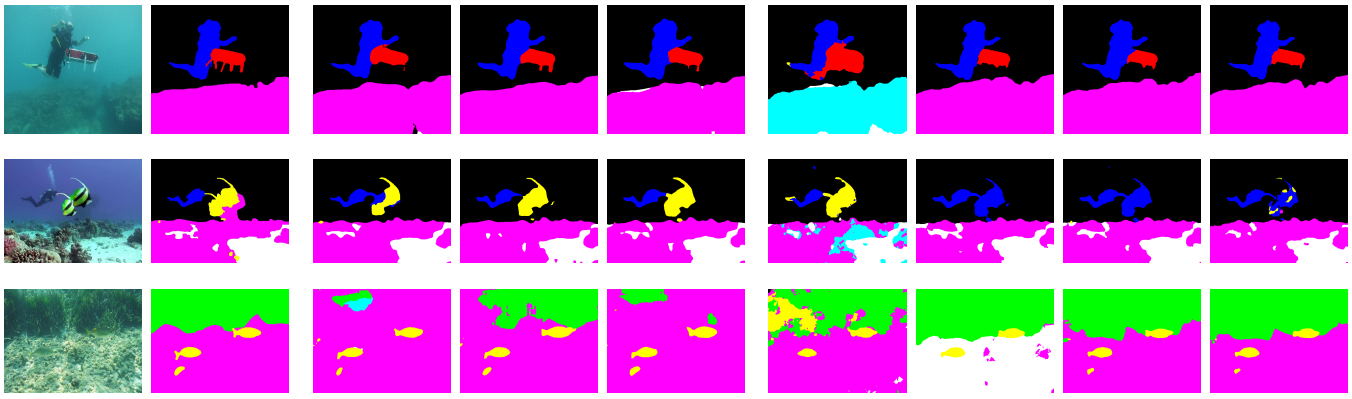


Fig. 1. Qualitative semantic segmentation results on the SUIM dataset. First two columns represent the considered image and the corresponding ground truth. Next three columns correspond to the results of CIRKD and CSKD methods using a student DeepLabV3 ResNet18. Last four columns correspond to the results of the three variations of TransKD methods using a student Segformer MiT-B0.

TABLE III
RESULTS ON SUIM OF KD ON THE DIFFERENT FCN-BASED ARCHITECTURES. THE LETTER T DENOTES THE TEACHER NETWORKS, AND THE LETTER S DENOTES THE STUDENT NETWORKS. R AND MN DENOTE RESNET AND MOBILENET ENCODERS RESPECTIVELY.

Method	Params (M)	mIoU (%)
T: DeepLabV3 R101	**61.1**	**69.66**
S: DeepLabV3 R18	13.6	67.72
+ CIRKD		65.73
+ CSKD		68.57
S: DeepLabV3 MNV2	3.2	66.03
+ CIRKD		64.66
+ CSKD		65.61
S: PSPNet R18	12.9	64.60
+ CIRKD		64.41
+ CSKD		66.86
T: PSPNet R101	**68.1**	**70.08**
S: DeepLabV3 R18	12.9	64.60
+ CIRKD		67.75
+ CSKD		66.70
S: PSPNet R18	13.6	67.72
+ CIRKD		69.98
+ CSKD		68.63

TABLE IV
RESULTS ON SUIM OF KD ON THE DIFFERENT VIT-BASED ARCHITECTURES. THE LETTER T DENOTES THE TEACHER NETWORKS, AND THE LETTER S DENOTES THE STUDENT NETWORKS.

Method	Params (M)	mIoU (%)
T: Segformer MiT-B2	**27.4**	**70.92**
S: Segformer MiT-B0	4.6	68.37
+ TransKD-Base		67.71
+ TransKD-EA		68.31
+ TransKD-GLMixer		68.58
T: Segformer MiT-B4	**64.0**	**72.20**
S: Segformer MiT-B0	4.6	68.37
+ TransKD-Base		62.15
+ TransKD-EA		62.77
+ TransKD-GLMixer		65.11

the performance obtained after training with CIRKD was lower than that obtained with conventional training without KD. For its part, with the exception of the DeepLabV3 MobileNetV2 student, all the training carried out with CSKD improved the performance of the student. In conclusion, the best performance is obtained from CSKD with a DeepLabV3 ResNet101 teacher. However, the best results are obtained when the same architecture is used as a teacher and student, in this case a DeepLab ResNet101 teacher and a DeepLab ResNet18 student.

Regarding to the training performed with the PSPNet ResNet101 as a teacher, the results are completely different. This time the CIRKD method allows obtaining better results than the CSKD method. In both cases, the performances of the student trained with a KD method are better than those obtained with a student trained without KD. In the same way as above, the student with the same architecture as the teacher obtains the better performances, in this case the PSPNet ResNet18 with the PSPNet ResNet101 teacher.

As for the FCN-based architectures, the result of KD on ViT-based architecture is very dependent on the choice of the student and the teacher. If we consider first the Segformer MiT-B2 teacher, the results are mixed. For all the TransKD variations, the performances of the student trained with KD are very close to the results of the student trained without, and only the TransKD-GLMixer allows to barely exceed it.

The different trainings performed with the Segformer MiT-B4 teacher are quite disappointing. None of the methods works with this choice of teachers. In all cases, the performance of the student trained with TransKD is far inferior to that of the student trained alone. The process of KD in this case doesn't transmit information, but rather loses it.

We show in fig. 1 some results of KD on the test dataset of SUIM. A first observation can be made is that for all images, the DeepLabV3 ResNet18 student seems to offer smoother segmentation than the MiT-B0 Segformer. The most

obvious example is the boundary between classes. For the DeepLabV3 ResNet18 they are clearly defined, even if they sometimes stray from the class boundaries of ground truth. On the contrary, for the Segformer MiT-B0, boundaries are noisy, even if the zones correspond to the right classes in relation to the ground truth. After the application of the different KD methods, we observe for all students an amelioration in the segmentation. For the DeepLabV3 ResNet18, the segmentation is more accurate and for the Segformer MiT-B0, boundaries are less noisy. If we compare the results of the students after KD, similar results can be observed for the first image. For the next two images, there is confusion for some classes for all students. For the second image, a class is incorrectly predicted by the Segformer MiT-B0 students trained with TransKD, whereas it was correctly predicted without knowledge distillation. For the last image, it is the DeepLabV3 ResNet18 students trained with CIRKD and CSKD that fail to correctly predict a class in the background. We conclude, however, that in all cases, there is still room for improvement to get closer to the ground truth.

2) Interpretation: The poor results of the CIRKD method in our investigation can be explained by the choice of the dataset. This method is based on a memory bank composed of pixel and region embeddings collected during the training. However, the SUIM dataset is very uneven in terms of class distribution. For example, the class "Robot" is very underrepresented in the dataset, thus its representation in the memory bank can be erroneous or incomplete and it can negatively impact the training for this class.

Other results can be explained by the difference between the number of parameters of both networks and by the size of the dataset. As the SUIM dataset is relatively small in comparison to datasets such as Cityscapes, the student has a limited diversity in the knowledge that is transferred from the teacher to learn the task of semantic segmentation. If in addition to that, the difference in terms of parameter is too high between the teacher and the student, it is harder for the student to reproduce the behavior of the teacher network and it leads to poor performances. It is the case for the Segformer MiT-B0 trained under the supervision of the Segformer MiT-B4, and the DeepLabV3 MobileNetV2 trained under the supervision of the DeepLab ResNet101.

However, it appears that KD can be effective in an underwater context. The CSKD method shows good performance, especially when the student and the teacher share the same architecture. For the ViT-based architecture, the results are mixed. If the networks trained without KD achieve good performances, the best we can obtain with KD is to equalize those performances. However, the KD in the field of ViT-based architectures is still in development and improvement is expected in the future. Another way of improving results could be to expand the underwater dataset, either by finding new images, which would involve both finding and annotating these images, or by improving the data augmentation process upstream of training.

VI. Conclusion

This paper presents an investigation of different KD methods for underwater image semantic segmentation. An alternative to KD would have been to reduce the size of large, high-performing networks using the pruning or quantization methods, mentioned in the introduction. We have not yet considered these methods, but it could be interesting to compare them with KD to compare the performance obtained, both in terms of network compression and semantic segmentation quality.

The KD methods considered in this investigation were CIRKD and CSKD for the FCN-based architectures and TransKD for the ViT-based architectures. The results of the investigation show that KD methods originally designed for urban semantic segmentation applications, and therefore not taking into account certain aspects such as water turbidity, can be effective in an underwater context. However some limitations reduce the effectiveness of KD such as the limited number of annotated images of underwater scenes. The difference in the number of parameters between the teacher and the student turned out to be a point of interest. Too great difference added to the limited amount of data cancels out the benefits of KD. The aim of this investigation was to evaluate the ability to compress neural networks for embedding in AUV. We investigated KD showing that it is possible to reduce the size of the network by a factor of 5 for the FCN-based architectures, while limiting the degration in performance.

As a perspective, we are planning to introduce a new KD method which takes into account the problems mentioned above. Another perspective would be to find a better set of parameters for the training. For this study, the choice of parameters, in particular the λ parameters used to weight the different losses, were the same as those proposed in the original papers of the methods used. One way of improving the performance of the KD methods used could be to carry out a grid search to optimize these parameters.

Acknowledgment

This research is financially supported by the National Association for Research and Technology (ANRT). All works were done in collaboration between Thales and ISEN Yncréa Ouest which are part of the GIS CORMORANT.

References

[1] A. Karine, N. Lasmar, A. Baussard, and M. El Hassouni, "Sonar image segmentation based on statistical modeling of wavelet subbands," in *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*. IEEE, 2015, pp. 1–5.

[2] T. Le Pennec, M. Jridi, C. Dezan, A. Alfalou, and F. Florin, "Underwater exploration by AUV using deep neural network implemented on FPGA," in *Pattern Recognition and Tracking XXXI*, vol. 11400. SPIE, 2020, pp. 61–66.

[3] J. Hajjami, T. Napoléon, and A. Alfalou, "Adaptation of Koschmieder dehazing model for underwater marker detection," in *Pattern Recognition and Tracking XXXI*, vol. 11400. SPIE, 2020.

[4] A. Toker, M. Eisenberger, D. Cremers, and L. Leal-Taixé, "Satsynth: Augmenting image-mask pairs through diffusion models for aerial semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 27 695–27 705.

[5] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 633–641.

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

[7] M. J. Islam, C. Edge, Y. Xiao, P. Luo, M. Mehtaz, C. Morse, S. S. Enan, and J. Sattar, "Semantic segmentation of underwater imagery: Dataset and benchmark," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 1769–1776.

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.

[10] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2881–2890.

[11] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[12] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 077–12 090, 2021.

[13] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: A survey," *Neurocomputing*, vol. 461, pp. 370–403, 2021.

[14] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[15] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[18] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. S. Torr *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6881–6890.

[19] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1290–1299.

[20] C. Yang, X. Yu, Z. An, and Y. Xu, "Categories of response-based, feature-based, and relation-based knowledge distillation," in *Advancements in Knowledge Distillation: Towards New Horizons of Intelligent Systems*. Springer, 2023, pp. 1–32.

[21] J. Xie, B. Shuai, J.-F. Hu, J. Lin, and W.-S. Zheng, "Improving fast segmentation with teacher-student learning," in *British Machine Vision Conference*, 2018, p. 205.

[22] T. He, C. Shen, Z. Tian, D. Gong, C. Sun, and Y. Yan, "Knowledge adaptation for efficient semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 578–587.

[23] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang, "Structured knowledge distillation for semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2604–2613.

[24] C. Yang, H. Zhou, Z. An, X. Jiang, Y. Xu, and Q. Zhang, "Cross-image relational knowledge distillation for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 319–12 328.

[25] A. Karine, T. Napoléon, and M. Jridi, "Channel-spatial knowledge distillation for efficient semantic segmentation," *Pattern Recognition Letters*, 2024.

[26] R. Liu, K. Yang, A. Roitberg, J. Zhang, K. Peng, H. Liu, Y. Wang, and R. Stiefelhagen, "TransKD: transformer knowledge distillation for efficient semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[27] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *International conference on machine learning*. PMLR, 2019, pp. 3519–3529.

[28] P. Chen, S. Liu, H. Zhao, and J. Jia, "Distilling knowledge via knowledge review," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 5008–5017.

APPENDIX A
CIRKD

A. Mini-batch-based Pixel-to-Pixel Distillation

The first distillation is performed among a mini-batch of N images $\{x_n\}_{n=1}^N$. The objective is here to compute the similarity between intermediate features obtained from the images of the mini-batch for the student network and for the teacher network. Then the similarities of the student network are compared with the ones of the teacher network to obtain a loss measurement. For the mini-batch, one considers the spatially flattened intermediate feature maps on the network that are noted $\left\{f'_n \in \mathbb{R}^{d\times hw}\right\}_{n=1}^N$. For two images x_k and x_l of the mini-batch, with $k, l \in \{1, 2, \ldots, N\}$, the cross-image pairwise similarity matrix is $s_{kl} = f_k^T f_l$. Now we can compute the similarity between images of the mini-batch, a first loss can be formulated as follows:

$$\mathcal{L}_{CIRKD}^{p2p-batch} = \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N}\mathcal{L}^{p2p}\left(s_{kl}^S, s_{kl}^T\right) \tag{8}$$

with:

$$\mathcal{L}^{p2p}\left(s_{kl}^S, s_{kl}^T\right) = \frac{1}{hw}\sum_{i=1}^{hw} KL\left(\sigma\left(\frac{s_{kl,i}^S}{\tau}\right), \sigma\left(\frac{s_{kl,i}^T}{\tau}\right)\right) \tag{9}$$

Here, τ denotes a temperature, s_{kl}^S and s_{kl}^T denote the similarity between the k-th and the l-th feature maps of the mini-batch for the student network and the teacher network respectively, and for both student and teacher networks, $s_{kl,i}$ denote the i-th row of the similarity matrix.

B. Memory-based Pixel-to-Pixel Distillation

Mini-batch size is often small, this is why a second distillation is performed using a memory bank. The memory bank takes the form of a class aware pixel queue $\mathcal{Q}_p$ updated with a *first-in-first-out* strategy. For a class, every corresponding pixel prediction is similar, thus it is not useful to fill the memory bank with every pixel corresponding to this class. In this context, at each iteration of the training, only a small number of pixel embeddings extracted from the teacher feature map are used to update the pixel queue.

With the same notations as before, for an image x_n of a given iteration, the flatten intermediates feature maps of the student and teacher networks are respectively $f_n^S \in \mathbb{R}^{d\times hw}$ and $f_n^T \in \mathbb{R}^{d\times hw}$. One forms a matrix $V_p \in \mathbb{R}^{K_p\times d}$ composed of K_p pixel embeddings sampled from the pixel queue $\mathcal{Q}_p$.

Then, for both student and teacher, the similarity matrices between the corresponding feature map and the matrix V_p are computed as follows:

$$p^S = \left(V_p f_n^S\right)^T \in \mathbb{R}^{hw \times K_p}, p^T = \left(V_p f_n^T\right)^T \in \mathbb{R}^{hw \times K_p} \quad (10)$$

Finally, the knowledge from the teacher is distilled to the student by using the Kullback-Leibler divergence once again between both similarities:

$$\mathcal{L}_{CIRKD}^{p2p-memory} = \frac{1}{hw} \sum_{i=1}^{hw} KL\left(\sigma\left(\frac{p_i^S}{\tau}\right), \sigma\left(\frac{p_i^T}{\tau}\right)\right) \quad (11)$$

Here, for both student and teacher network, p_i denote the i-th row of the similarity matrix.

C. Memory-based Pixel-to-Region Distillation

The last distillation is similar to the memory-based pixel-to-pixel distillation, except this time instead of using a queue composed of pixel embedding, a region queue $\mathcal{Q}_r \in \mathbb{R}^{C \times N_r \times d}$ composed of region embeddings is used, where N_r is the length of the queue for each class. The region queue is composed of C channels, one for each class, and each channel is filled with N_r region embedding of size d. Instead of updating the queue with several pixel embeddings at each iteration, $\mathcal{Q}_r$ is updated with the mean value of all pixel embeddings for each class.

Here again, for an image x_n of a given iteration, the spatially flatten intermediates feature maps of the student and teacher networks are respectively $f_n^S \in \mathbb{R}^{d \times hw}$ and $f_n^T \in \mathbb{R}^{d \times hw}$. One forms this time a matrix $V_r \in \mathbb{R}^{K_r \times d}$ composed of K_r pixel embeddings sampled from the region queue $\mathcal{Q}_r$. Then, for both student and teacher, the similarity between the corresponding feature map and the matrix V_r is computed as follows:

$$r^S = \left(V_r f_n^S\right)^T \in \mathbb{R}^{hw \times K_r}, r^T = \left(V_r f_n^T\right)^T \in \mathbb{R}^{hw \times K_r} \quad (12)$$

Finally, the knowledge from the teacher is distilled to the student by using the Kullback-Leibler divergence once again between both similarities:

$$\mathcal{L}_{CIRKD}^{p2r-memory} = \frac{1}{hw} \sum_{i=1}^{hw} KL\left(\sigma\left(\frac{r_i^S}{\tau}\right), \sigma\left(\frac{r_i^T}{\tau}\right)\right) \quad (13)$$

D. Complete Distillation Process

Finally, the complete distillation loss of the CIRKD method if formulated as follows:

$$\begin{aligned} \mathcal{L}_{CIRKD} = &\mathcal{L}_{CE} + \mathcal{L}_{KD} + \lambda_1 \mathcal{L}_{CIRKD}^{p2p-batch} \\ &+ \lambda_2 \mathcal{L}_{CIRKD}^{p2p-memory} + \lambda_3 \mathcal{L}_{CIRKD}^{p2r-memory} \end{aligned} \quad (14)$$

Here, λ_1, λ_2 and λ_3 are weights set to 1, 0.1 and 0.1 respectively.

APPENDIX B
CSKD

A. CAM

The objective of the CAM is to capture information between the channels of a semantic segmentation network output. For this, the module computes a matrix to weight the feature map and select the channel information. This operation is done for both student and teacher networks and the resulting matrix are used for the distillation.

Given a feature map $f \in \mathbb{R}^{d \times h \times w}$, it is flattened spatially to get $f' \in \mathbb{R}^{d \times hw}$. To obtain the relation between the i-th and the j-th channels of the feature map f', the channel attention weights $\omega_{ji}^{CAM} \in \mathbb{R}$ are computed as follows:

$$\omega_{ji}^{CAM} = \frac{exp\left(\frac{f_j' \cdot f_i'}{\tau}\right)}{\sum_{i=1}^{C} exp\left(\frac{f_j' \cdot f_i'}{\tau}\right)} \quad (15)$$

Here, f_j' and f_i' denote respectively the j-th and the i-th rows of the feature map f', and τ denotes a temperature. Once the channel attention weights ω_{ji} are computed, one can obtain the weighted feature $e^{CAM} \in \mathbb{R}^{d \times h \times w}$, channel by channel, as follows:

$$e_j^{CAM} = \beta \sum_{i=1}^{d} \left(\omega_{ji}^{CAM} f_i'\right) + f_j' \quad (16)$$

The resulting channel of the final feature map is the original feature map channel added to the weighted sum of all channels multiplied by a parameter β learned during the training.

B. PAM

The objective of the PAM is to capture spatial information between the pixels of a semantic segmentation network output. As for the CAM, the starting point is the logit output of the network f. With two additional convolution layers, two new feature maps $a, b \in \mathbb{R}^{d_r \times h \times w}$ are computed from f, with $d_r < d$, and they are flattened spatially to obtain feature maps $a, b \in \mathbb{R}^{d_r \times hw}$. Then similarly to the CAM, position attention weights $\omega_{ji}^{PAM} \in \mathbb{R}$ are computed as follows:

$$\omega_{ji}^{PAM} = \frac{exp\left(\frac{a_i \cdot b_j}{\tau}\right)}{\sum_{i=1}^{C} exp\left(\frac{a_i \cdot b_j}{\tau}\right)} \quad (17)$$

Here, a_i and b_j denote respectively the j-th and the i-th rows of the feature map a and b, and τ denotes a temperature, the same used in equation (15). For the next step one computes, with another convolution layer, a new feature $c \in \mathbb{R}^{d \times h \times w}$ and flatten it spatially to obtain $c \in \mathbb{R}^{d \times hw}$. Now the position attention weights ω_{ji} are computed, one can obtain the weighted feature $e^{PAM} \in \mathbb{R}^{d \times h \times w}$, position by position, as follows:

$$e_j^{PAM} = \alpha \sum_{i=1}^{hw} \left(\omega_{ji}^{PAM} c_i\right) + a_j \quad (18)$$

The resulting position embedding of the final feature map at the position j is the pixel at the position j in the feature map

a added to the weighted sum of all the pixels of the feature map c multiplied by a parameter α learned during the training.

C. Complete Distillation Process

The metric used to compare the feature maps obtained with CAM and PAM for the student network and the teacher network is the centered kernel alignment (CKA) [27]. If one notes $e^{-,S}$ and $e^{-,T}$ the feature obtained with CAM or PAM for the student network and the teacher network respectively, the CKA metric is defined as follows:

$$CKA\left(e^{-,S}, e^{-,T}\right) = \frac{\left\|(e^{-,T})^T e^{-,S}\right\|_F^2}{\left\|(e^{-,S})^T e^{-,S}\right\|_F \left\|(e^{-,T})^T e^{-,T}\right\|_F} \quad (19)$$

Here $\|\cdot\|_F$ denote the F-norm. this metric has values between 0 and 1. From here, two losses can be defined, one for CAM and the other for PAM:

$$\mathcal{L}_{CSKD}^{CAM} = -log\left(CKA\left(e^{CAM,S}, e^{CAM,T}\right)\right) \quad (20)$$

$$\mathcal{L}_{CSKD}^{PAM} = -log\left(CKA\left(e^{PAM,S}, e^{PAM,T}\right)\right) \quad (21)$$

Finally, the complete distillation loss of the CSKD method if formulated as follows:

$$\begin{aligned}\mathcal{L}_{CSKD} =& \mathcal{L}_{CE} + \lambda_{KD}\mathcal{L}_{KD} \\ &+ \lambda_{PAM}\mathcal{L}_{CSKD}^{PAM} + \lambda_{CAM}\mathcal{L}_{CSKD}^{CAM}\end{aligned} \quad (22)$$

Here, λ_{PAM} and λ_{CAM} are weights for the PAM and CAM losses set to 0.8 and 0.3 respectively.

APPENDIX C
TRANSKD

A. Patch Embedding Distillation

For the patch embedding distillation, the basic method is called TransKD-Base. To perform the distillation one uses a module, composed of dense layers, on the student network patch embeddings to match the size of the teacher patch embeddings. Then a mean square error (MSE) loss is used to compare the student and teacher patch embeddings as follows:

$$\mathcal{L}_{TransKD}^{PE-Base} = \sum_{i=1}^{4} \lambda_{PE,i} MSE\left(\mathcal{M}_{PE-Base}\left(e_i^S\right), e_i^T\right) \quad (23)$$

Here, $e_i^S \in \mathbb{R}^{N\times d_i^S}$ and $e_i^T \in \mathbb{R}^{N\times d_i^T}$ denote the patch embedding of the i-th Transformer block of the encoder for the student and the teacher respectively, with $i \in \{1,2,3,4\}$, N the number of patches used for the patch embedding and d_i^S and d_i^T the patch embedding size of the i-th Transformer block for the student and the teacher respectively. Then, λ_{PE} denotes a weight vector set to $[0.1, 0.1, 0.5, 1]$. Finally, $\mathcal{M}_{PE-Base}$ denotes the dense module used to resize the student patch embedding.

The two other variations, TransKD-EA and TransKD-GLMixer, use the previous method as a base. For TransKD-EA, an embedding assistant is used in addition to the dense layer module to obtain a more adapted version of the student patch embedding. For the TransKD-GLMixer method, only the last patch embedding distillation is modified. In addition to the dense module, the student patch embedding passes through another module composed of convolutions and attention computations to mix global and local information. In both cases, the patch embedding loss is also computed with the MSE.

B. Feature Map Distillation

The feature map distillation is common to all three variations of TransKD. Despite the fact that it is not commonly done by most of feature-based distillation methods, information is here distilled across different depths of the Segformer encoder. For a given Transformer block of the student encoder, one notes f_i^S the output feature map of this block, with i the index of the Transformer block. To be compared with the corresponding feature map f_i^T of the teacher encoder, f_i^S is mixed with the feature map computed by the next Transformer block with a module noted $\mathcal{M}_F$ to obtain a new feature map g_i^S. The operation is done as follows:

$$g_i^S = \mathcal{M}_F\left(f_i^S, g_{i+1}^S\right) \quad (24)$$

Then, g_i^S passes through a convolution layer to obtain the final feature maps $f_i^{final,S} = conv_{3\times3}\left(g_i^S\right)$. Finally, for each Transformer block, $f_i^{final,S}$ is compared to f_i^T using the hierarchical context loss (HCL) [28]:

$$\mathcal{L}_{TransKD}^{Feature} = \sum_{i=1}^{4} \lambda_{F,i} HCL\left(f_i^{final,S}, f_i^T\right) \quad (25)$$

Here, λ_F denotes a weight vector set to $[1, 1, 1, 1]$.

C. Complete Distillation Process

Finally, the complete distillation loss of the TransKD method if formulated as follows:

$$\mathcal{L}_{TransKD} = \mathcal{L}_{CE} + \mathcal{L}_{TransKD}^{PE} + \mathcal{L}_{TransKD}^{F} \quad (26)$$

POPCORN : IA d'extraction d'information à partir de sources textuelles pour le renseignement militaire

Cédric Lopez
Emvista
34830 Jacou, France
cedric.lopez@emvista.com

Sylvain Verdy
Emvista
34830 Jacou, France
sylvain.verdy@emvista.com

Guillaume Gadek
Airbus Defence & Space
78990 Elancourt, France
guillaume.gadek@airbus.com

Maxime Prieur
Airbus Defence & Space
78990 Elancourt, France
maxime.prieur@airbus.com

Didier Schwab
Univ. Grenoble Alpes, CNRS, LIG
38000 Grenoble, France
didier.schwab@univ-grenoble-alpes.fr

Gilles Sérasset
Univ. Grenoble Alpes, CNRS, LIG
38000 Grenoble, France
gilles.serasset@imag.fr

Nakanyseth Vuth
Univ. Grenoble Alpes, CNRS, LIG
38000 Grenoble, France
nakanyseth.vuth@univ-grenoble-alpes.fr

***Abstract*—Le projet de recherche collaboratif POPCORN (Peuplement OPérationnel de bases de COnnaissances et Réseaux de Neurones) a pour objectif de porter à maturité des technologies d'extraction d'informations contenues dans des documents textuels. L'article présente nos contributions autour de l'un de nos cas d'usage "renseignement militaire" concernant trois questions : Quelles données utiliser pour l'entraînement de nos modèles d'intelligence artificielle ? Quelles informations extraire ? Quels modèles adopter ? Nous exposons dans cet article les résultats obtenus sur deux tâches : l'extraction d'entités d'intérêt et l'extraction des relations entre ces entités.**

***Index Terms*—extraction d'information, renseignement militaire, traitement automatique du langage naturel, intelligence artificielle**

I. Introduction

Le projet de recherche collaboratif POPCORN a pour objectif de porter à maturité des technologies d'extraction d'informations contenues dans des documents textuels. Ce projet de 36 mois est subventionné par l'Agence de l'Innovation de Défense (AID) et implique les entreprises Emvista et Airbus (Defence and Space) ainsi que le Laboratoire d'Informatique de Grenoble (LIG). Le consortium ainsi formé est fort de spécialistes du domaine de recherche de l'Extraction d'Information (EI) et plus généralement des techniques de Traitement Automatique du Langage Naturel (TALN). Le projet touche à sa fin et nous présentons dans cet article nos contributions à l'état de l'art scientifique ainsi que quelques retours d'expériences et réflexions autour de l'un de nos cas d'usage « renseignement militaire ».

La section II décrit ce cas d'usage et explicite les contraintes métiers et techniques ainsi que les objectifs à atteindre. Un état de l'art scientifique relatif aux verrous à lever dans le cadre de ce cas d'usage est dressé en section III. Pour chacun des verrous identifiés, nous détaillons quelles approches ont été expérimentées pour contribuer à leur levée et les résultats obtenus. Enfin, nous présentons en section IV l'intégration de nos résultats dans différents démonstrateurs.

Le projet POPCORN duquel ces réflexions sont issues a bénéficié d'une subvention de l'Agence Innovation Défense (AID) et de l'accompagnement de la Direction Générale de l'Armement (DGA).

II. Cas d'usage « renseignement militaire »

Nous nous focaliserons dans cet article sur l'un des cas d'usage de POPCORN qui concerne la compréhension des menaces par un suivi des individus et des groupes subversifs, criminels ou terroristes et de leurs activités dans le domaine du renseignement de sécurité/défense. Pour tendre vers une maîtrise de ces menaces, POPCORN s'est concentré sur la capacité à extraire des informations à partir de bulletins de renseignements et autres sources de données textuelles. Il s'agit de se focaliser dans un premier temps sur l'analyse de bulletins en français car peu de travaux scientifiques traitent cette langue pour les tâches d'EI concernées par ce cas d'usage (cf. section III).

Dans ce cadre, il existe plusieurs systèmes (Palantir, IBM i2 Analyst's Notebook, DCGS, SIEM, etc.) qui disposent de capacités permettant à des utilisateurs de capitaliser les connaissances dans une base dédiée, mais ces fonctions sont aujourd'hui encore en grande partie quasi entièrement exécutées manuellement et demandent aux utilisateurs de s'approprier de grands ensembles documentaires afin de les structurer, ce qui est une charge lourde et coûteuse. Par la mise en œuvre de techniques automatisées d'extraction d'information, le projet POPCORN vise à simplifier le travail des utilisateurs en automatisant la transformation d'informations non structurées en informations structurées afin de peupler les bases de connaissances. Cette automatisation permettrait notamment d'augmenter la quantité d'informations en bases de connaissances et de réduire le temps nécessaire à l'intégration de ces connaissances dans lesdites bases.

Les avantages procurés par une base ainsi structurée à partir de milliers de documents consistent globalement en l'apport de connaissances aux opérationnels qui impactent leur compréhension de la situation. Par ailleurs, des alertes sur des entités critiques peuvent être levées automatiquement dans le

but d'augmenter la réactivité des opérationnels. Par exemple, en sécurité maritime, une alerte peut concerner un navire de transport devenu suspect en raison des relations entretenues par ses propriétaires.

III. État de la question

Dans le but de peupler une base de connaissances à partir d'informations extraites de textes tout venant (non structurés), il est nécessaire de considérer *a minima* les deux tâches suivantes :

- Extraction d'entités d'intérêt (EEI) : il s'agit d'extraire les éléments textuels qui ont un intérêt pour le cas d'usage considéré et de les classer, par exemple des termes désignant des exercices militaires ou des incidents diplomatiques (cf. Fig. 1) ;
- Extraction de relations d'intérêt (ER) : il s'agit de repérer et de classer les relations explicites et implicites qui existent entre deux entités d'intérêt. Par exemple les relations temporelles ou locatives des événements (cf. Fig. 1).

Les progrès récents en intelligence artificielle laissent penser que ces tâches appliquées à des cas d'usage sécurité et défense pourraient atteindre des résultats satisfaisants. Dans la suite, nous exposons les trois verrous traités dans POPCORN ainsi que nos contributions, respectivement sur la représentation des connaissances, l'accès aux données, et la performance des modèles.

A. Quelle représentation adopter ?

Les deux tâches décrites ci-avant nécessitent l'existence d'une liste de classes d'intérêt prédéfinies (au moins pour une phase d'évaluation des systèmes) pertinentes pour le métier. Ces classes liées par les relations forment l'ontologie qui permet de représenter les connaissances d'intérêt pour un cas d'usage donné. Étant donné que le consortium de POPCORN n'a pas accès aux ontologies réellement utilisées par les services de renseignement, le consortium a expérimenté deux ontologies :

- **Ontologie POPCORN.** D'une part, le consortium s'est appuyé sur l'ontologie MIM (Multilateral Information Model) (cf. Fig. 3), développée dans le cadre du Multilateral Interoperability Programme (MIP), un organe de standardisation qui comprend 24 nations, l'agence européenne de défense et l'OTAN. Ce modèle adapté du MIM est structuré autour du pentagramme du renseignement (Joint C3 Information Exchange Data Model - JC3IEDM) et vise à fournir un standard d'interopérabilité pour les applications de C2 (Command and Control) et d'ISR (Intelligence Surveillance and Recognition). De cette ontologie qui représente 1200 classes d'intérêts, nous avons décliné une ontologie plus modeste, contenant une cinquantaine de classes d'intérêts, utilisables concrètement dans le cadre de nos expériences (cf. Fig. 2). La conception de cette ontologie a été guidée par le besoin d'Airbus Defence and Space.
- **Ontologie MR4AP.** En guise d'alternative à l'ontologie POPCORN et afin d'éviter le développement de solutions ayant une dépendance trop forte avec le métier, nous avons conçu et développé une ontologie qui est en mesure de représenter la totalité de l'information véhiculée dans le texte quel que soit le métier. Elle se distingue nettement de l'ontologie POPCORN par le fait qu'elle est centrée « événement ». Cette ontologie, nommée MR4AP (*Meaning Application For Application Purposes)* permet de structurer la totalité de l'information (i.e. pas uniquement les informations d'intérêt pour un cas d'usage) d'un document et est robuste au multilinguisme [7]. MR4AP est accessible librement[1].

Dans la suite, nous présentons les expérimentations de structuration d'information en utilisant de part et d'autre les deux ontologies.

B. Quelles données utiliser ?

En raison du caractère extrêmement sensible de la donnée opérationnelle, le consortium de POPCORN n'a pas accès aux données réelles du cas d'usage. Pourtant, les données sont nécessaires pour le développement d'intelligences artificielles, au moins pour leur évaluation.

Concernant la tâche d'EEI, en particulier d'entités nommées (noms de personnes, d'organisations, de lieux, etc.), de nombreux jeux de données en français annotés ont été publiés. Certains sont commercialisés (par exemple ESTER[2]), d'autres sont inaccessibles (par exemple DAWT [25]) ou distribués à usage non commercial uniquement (par exemple WikiNeural [34]). Enfin, certains sont annotés avec les URI DBpedia mais pas directement avec les types d'entités (par exemple [36]) et sont plutôt destinés à une tâche de liage d'entités. Notons que des approches permettent de générer des jeux de données annotés en entités nommées « à la volée » en fonction de certains critères (par exemple GeNER [52]).

Nous avons finalement recensé onze jeux de données en français qui sont à la fois accessibles et libres d'utilisation (cf. Tab. I). Dix des onze jeux de données sont annotés avec un nombre de classes inférieur à quinze. Sur cet aspect, le jeu de données Wikipedia-ner [40] se distingue des autres puisqu'il contient 41 classes bien qu'il soit limité en taille (21 855 tokens). Il apparaît ainsi qu'au lancement du projet, il n'existait aucun grand jeu de données en français annoté en entités et en relations selon des ontologies représentant plusieurs dizaines de classes et de relations d'intérêts pour le renseignement militaire.

Suite à ces constats, dans le cadre du projet POPCORN, nous avons construit les jeux de données suivants :

- **DWIE-FR** : jeu de données en français développé à partir d'un jeu de données en anglais nommé DWIE [37]. La totalité du jeu de données ainsi que la méthode suivie pour son développement ont été publiés [2]. Ce jeu de

[1]https://github.com/Emvista/MR4AP/tree/main

[2]http://catalog.elra.info/product_info.php?products_id=999

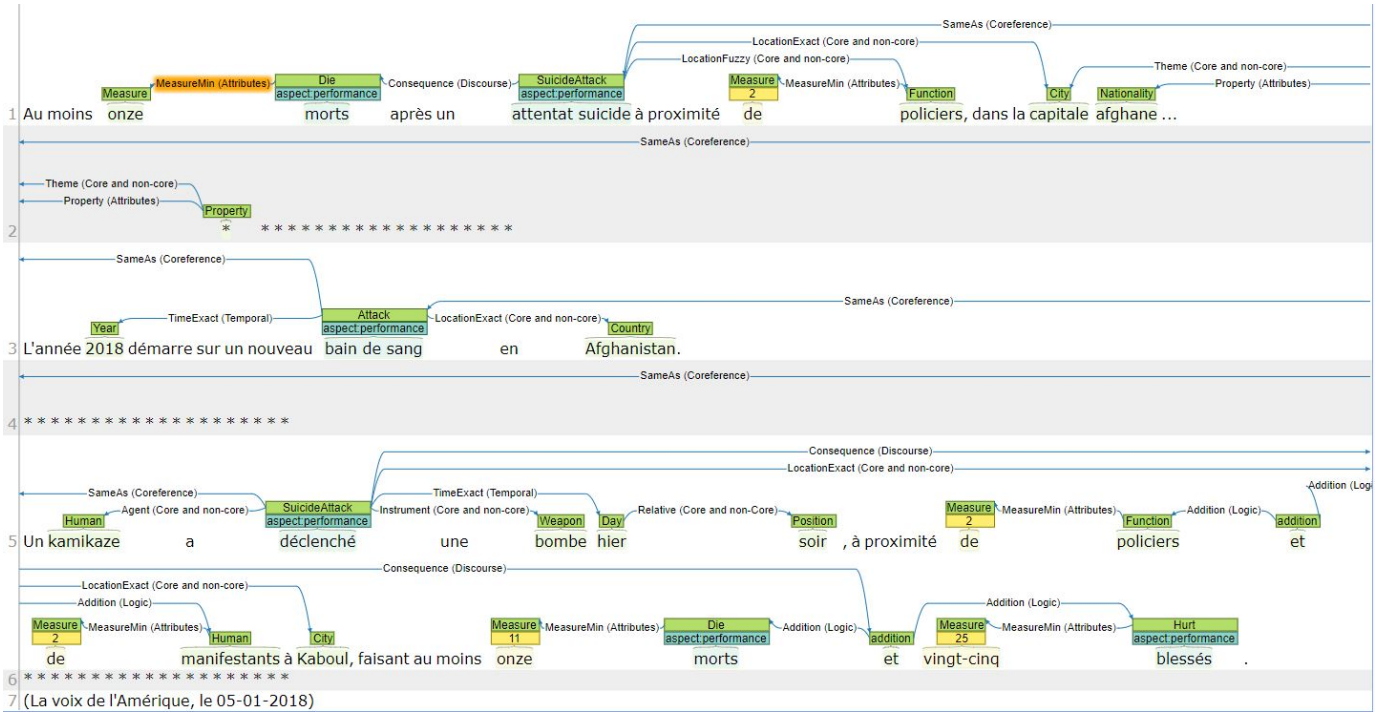


Fig. 1. Exemple d'un texte annoté manuellement avec l'ontologie MR4AP. Cette annotation comporte l'identification des entités d'intérêts (cadres verts) et des relations reliant ces entités (liens étiquetés). [7].

Entity Classes	Definition
Actor	Person or Organization
Organization	Administrative or functional structure. Remarks: An Organization is constituted to accomplish an aim, purpose, or mission
Government Organization	Organization controlled by a national or international government
Military Organization	Government Organization that is officially sanctioned and is trained and equipped to exert force
Non-Military Organization	Organization that controls and administers public policy either under a national or international mandate
Group of Individuals	Group of people gathered under a label for a specific purpose
Intergovernmental Organization	Organization conducted by two or more governments
Non-Governmental Organization	Organization that doesn't belong to the government
Person	Human being
Civilian	Person not in the armed services or the police force
Criminal	Person who violates the law or attempts to further their views by a system of coercive intimidation
Military	Person who belongs to a military force
Event	Routine, cyclical, planned, or spontaneous activities that significantly affect organizations, people, and military operations
Accident	Unfortunate event, especially one causing physical harm or damage, brought about unintentionally
CBRN Event	Event that involves chemical, biological, radiological or nuclear material individually or in combination, and are NOT attacks
Civil Unrest	Event that expresses dissatisfaction of citizens through disturbance and agitation, typically involving public demonstrations or disorder
Agitating Trouble Making	Stirring up of public interest on a matter of controversy, such as a political or social issue
Civil War Outbreak	Events related to a war among fellow citizens or within the limits of one community
Coup d'État	Violent or illegal seizure of power
Demonstration	Public meeting or march legally expressing protests, opinions or feelings towards a cause,

Fig. 2. Aperçu de l'ontologie POPCORN

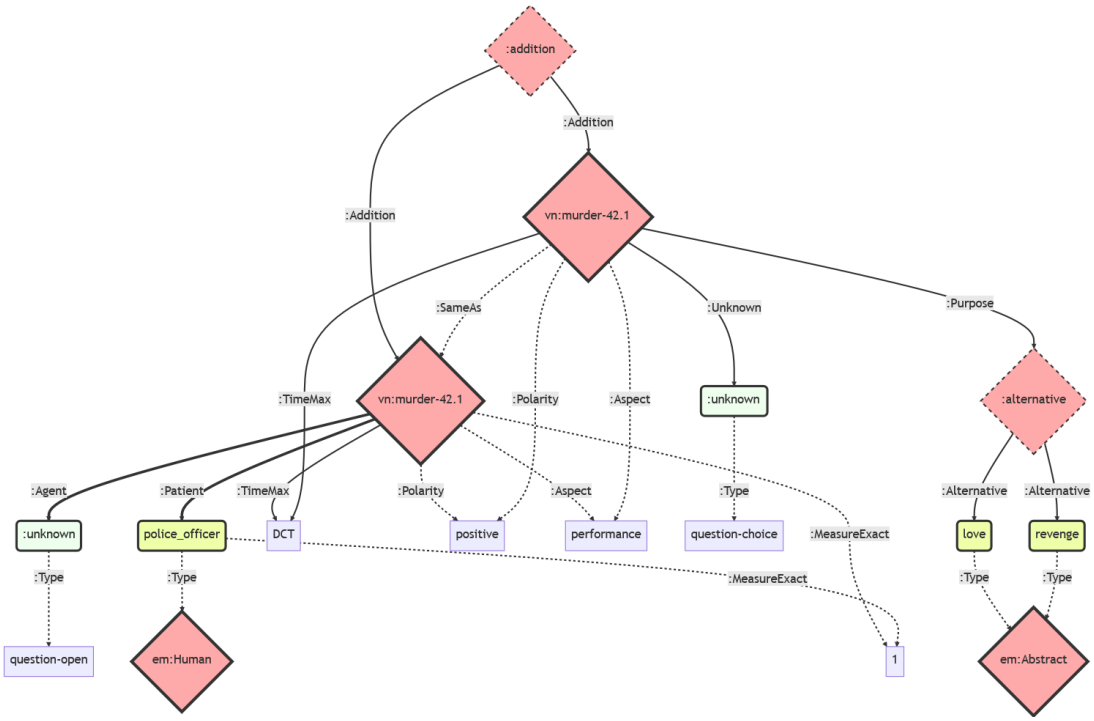


Fig. 3. Exemple de graphe MR4AP résultant de l'analyse du texte « Who committed the murder of that police officer, and was it for revenge or for love? » ; le même graphe MR4AP est obtenu pour le texte « Was this murder perpetrated out of revenge or out of love? And who killed that policeman? »

Jeu de données	Tokens	Annotations	Classes	Références
MultiNERD	4 300 000	279 300	15	[35]
WikiNeural	3 240 000	231 000	4	[34]
Le tour du monde en 80 jours	84 972	6 076	12	[40]
WiNER-Fr	322 931	24 144	7	[53]
Wikipedia-ner	21 855	6 132	41	[40]
CAP Twitter	env. 60 000	6 562	13	[38]
Europeana-newpapers-ner	207 000	13 860	3	[42]
Quaero French Medical Corpus	72 183	16 233	10	[43]
French TreeBank	350 931	11 636	7	[44]
WikiNer-Fr	3 499 695	420 061	4	[45]
UkraiNER	env. 400 000	43 623	8	[24]

TABLE I
JEUX DE DONNÉES EN FRANÇAIS, ANNOTÉS EN ENTITÉS ET ACCESSIBLES

données contient 589 394 tokens dont 60 292 annotés en entités d'intérêt selon 169 classes ;

- **POPCORN-data-manual** : rédigé et annoté en partie par un prestataire[3], ce jeu de données contient 2 000 textes simulant des bulletins d'information. Ce jeu de données a été annoté avec l'ontologie POPCORN. Une partie du jeu de données a été publié [26] ;
- **POPCORN-data-auto** : jeu de données annoté selon l'ontologie POPCORN en utilisant des IA génératives. Une partie du jeu de données a été publié [26] ;
- **Renseignor** : bulletins d'informations issus de Renseignor[4]. Ces données sont en cours d'annotation d'une part avec l'ontologie POPCORN et d'autre part avec l'ontologie MR4AP.

Concernant la tâche d'ER, il n'existe à notre connaissance aucun jeu de données en français accessible librement qui soit annoté dans un contexte sécurité défense. Parmi les jeux de données annotés en entités que nous avons produits, POPCORN-data-manual a également été annoté en relations, entièrement manuellement, et Renseignor est en cours d'annotation. Quelques centaines de textes déjà annotés permettent de publier les premiers résultats dans cet article (cf. section III-C).

À la fin du projet, POPCORN aura produit quatre jeux de données en français annotés avec plusieurs dizaines de classes d'entités d'intérêts pour le renseignement militaire, dont deux jeux de données annotés en relations d'intérêts. Notons que, bien que le jeu de données UkraiNER indiqué

[3]https://fr.isahit.com/
[4]https://cf2r.org/publications/lettre/renseignor/

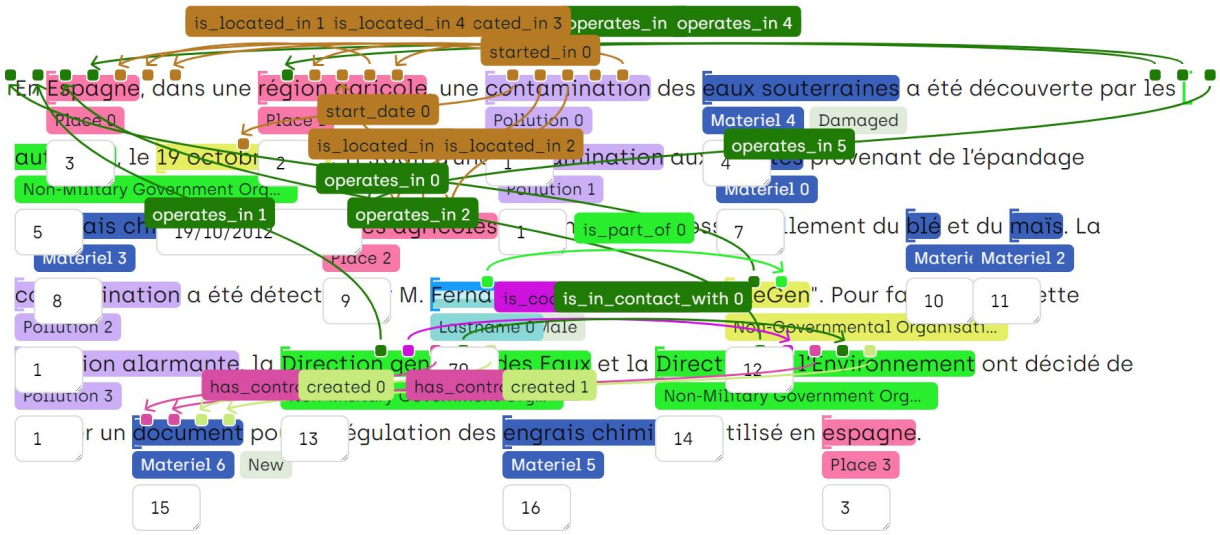


Fig. 4. Exemple d'un texte annoté manuellement issu de POPCORN-data-manual (texte 5_1470)

en section III soit annoté avec seulement 8 classes génériques (non spécifique à la sécurité / défense, la thématique du corpus en fait le quatrième jeu de données annoté connu jusqu'à maintenant (cf. II). À ce jour, les seuls jeux de données « sécurité/défense » accessibles qui sont annotés à la fois en entités et en relations sont les jeux de données POPCORN-data-manual et POPCORN-data-auto.

C. Quels modèles d'IA pour structurer et extraire l'information ?

Le troisième verrou qui doit être résolu dans POPCORN concerne les performances limitées des modèles supervisés pour l'extraction d'entités et de relations d'intérêt, en particulier lorsque l'on traite plusieurs dizaines de types d'entités et de relations. Le consortium de POPCORN a donc entreprit la reproduction et l'expérimentation de modèles à l'état de l'art, dans le but de les comprendre et de comparer leurs performances sur les jeux de données issus du consortium.

Les premières expérimentations ont porté sur des modèles d'EEI « classiques » reposant sur des modèles de langage francophones largement utilisés pour leur capacité à vectoriser l'information dans les documents, tels que CamemBERT [28] et FlauBERT [27], augmentés d'une couche linéaire de classification pour l'étiquetage de mots.

Les préoccupations récentes en extraction d'entités, telles que l'attention portée à l'extraction de fragments plutôt qu'à l'étiquetage de tokens, la détection d'entités imbriquées et la prédiction de multiples labels pour une même mention, nous ont conduit à explorer des modèles plus récents, notamment DeepSpanRepresentations [29], Biaffine-NER [30], Boundary Smoothing (BS) [8] et PromptNER [31], des modèles représentant efficacement les fragments textuels par des opérations bi-affine, des Bi-LSTM ou une meilleure gestion de l'ambiguïté liée aux frontières des mentions. Par ailleurs des modèles d'extraction de relations à l'état de l'art tels que DREEAM [48], ATLOP [51], PEMSCL [49] et KD-DocRE [50] ont aussi été expérimentés. En parallèle, nous avons également développé notre propre modèle résolvant les différentes modalités, le modèle unifié [23].

Le développement du jeu de données POPCORN-data-auto (cf. section III-B) a été rendu possible grâce à l'utilisation d'un grand modèle de langage génératif (LLM), Vigostral-7B[5], un modèle conversationnel en français librement accessible et affiné à partir de Mistral-7B [9].

Le développement du jeu de données POPCORN-data-auto (cf. section III-B) a nécessité l'expérimentation de processus sophistiqués impliquant des grands modèles de langage génératif (LLM) tels que Vigostral-7B[6], un modèle conversationnel en français affiné à partir de Mistral-7B [9].

Les expériences menées avec les modèles mentionnés, leurs résultats et les analyses qui en découlent sont présentés dans nos contributions [23] et [26]. Nous reportons ici quelques résultats obtenus sur l'ontologie POPCORN avec des modèles entraînés sur peu de données en comparaison avec les données dont dispose le consortium : 400 textes issus de POPCORN-data-manuel enrichis de 400 textes de POPCORN-data-auto (cf. Tableau III). Notons que ce rapport de 1 pour 1 (400 textes rédigés et annotés manuellement pour 400 produits de façon automatique) a apporté les meilleurs résultats à ce stade. Ces 800 textes sont librement accessibles afin d'assurer la reproductibilité de nos expériences par des tiers.

D. Quelles performances ?

1) Avec l'ontologie POPCORN: Les modèles traitant de façon conjointe les deux tâches (noté « Modèle joint » dans le Tableau III) sont légèrement plus performants, ce qui a par

[5]https://huggingface.co/bofenghuang/vigostral-7b-chat
[6]https://huggingface.co/bofenghuang/vigostral-7b-chat

Jeu de données	Tokens	Entités annotées	Relations annotées	Classes d'entités	Classes de relations	Références
DWIE-FR	589 394	60 292	0	169	0	[2]
POPCORN	114 469	43 264	37 065	55	36	[26]
Renseignor	en cours	en cours	en cours	70	39	à paraître

TABLE II
JEUX DE DONNÉES ACCESSIBLES OU PROCHAINEMENT ACCESSIBLES, EN FRANÇAIS, ANNOTÉS AVEC DES CLASSES MÉTIERS « SÉCURITÉ/DÉFENSE ».

ailleurs encouragé l'annotation des jeux de données à la fois en entités et en relations. Il apparaît que les meilleurs résultats sont de l'ordre de 81% de F1 Micro pour la reconnaissance d'entités d'intérêt et de 59.01% pour la reconnaissance des événements. On remarque une difficulté à extraire les entités de type « événements » bien que la tâche semble très proche de la reconnaissance d'entités.

Dans le cadre de cette évaluation, les modèles d'extraction de relations ont pris en entrée le texte déjà annoté (manuellement) en entités d'intérêts ; les résultats présentés ici correspondent donc uniquement à la tâche de reconnaissance de relations et d'étiquetage. L'extraction de relations atteint un F1 Micro de 59.38% avec le modèle ATLOP contraint par les restrictions sémantiques sur les domaines (i.e. tête) et co-domaines (i.e. queue des relations).

Les évaluations menées dans le cadre de POPCORN ont généralement montré une faiblesse de ces modèles concernant la détection des frontières gauche et droite des entités d'intérêt. Une augmentation jusqu'à 10% sur le score F1 est observée lorsque l'on fait l'hypothèse d'une identification parfaite des frontières[7]. Dans ce sens, des expériences sont en cours : dans un premier temps, nous avons choisi de remplacer le module Bi-LSTM de l'architecture Biaffine-ner par un GCN (Graph Convolutional Networks). Nous aimerions savoir si l'ajout d'une représentation « graphe » est intéressante pour la détection de frontières.

2) Avec l'ontologie MR4AP: La tâche de structuration de l'information selon des formalismes de représentation sémantique tels que AMR [32] ou MR4AP [7] est complexe. De nombreuses recherches sont effectuées à ce sujet. Le premier modèle accessible permettant de transformer un texte en français en un graphe régi par un tel formalisme a été publié par l'un d'entre nous en 2023 [33]. Les résultats obtenus avec les méthodes à l'état de l'art (LLM, apprentissage par transfert, ou encore méta-apprentissage) ne permettent pas pour l'instant de dépasser des méthodes plus classiques fondées sur l'idée d'un système hybride (i.e. utilisant à la fois des techniques d'apprentissage et de règles linguistiques). La raison est certainement liée au manque de données annotées. Cela est illustré en tableau IV dans lequel le système hybride RLA (*Recursive Linguistic Analyzer*) dont les prémisses ont été publiées en 2019 [41], obtient les meilleurs résultats notamment en termes de précision probablement grâce aux règles linguistiques qui guident les modèles sous-jacents. À noter qu'aucune adaptation n'a été apportée au RLA dans le cadre de cette évaluation. L'apport de connaissances issues du jeu d'apprentissage (par exemple, des listes de noms d'organisations) améliorerait sans aucun doute les résultats. Il faut noter que les résultats mettent en évidence que la structuration selon MR4AP est plus complexe qu'avec l'ontologie métier POPCORN : sur des tâches identiques, les modèles communs appliqués aux deux ontologies obtiennent des résultats inférieurs au RLA d'au moins 13 points sur les scores F1 Macro et F1 Micro.

Dans le cadre de cette évaluation, comme dans le cadre des expériences avec l'ontologie POPCORN, les modèles d'extraction d'information ont pris en entrée le texte déjà annoté (manuellement) en entités d'intérêt ; au contraire le système RLA a directement traité le texte brut en entrée ce qui complique la tâche. Néanmoins, la précision haute (88.09%) permet d'envisager l'intégration de ce système qui a pour vocation de structurer 100% des informations véhiculées dans le texte dans une application où moins de 12% des résultats devraient être corrigés par un utilisateur. En revanche, le rappel est très faible (environ 25%) ce qui pourrait être rattrapé par la prise en compte des résultats d'autres modèles.

IV. INTÉGRATION DES RÉSULTATS

Dans cette section, nous présentons succinctement deux démonstrateurs, l'un utilisant les modèles d'EEI et d'ER selon l'ontologie POPCORN et l'autre selon l'ontologie MR4AP. Cette dernière n'étant pas une ontologie métier, nous donnons l'intuition du processus qui permet à un utilisateur de l'adapter rapidement à son cas d'usage.

A. Avec l'ontologie POPCORN

L'adaptation d'outils d'extraction d'information aux métiers du renseignement a montré l'importance de la sélection des connaissances capitalisées en base et du maintien de leur cohérence. Les résultats exposés précédemment montrent que l'amélioration de la qualité de l'extraction d'information à l'état de l'art ne suffit pas encore à garantir cette cohérence, notamment à cause de l'accumulation des erreurs en cas de capitalisation automatique sans vérification ni modification humaine.

Dans le cadre de POPCORN, un démonstrateur fonctionnel permet d'évaluer la difficulté du rôle de l'humain dans la chaîne globale de peuplement de bases de connaissances. Il est illustré en figure 5. Ici, l'humain est le validateur final des éléments extraits par l'IA. La réalisation du démonstrateur a requis de lister les modifications possibles par l'humain et de décider de la manière la plus pertinente de les lui proposer : au niveau des entités ou de leurs mentions, dans le texte ou dans le graphe. Certaines modifications n'ont

[7]Cette observation est d'importance lorsque l'on considère un module de liage des mentions du texte avec les entités de la base qui se situent en aval des modèles étudiés ici.

Tâche	Modèle	Précision	Rappel	F1 Macro	F1 Micro
Extraction des événements	Modèle joint	**52.87**	43.21	**44.79**	**59.38**
	Biaffine-NER + Boundary Smoothing	41.13	**49.16**	43.92	55.94
	Camembert-base(p.e. Wikikner) + BiLSTM + CRF	46.84	44.76	44.02	57.80
EEI sauf événements	Modèle joint + contraintes domaines	**76.23**	**67.01**	**69.23**	**82.31**
	Biaffine-NER + Boundary Smoothing	62.65	66.14	65.82	81.14
	Camembert-base(p.e. Wikikner) + BiLSTM + CRF	72.18	65.48	66.67	81.75
ER	Modèle joint + contraintes domaines	54.73	49.88	47.78	58.02
	Dreeam + contraintes domaines	**72.11**	47.65	53.48	59.24
	ATLOP + contraintes domaines	68.35	**50.27**	**54.33**	**59.38**

TABLE III
RÉSULTATS OBTENUS PAR LES MODÈLES EXPÉRIMENTÉS SUR L'ONTOLOGIE POPCORN ET UN JEU DE DONNÉES CONSTITUÉ DE 400 TEXTES ISSUS DE POPCORN-DATA-MANUEL ENRICHI DE 400 TEXTES SYNTHÉTIQUES ISSUS DE POPCORN-DATA-AUTO

Tâche	Modèle/Système	Précision	Rappel	F1 Macro	F1 Micro
Extraction des événements	Biaffine-NER + BS (p.e. WikiNer-FR)	11.18	10.49	9.90	34.45
	Camembert-base (p.e. WikiNer) + BiLSTM + CRF	13.79	12.35	12.07	41.80
	RLA	**53.98**	**55.41**	**57.05**	**59.69**
EEI sauf événements	Biaffine-NER + BS (p.e. WikiNer-FR)	20.87	21.30	19.95	**58.78**
	Camembert-base (p.e. WikiNer) + BiLSTM + CRF	28.74	25.97	25.19	**62.38**
	RLA	**69.99**	**58.11**	**55.17**	55.24
ER	ATLOP	19.41	14.25	15.84	**66.82**
	RLA	**88.09**	**25.75**	**34.90**	39.85

TABLE IV
RÉSULTATS OBTENUS PAR LES MODÈLES ET LE SYSTÈME EXPÉRIMENTÉS SUR LES DONNÉES RENSEIGNOR, WIKIPEDIA, WIKINEWS ANNOTÉES AVEC L'ONTOLOGIE MR4AP (EN COURS D'ANNOTATION ; EXPÉRIENCES MENÉES AVEC 19K *tokens* ANNOTÉS DONT 1835 MENTIONS D'ÉVÉNEMENTS ; P = PRÉCISION ; R = RAPPEL. L'ABBRÉVIATION "P.E." SIGNIFIE "PRÉ-ENTRAÎNÉ SUR".

pas nécessairement à être accessibles depuis chaque vue. Ainsi, l'utilisateur est capable de modifier toute information détectée (relations, attributs, mentions textuelles de ces entités et leurs coréférences, entités de référence en base), et d'ajouter de nouveaux éléments non détectés. Côté graphe, l'interface utilise les couleurs et icônes ainsi que les épaisseurs de traits pour expliciter les natures et provenances des informations. Une fois que les résultats de l'IA ont été revus par l'humain, la base de connaissances est mise à jour et les prochaines analyses bénéficieront de l'ensemble du contenu de la base.

B. Avec l'ontologie MR4AP

L'utilisation de l'ontologie MR4AP a pour intérêt de ne pas restreindre l'apprentissage de l'IA à un seul cas d'usage. Techniquement, MR4AP est une ontologie qui contient des centaines de concepts et seulement 44 relations qui permettent de représenter la totalité de l'information véhiculée dans un texte. Nous avons développé une interface qui permet à un utilisateur d'aligner les concepts et relations génériques de MR4AP avec les siens. Ainsi les informations visualisées sur l'interface adoptent le vocabulaire métier nécessaire aux opérationnels.

La figure 6 montre un bulletin d'information à partir duquel un événement MR4AP de type « Murder » a été détecté. Des informations relatives à cet événement ont été détectées : la date (normalisée afin d'être intégrée en base de données), l'agent (celui qui fait l'action), le patient (celui qui subit l'action), le lieu. Ces informations sont liées à une base de données (DBpedia pour ce démonstrateur) lorsque cela est rendu possible. Dans l'exemple, la fiche contenant des informations sur le lieu de l'événement peut être visualisée. Toutes les informations peuvent être éditées avant leur intégration dans la base de connaissances.

V. CONCLUSION

Le projet POPCORN a contribué aux méthodes scientifiques de structuration d'information pour des cas d'usage sécurité/défense. De nombreuses expériences ont été effectuées pour extraire les entités d'intérêts et les relations qu'elles entretiennent entre elles. Les résultats obtenus permettent d'envisager l'insertion de certains types d'entités et relations en bases de données avec leurs indices de confiance et les sources desquelles elles proviennent afin d'être vérifiées par un analyste qui en aurait l'utilité. Pour un cas d'usage où il s'agirait de peupler les bases de données avec les informations extraites, il est aujourd'hui encore nécessaire que l'humain valide certaines informations extraites automatiquement par l'IA, notamment pour les types d'informations dont les résultats sont trop faibles.

Le système global répondant au cas d'usage est en cours de développement : à partir d'une évaluation plus fine que celle présentée ici, nous avons pu déterminer quel type d'information est le mieux inféré pour chaque modèle. Même si de façon globale les types obtiennent des résultats similaires, certains modèles sont plus robustes aux faibles quantités d'exemples présents dans leur jeu d'entraînement.

L'étude des données synthétiques est prometteuse pour la suite de nos recherches. Ces données résultent en une augmentation significative des performances qualitatives des modèles. La qualité des données générées automatiquement

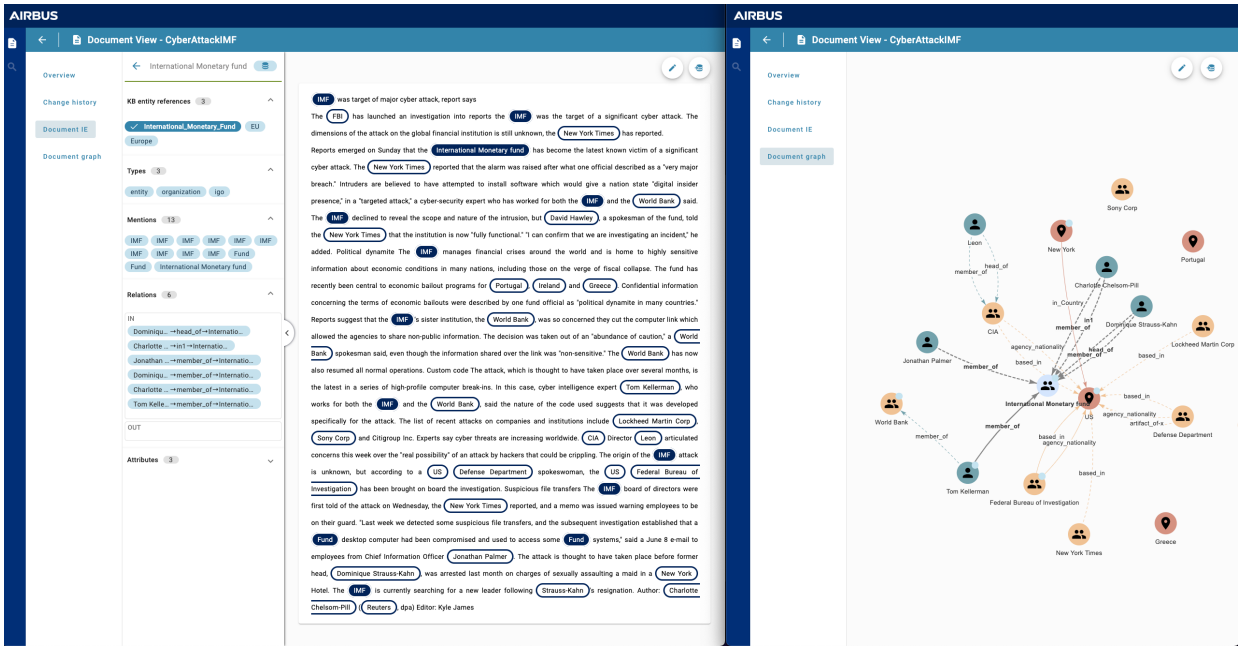


Fig. 5. Aperçu du démonstrateur POPCORN : vues synchronisées du texte et des connaissances extraites.

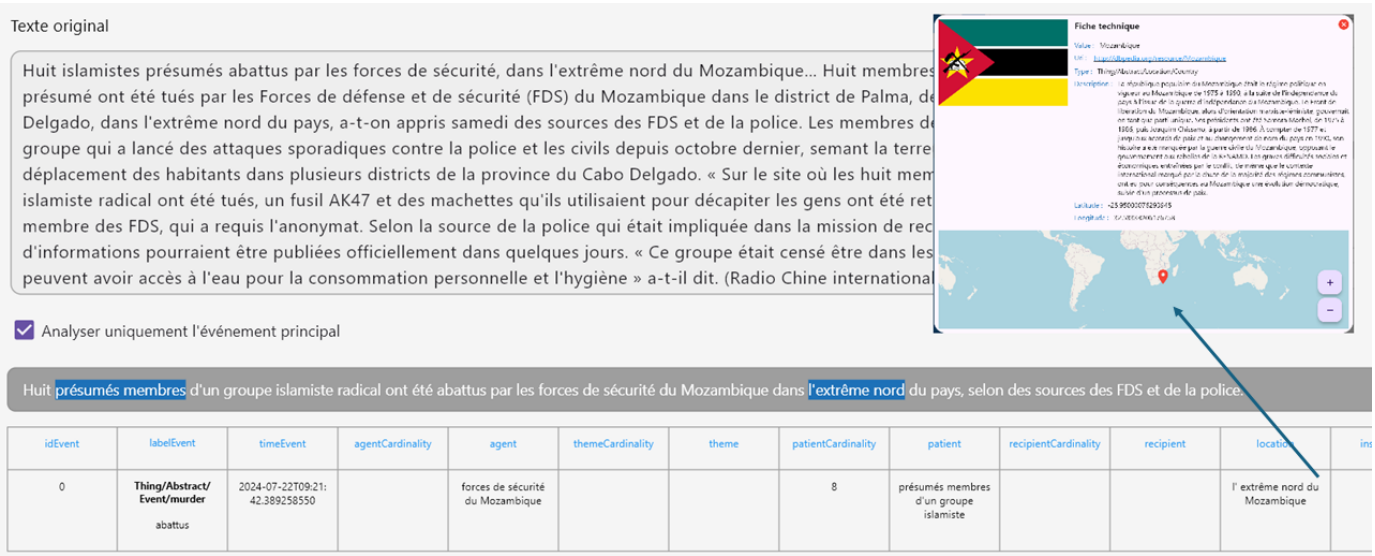


Fig. 6. Exemple d'un texte annoté automatiquement issu de Renseignor. Image avant l'étape de configuration "métier". Le tableau contient une ligne générée automatiquement qui représente un événement "Murder", sa date, les personnes impliquées, son lieu. La fiche technique contient des informations de la base de connaissance associées au lieu identifié.

fait actuellement l'objet de recherches et des données de meilleure qualité à venir dans les prochains mois permettront de produire de nouvelles versions des modèles.

L'apparition des LLM conversationnels au cours du projet POPCORN a permis de les considérer sérieusement comme une alternative aux modèles déterministes avec l'avantage qu'ils ne nécessitent pas (ou peu) de données annotées. Néanmoins, les LLM à eux seuls s'avèrent peu performants pour structurer l'information selon des ontologies complexes (cf. EvalLLM[8]). Une voie de prolongement de ces travaux consisterait à tester des modèles de taille bien supérieure (d'un facteur 10 ou 100) et/ou des modèles plus centrés sur le français afin de déterminer si nos méthodes de création de données synthétiques résultent en un gain qualitatif dans ces contextes.

[8]Atelier sur l'évaluation des modèles génératifs (LLM) et challenge d'extraction d'information few-shot : https://evalllm2024.sciencesconf.org/program?lang=fr

REFERENCES

[1] Zaporojets, Klim, Deleu, Johannes, Develder, Chris and Demeester, Thomas. (2021) "DWIE: An entity-centric dataset for multi-task document-level information extraction". *Information Processing and Management*. **58**(4): 102563

[2] Verdy, S., Prieur, M., Gadek, G. & Lopez, C. DWIE-FR : Un nouveau jeu de données en français annoté en entités nommées. *Actes De CORIA-TALN 2023. Actes De La 30e Conférence Sur Le Traitement Automatique Des Langues Naturelles, TALN 2023 - Volume 2 : Travaux De Recherche Originaux - Articles Courts, Paris, France, June 5-9, 2023*. pp. 63-72 (2023)

[3] Sang, E. and Meulder, F. (2003) "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition". *Proceedings Of The Seventh Conference On Natural Language Learning, CoNLL 2003, Held In Cooperation With HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*. pp. 142-147.

[4] Serrano, L., Bouzid, M., Charnois, T., Brunessaux, S. & Grilheres, B. Extraction et agrégation automatique d'événements pour la veille en sources ouvertes: du texte à la connaissance. (2013) *IC-24èmes Journées Francophones D'Ingénierie Des Connaissances*.

[5] Prieur, M., Mouza, C., Gadek, G. & Grilhères, B. Evaluating and Improving End-to-End Systems for Knowledge Base Population. *Proceedings Of The 15th International Conference On Agents And Artificial Intelligence, ICAART 2023, Volume 3, Lisbon, Portugal, February 22-24, 2023*. pp. 641-649 (2023)

[6] Yao, Y., Ye, D., Li, P., Han, X., Lin, Y., Liu, Z., Liu, Z., Huang, L., Zhou, J. & Sun, M. DocRED: A Large-Scale Document-Level Relation Extraction Dataset. *Proceedings Of The 57th Conference Of The Association For Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. pp. 764-777 (2019)

[7] Giordano, B. & Lopez, C. (2023). MR4AP: Meaning representation for application purposes. In Proceedings of the Fourth International Workshop on Designing Meaning Representations, pp. 110-121.

[8] Zhu, Li (2022) "Boundary Smoothing for Named Entity Recognition" *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*

[9] Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., Casas, D.D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L.R., Lachaux, M., Stock, P., Scao, T.L., Lavril, T., Wang, T., Lacroix, T., & Sayed, W.E. (2023). Mistral 7B. ArXiv, abs/2310.06825.

[10] Sosuke Kobayashi. 2018. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pp. 452–457

[11] Jason Wei and Kai Zou. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 6382–6388

[12] Zhang, Danqing & Li, Tao & Zhang, Haiyang & Yin, Bing. (2020). On Data Augmentation for Extreme Multi-label Classification.

[13] George A. Miller. 1994. WordNet: A Lexical Database for English. In Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994.

[14] Sérasset, Gilles. 'DBnary: Wiktionary as a Lemon-based Multilingual Lexical Resource in RDF'. 1 Jan. 2015 : 355 – 361.

[15] Mikolov, Tomas & Chen, Kai & Corrado, G.s & Dean, Jeffrey. (2013). Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR. 2013.

[16] Jeffrey Pennington, Richard Socher, and Christopher Manning. (2014) GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543

[17] Xie, Z., Wang, S. I., Li, J., Lévy, D., Nie, A., Jurafsky, D., & Ng, A. Y. (2019). Data noising as smoothing in neural network language models. Paper presented at 5th International Conference on Learning Representations, ICLR 2017, Toulon, France.

[18] Alexander Fabbri, Simeng Han, Haoyuan Li, Haoran Li, Marjan Ghazvininejad, Shafiq Joty, Dragomir Radev, and Yashar Mehdad. 2021. Improving Zero and Few-Shot Abstractive Summarization with Intermediate Fine-tuning and Data Augmentation. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 704–717

[19] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. & Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances In Neural Information Processing Systems 35: Annual Conference On Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*. (2022)

[20] Roland Roller, Eneko Agirre, Aitor Soroa, and Mark Stevenson. 2015. Improving distant supervision using inference learning. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp. 273–278, Beijing, China. ACL.

[21] Xiang Deng and Huan Sun. 2019. Leveraging 2-hop Distant Supervision from Table Entity Pairs for Relation Extraction. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 410–420, Hong Kong, China. Association for Computational Linguistics.

[22] Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E.H., & Zhou, D. (2022). Self-Consistency Improves Chain of Thought Reasoning in Language Models. ArXiv, abs/2203.11171.

[23] Prieur, M., Mouza, C., Gadek, G. & Grilhères, B. Shadowfax: Harnessing Textual Knowledge Base Population. *Proceedings Of The 47th International ACM SIGIR Conference On Research And Development In Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*. pp. 2796-2800 (2024), https://doi.org/10.1145/3626772.3657666

[24] Lauriane Aufrant, Lucie Chasseur (2024). UkraiNER: A New Corpus and Annotation Scheme towards Comprehensive Entity Recognition. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024) pp. 16941-16952.

[25] Spasojevic, N., Bhargava, P., & Hu, G. (2017). Dawt: Densely annotated wikipedia texts across multiple languages. In Proceedings of the 26th International Conference on World Wide Web Companion, pp. 1655-1662.

[26] Bastien Giordano, Maxime Prieur, Nakanyseth Vuth, Sylvain Verdy, Kévin Cousot, Gilles Sérasset, Guillaume Gadek, Didier Schwab, Cédric Lopez (2024) POPCORN: Fictional and Synthetic Intelligence Reports for Named Entity Recognition and Relation Extraction Tasks. In Proceedings of KES, Sevilla, Spain, to appear.

[27] Le, H., Vial, L., Frej, J., Segonne, V., Coavoux, M., Lecouteux, B., ... & Schwab, D. (2020, June). FlauBERT: des modèles de langue contextualisés pré-entraînés pour le français. In 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 2: Traitement Automatique des Langues Naturelles (pp. 268-278). ATALA; AFCP.

[28] Martin, L., Muller, B., Suárez, P., Dupont, Y., Romary, L., Clergerie, Seddah, D. & Sagot, B. CamemBERT: a Tasty French Language Model. *Proceedings Of The 58th Annual Meeting Of The Association For Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. pp. 7203-7219 (2020), https://doi.org/10.18653/v1/2020.acl-main.645

[29] Zhu, E., Liu, Y., & Li, J. (2022). Deep span representations for named entity recognition. arXiv preprint arXiv:2210.04182.

[30] Yu, J., Bohnet, B. & Poesio, M. Named Entity Recognition as Dependency Parsing. *Proceedings Of The 58th Annual Meeting Of The Association For Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. pp. 6470-6476 (2020), https://doi.org/10.18653/v1/2020.acl-main.577

[31] Shen, Y., Tan, Z., Wu, S., Zhang, W., Zhang, R., Xi, Y., Lu, W. & Zhuang, Y. PromptNER: Prompt Locating and Typing for Named Entity Recognition. *Proceedings Of The 61st Annual Meeting Of The Association For Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*. pp. 12492-12507 (2023), https://doi.org/10.18653/v1/2023.acl-long.698

[32] Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., ... & Schneider, N. (2013, August). Abstract meaning representation for sembanking. In Proceedings of the 7th linguistic annotation workshop and interoperability with discourse (pp. 178-186).

[33] Kang, J., Coavoux, M., Lopez, C., & Schwab, D. (2023). Analyse sémantique AMR pour le français par transfert translingue. In 18e

Conférence en Recherche d'Information et Applications–16e Rencontres Jeunes Chercheurs en RI–30e Conférence sur le Traitement Automatique des Langues Naturelles–25e Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (pp. 55-62). ATALA.

[34] Tedeschi, S., Maiorca, V., Campolungo, N., Cecconi, F., & Navigli, R. (2021, November). WikiNEuRal: Combined neural and knowledge-based silver data creation for multilingual NER. In Findings of the association for computational linguistics: EMNLP 2021 (pp. 2521-2533).

[35] Tedeschi, S., & Navigli, R. (2022, July). MultiNERD: A multilingual, multi-genre and fine-grained dataset for named entity recognition (and disambiguation). In Findings of the Association for Computational Linguistics: NAACL 2022 (pp. 801-812).

[36] Hellmann, S., Lehmann, J., Auer, S., & Brümmer, M. (2013). Integrating NLP using linked data. In The Semantic Web–ISWC 2013: 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II 12 (pp. 98-113). Springer Berlin Heidelberg.

[37] Zaporojets, K., Deleu, J., Develder, C. & Demeester, T. DWIE: An entity-centric dataset for multi-task document-level information extraction. *Information Processing & Management*. **58**, 102563 (2021)

[38] Lopez, C., Partalas, I., Balikas, G., Derbas, N., Martin, A., Reutenauer, C., ... & Amini, M. R. (2017). Cap 2017 challenge: Twitter named entity recognition. arXiv preprint arXiv:1707.07568.

[39] Dupont, Y. (2019). Un corpus libre, évolutif et versionné en entités nommées du français. In TALN 2019-Traitement Automatique des Langues Naturelles.

[40] Lopez, C., Mekaoui, M., Aubry, K., Bort, J., & Garnier, P. (2019, January). Reconnaissance d'entités nommées itérative sur une structure en dépendances syntaxiques avec l'ontologie nerd. In Extraction et Gestion des Connaissances: Actes de la conférence EGC (Vol. 79, pp. 81-92).

[41] Lopez, C., Mekaoui, M., Aubry, K., Bort, J., & Garnier, P. (2019, January). Reconnaissance d'entités nommées itérative sur une structure en dépendances syntaxiques avec l'ontologie nerd. In Extraction et Gestion des Connaissances: Actes de la conférence EGC (Vol. 79, pp. 81-92).

[42] Neudecker, C. (2016, May). An open corpus for named entity recognition in historic newspapers. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16) (pp. 4348-4352).

[43] Grouin, A. N. C., Leixa, J., Rosset, S., & Zweigenbaum, P. (2014). The Quaero French Medical Corpus: A Ressource for Medical Entity Recognition and Normalization.

[44] Sagot, B., Richard, M., & Stern, R. (2012). Annotation référentielle du Corpus Arboré de Paris 7 en entités nommées. In Traitement Automatique des Langues Naturelles (TALN) (Vol. 2).

[45] Nothman, J., Curran, J. R., & Murphy, T. (2008, December). Transforming Wikipedia into named entity training data. In Proceedings of the Australasian Language Technology Association Workshop 2008 (pp. 124-132).

[46] Serrano, L. Vers une capitalisation des connaissances orientée utilisateur: extraction et structuration automatiques de l'information issue de sources ouvertes. (Universté de Caen,2014)

[47] Gerz, M., Mulikita, M., Bau, N. & Gökgöz, F. The MIP information model-a semantic reference for command & control. *2015 International Conference On Military Communications And Information Systems (ICMCIS)*. pp. 1-11 (2015)

[48] Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., & Wu, X. (2024). Unifying large language models and knowledge graphs: A roadmap. IEEE Transactions on Knowledge and Data Engineering.

[49] Guo, J., Kok, S., & Bing, L. (2023). Towards integration of discriminability and robustness for document-level relation extraction. arXiv preprint arXiv:2304.00824.

[50] Tan, Q., He, R., Bing, L., & Ng, H. T. (2022). Document-level relation extraction with adaptive focal loss and knowledge distillation. arXiv preprint arXiv:2203.10900.

[51] Zhou, W., Huang, K., Ma, T., & Huang, J. (2021). Document-level relation extraction with adaptive thresholding and localized context pooling. In Proceedings of the AAAI conference on artificial intelligence (Vol. 35, No. 16, pp. 14612-14620).

[52] Kim, H., Yoo, J., Yoon, S., Lee, J., & Kang, J. (2021). Simple questions generate named entity recognition datasets. arXiv preprint arXiv:2112.08808.

[53] Dupont, Y. (2019). Un corpus libre, évolutif et versionné en entités nommées du français. In TALN 2019 - Traitement Automatique des Langues Naturelles.

Pourquoi se limiter à une recherche quand on peut l'étendre ? Amélioration de l'architecture RAG par des stratégies d'expansion de requêtes et d'agrégation de documents

Louis Jourdain
ChapsVision
Paris, France
ljourdain@chapsvision.com

Skander Hellal
ChapsVision
Paris, France
shellal@chapsvision.com

Tony Marini
ChapsVision
Paris, France
tmarini@chapsvision.com

***Abstract*—Le *Retrieval Augmented Generation* (RAG) améliore les réponses générées par des grands Modèles de Langage (LLMs) en récupérant des informations externes. Cependant, la performance de ces systèmes dépend fortement de la qualité des documents récupérés. Dans cet article, nous proposons d'employer plusieurs stratégies d'expansion de requêtes, telles que la reformulation, la décomposition de requêtes et la génération de documents hypothétiques (HyDE) afin d'améliorer cette phase de récupération. Nous intégrons ces techniques dans une architecture RAG optimisée, qui comprend un routeur sémantique pour sélectionner la meilleure stratégie et un module d'agrégation pour fusionner les résultats des requêtes étendues. Nos expériences sur un corpus de 80 000 documents montrent une amélioration notable du rappel et de la qualité des réponses pour les questions complexes, tout en limitant les hallucinations. Cette nouvelle architecture présente un potentiel prometteur pour améliorer les systèmes RAG dans des domaines spécialisés.**

Mots-clés : Retrieval-Augmented Generation, expansion de requêtes, grands modèles de langage, agrégation de documents, recherche d'information.

I. Introduction

Si les grands Modèles de Langage (ou LLM pour *Large Language Models*) sont capables de générer automatiquement du texte d'une qualité qui tient la comparaison avec les productions humaines [1], le fait que leurs connaissances paramétriques soient limitées aux données suffisamment saillantes dans leur corpus d'entraînement les restreint dans de nombreux cas d'usage. Il n'est, par exemple, pas viable d'utiliser directement un LLM comme un système de *Question Answering* dans un domaine spécifique. Pour pallier cette limitation, une nouvelle architecture, nommée RAG (*Retrieval Augmented Generation*), a été proposée [2]. Elle consiste à combiner une étape de recherche documentaire (*retrieve*) dans un large corpus constitué en amont, afin de sélectionner les documents les plus pertinents par rapport à la requête de l'utilisateur (*query*) et d'inclure ces documents dans l'instruction (*prompt*) fournie au LLM, dans le but de lui fournir des connaissances non paramétriques utiles à la réalisation de la tâche de génération.

Bien que l'architecture RAG apparaisse comme une alternative prometteuse et frugale par rapport à d'autres méthodes, telles que le *fine-tuning*, pour adapter les LLMs à des domaines de spécialité [3], son succès dépend de la qualité de la recherche documentaire effectuée. En effet, l'ajout de contextes non pertinents dans l'instruction augmente non seulement le temps et le coût de réponse, mais dégrade également la qualité [4].

Améliorer la phase de récupération des documents est donc crucial pour optimiser les performances d'un système de *Question Answering* basé sur une architecture RAG, ce qui nous amène à aborder des problématiques d'optimisation des moteurs de recherche documentaire. La plupart des *retrievers* utilisés dans les architectures RAG reposent soit sur la fréquence des mots, comme le modèle BM25 [5], soit sur la similarité vectorielle entre le plongement lexical (*embedding*) de la requête et celui des documents, en utilisant des algorithmes tels que *K-Nearest Neighbours* (KNN) ou *Approximate Nearest Neighbours* (ANN) [6]. Avec cette dernière méthode, de loin la plus répandue actuellement, la qualité de la récupération dépend directement du modèle d'*embedding* utilisé. En effet, le modèle employé va mettre l'accent sur certains aspects de la question mais en ignorer d'autres.

De manière plus générale, la qualité de la récupération dépend fortement de la formulation précise de la question ; même une simple erreur orthographique peut réduire significativement la pertinence des documents sélectionnés. Les systèmes de recherche d'information (RI) sont connus pour être sensibles à la variation de leur entrée et moins efficaces lorsqu'ils sont confrontés à des requêtes ambiguës ou très courtes. Ainsi, les documents réellement pertinents sont souvent écartés au profit de ceux qui sont plus proches de la formulation de la question. Dès lors, il sera intéressant de s'inspirer des différents travaux en recherche d'information pour éviter ce phénomène.

Un moteur de recherche commence généralement par une phase de *query understanding*, consistant à extraire de la question l'information pertinente. Cette phase comporte le plus souvent une étape d'élimination des erreurs d'orthographe, de lemmatisation, mais aussi de reformulation (*query rewriting*) en une requête capturant mieux l'intention de la recherche. Optionnellement, une expansion de la requête (*query expansion*) [7] est réalisée pour améliorer la performance de l'extraction, est réalisée. Les techniques classiques consistaient à établir des dictionnaires de synonymes et de termes proches sémantiquement, ou à utiliser des ressources annotées. Cependant, on s'attendait à ce que le passage par des *embeddings* assure déjà l'utilisation de synonymes.

L'intersection de la recherche d'information (RI) et des LLMs a déjà été explorée sous deux angles. Certains auteurs, focalisés sur l'amélioration du RAG, présentent l'intégration de l'expansion de requêtes comme une piste intéressante à creuser [8]. Ainsi, Ma et al. [9] soulignent que, bien que plusieurs équipes de chercheurs aient essayé d'améliorer les modèles d'extraction de documents en les alignant par fine-tuning sur le modèle de génération lors d'une phase d'entraînement commune, peu de travaux ont été réalisés en amont pour tenter de transformer la question de l'utilisateur. Ils suggèrent donc d'investiguer une nouvelle architecture en trois étapes : *rewrite*, *retrieve*, *read*. La transformation qu'ils proposent est d'ordre vectoriel, modifiant le modèle d'*embedding* de la question de l'utilisateur *via* un apprentissage par renforcement. Or, modifier un modèle de cette manière nécessite de disposer d'un grand corpus avec des questions, les documents pertinents et des propositions de réponse, ce qui rend le processus non reproductible dans tout domaine de spécialité sans un lourd travail d'annotation.

Avant que l'architecture RAG ne se démocratise, des chercheurs comme Claveau [10] ont tenté, avec des résultats prometteurs, d'utiliser des modèles génératifs pour effectuer une expansion de requêtes dans le but d'améliorer des systèmes de recherche documentaire [11]–[13]. Cette approche, plus facilement généralisable, a donné lieu à de nombreux travaux établissant le lien entre le paradigme de *query expansion* et le *prompt engineering* [14], ainsi qu'à la proposition de différentes stratégies d'expansion.

Néanmoins, si l'expansion de requêtes est une technique prometteuse pour augmenter le rappel de la phase de récupération, s'appuyer sur des LLMs, susceptibles aux hallucinations [15] risque de dégrader les performances du système, notamment sa précision. Dès lors, comment intégrer l'expansion à l'architecture RAG de manière à limiter les risques ponctuels de dégradation des résultats ?

Cet article se propose :

- d'opérer une présentation synthétique des différentes stratégies d'expansion de requêtes par LLMs et de comparer leurs apports et défauts.
- de proposer une nouvelle architecture RAG plus complexe intégrant l'expansion de requêtes au moyen d'un routeur sémantique et procédant à une agrégation des documents en cas de requêtage multiple.
- de comparer cette nouvelle architecture à celle du RAG classique en les testant sur un corpus spécialisé.
- de proposer un début de réflexion sur l'évaluation non triviale des nouveaux composants de ce système.

II. Stratégies d'expansion de requêtes

Un certain nombre de travaux propose des techniques de reformulation de la question de l'utilisateur pour améliorer les résultats d'un RAG. Si certaines de ces idées semblent prometteuses, la rigueur de leur présentation et de l'évaluation est très variable. Ces méthodes ont rapidement été implémentées dans des frameworks permettant de construire une architecture RAG en quelques lignes de code, mais selon une implémentation opaque pour l'utilisateur. À notre connaissance, aucune étude à ce jour n'a regroupé de manière exhaustive ou comparé ces techniques. Il convient donc de déterminer leurs mérites respectifs.

A. Le Step Back

Cette technique est directement inspirée d'un des premiers articles sur le *prompt engineering*, qui suggérait qu'encourager un LLM à décomposer sa réponse en lui demandant « *take a deep breath and think* » améliorait ses performances sur des tâches de

raisonnement [16]. Le principe du *Step Back* [17] est d'utiliser un LLM avec un prompt demandant de simplifier la question de l'utilisateur en prenant du recul. On obtient alors une question plus générale, ce qui risque d'apporter des documents fournissant plus d'informations contextuelles et d'éviter que la formulation de la question initiale n'influence trop la récupération. Cette technique peut être considérée comme l'équivalent de l'emploi d'hyperonymes dans la phase de *query rewriting* d'un moteur de recherche. Cependant, effectuer un *Step Back* n'est pas nécessairement bénéfique, surtout si la question posée demandait une information précise, ce qui pourrait entraîner une baisse de précision.

B. La reformulation de question (query rewriting)

Cette technique consiste à demander à un LLM de reformuler la question de base de plusieurs manières différentes. En combinant les résultats de recherches effectuées à partir de formulations différentes, on espère récupérer tous les documents intéressants, indépendamment de la lettre exacte de la question de départ. Cette stratégie peut être considérée comme l'équivalent de l'emploi de synonymes dans la phase de *query rewriting* d'un moteur de recherche.

C. La décomposition de requête (query decomposition)

Cette méthode consiste à demander à un LLM de décomposer la question en sous-questions dont les réponses sont nécessaires pour pouvoir répondre à la question globale. Cette décomposition tire parti des capacités de raisonnement et de planification des LLMs et peut faciliter l'extraction d'informations précises, qui, combinées lors de la génération finale, permettent de trouver la bonne réponse (par exemple, « Quelle ville entre Paris et Londres est la plus peuplée ? » se décompose en [« Combien d'habitants y a-t-il à Londres ? », « Combien d'habitants y a-t-il à Paris ? »]). Sur des questions plus abstraites, cette technique peut encourager l'exploration de différentes pistes concrètes étayées par des récupérations plus précises. On utilise donc les capacités de raisonnement des LLMs pour améliorer le résultat de la collecte d'information. L'idée de décomposer une requête en sous-requêtes thématiques est déjà explorée en RI depuis bientôt 20 ans (*Topical Query Decomposition*) [18].

D. Document fictif / HyDE

Lors de la récupération, on compare par similarité vectorielle une question (modalité interrogative, phrase généralement assez brève) à un extrait de document (modalité déclarative, texte plus long, factuel) ce qui peut entraîner une dissonance sémantique qui peut expliquer l'imprécision des résultats. Pour pallier ce problème, il faudrait comparer des objets similaires, ce qui implique soit de créer pour chaque document une liste de questions auxquelles il répond et effectuer la recherche de similarité entre les questions, soit de convertir la question en document fictif. Associer des questions à chaque document entraîne une indexation coûteuse et inefficace si la question diffère trop des questions indexées. La piste du document fictif a été développée dans l'article [19] qui présente la technique HyDE *Hypothetical Document Embeddings*. Si cette proposition est stimulante, utiliser un LLM de la sorte risque d'introduire du bruit car il pourrait halluciner lors de la génération du document fictif, et le contenu de ces hallucinations serait utilisé pour calculer les similarités. L'idée de partir d'un document pour étendre une requête rappelle la technique PRF (*Pseudo-Relevance Feedback*) [20] employée en RI depuis les années 80 et qui consiste à améliorer les résultats en enrichissant automatiquement la requête initiale avec des termes extraits des premiers documents jugés pertinents.

E. Autres stratégies

D'autres stratégies comme par exemple *Least-to-Most prompting* [21] ont été proposées notamment pour traiter les questions à rebonds multiples (*multi hop questions* comme « Quel est le nom du mari de la fille du maire de Héron-ville? »). Ces méthodes nécessitent une architecture itérative différente de celle considérée dans le présent article et ne seront donc pas approfondies.

F. Bilan

Chaque stratégie présente des intérêts particuliers et aide à répondre à certains types de questions, au risque de parfois dégrader la réponse de questions simples et factuelles pour lesquelles la récupération avec la *query* initiale donnait une réponse correcte.

Technique d'expansion	**Types de question**
Step Back	Question comportant des détails trop précis
Reformulation	Question ambiguë ou à la formulation trop spécifique
Décomposition	Question abstraite ou générale comportant plusieurs dimensions, Question dont la réponse nécessite un raisonnement
HyDE / document fictif	Question où la réponse apparaît dans un type de document spécifique

TABLE 1: Usage respectif des stratégies d'expansion selon le type de question posée

Transformer la requête pourrait dégrader les résultats de la recherche compte tenu de la propension des LLMs à commettre des hallucinations. Pour limiter

ce risque, il serait prudent de ne pas utiliser uniquement la requête transformée pour la récupération mais d'amalgamer ces résultats avec ceux de la question originale. De plus, certaines stratégies proposent plusieurs reformulations de la question. Il convient donc de complexifier l'architecture RAG classique pour prévoir le choix d'une méthode d'expansion et possiblement la gestion de plusieurs récupérations en parallèle à partir de différentes requêtes étendues.

III. Adaptation de l'architecture RAG à l'expansion de requêtes

A. Nouvelle architecture proposée

L'intégration de l'expansion de requêtes à de potentielles récupérations multiples soulève plusieurs questions :

- Comment choisir la stratégie d'expansion?
- Comment optimiser la transformation de la requête ?
- Comment amalgamer les résultats produits par différents processus de récupération ?

Chacun de ces trois rôles sera pris en compte par un nouveau composant de l'architecture. En effet l'architecture du RAG classique ne suffit plus à englober un tel système et doit être élargie de façon modulaire [22]. Un module de redirection (ou routeur sémantique) est chargé de déterminer si la question nécessite une expansion, et le cas échéant quelle technique serait la plus bénéfique. Un module d'expansion est chargé de procéder à la transformation de la requête. Après les phases de récupération parallèles, un module d'agrégation amalgame les résultats (voir annexe 1 pour plus de détails).

La phase de récupération peut inclure un *reranking* pour affiner les résultats de similarité vectorielle. Des modèles spécialisés, comme les *cross encoders* sont utilisés pour évaluer la pertinence d'un document par rapport à une requête et produire un classement plus précis [23].

B. Le routeur : rôle et fonctionnement

L'étude de la littérature a permis de sélectionner quatre stratégies d'expansion prometteuses. Néanmoins, chaque article n'évalue que la méthode qu'il introduit sans la comparer à d'autres méthodes d'expansion. De plus, pour les questions simples (« Quelle est la capitale de la France ? »), aucune expansion n'est *a priori* nécessaire. Pour éviter qu'un emploi systématique de l'expansion ne dégrade la précision du système sur les questions simples, il faudrait pouvoir sélectionner la bonne action à effectuer sur une requête (expansion ou non, et le cas échéant quelle stratégie). Pour ce faire, il est possible de s'inspirer de la notion de *routing*, également issue des travaux sur les moteurs de recherche.

Un LLM peut être employé comme agent responsable de la redirection comme démontré par certains travaux [24]. Il s'agit alors d'écrire un prompt décrivant la tâche et les outils disponibles, avec éventuellement des exemples via *few-shot prompting*. La réponse du LLM est analysée pour déterminer la technique d'expansion choisie. Cependant, utiliser un LLM n'offre pas toujours une explication claire du choix de la méthode. Par exemple, l'ordre de présentation des stratégies d'expansion peut influencer les résultats (*order bias*) [25] et sélectionner un mode d'expansion n'est pas trivial.

Notre implémentation du routeur a été testée sur un benchmark de 150 questions réparties équitablement en 5 catégories de difficulté progressive.

- Catégorie 1: questions factuelles admettant une réponse unique.
- Catégorie 2: questions factuelles admettant plusieurs réponses.
- Catégorie 3: questions nécessitant la présentation structurée d'un déroulé chronologique ou d'une chaîne de causalité.
- Catégorie 4: questions abstraites.
- Catégorie 5: questions volontairement ambiguës ou contenant des erreurs.

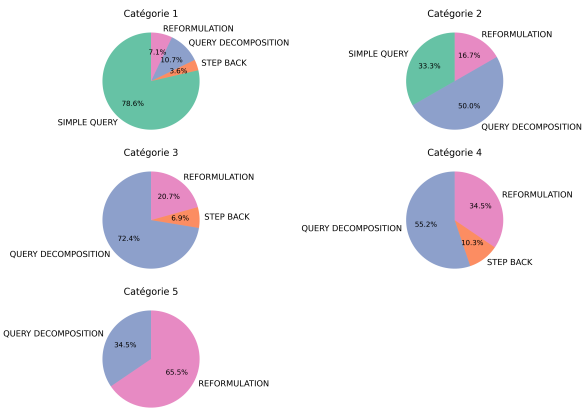


Fig. 1: Choix du routeur de la stratégie d'expansion selon la catégorie de difficulté de la question

On a ensuite analysé les choix du routeur en fonction de la catégorie de difficulté des questions. Pour la plupart des questions simples (deux premières catégories) le routeur choisit de ne pas réaliser d'expansion. Quand la question est plus complexe et nécessite un raisonnement (catégories 3 et 4), le routeur choisit

le plus souvent d'effectuer une décomposition. Enfin lorsque la question est ambiguë ou contient des informations fausses, c'est le plus souvent une reformulation qui est appliquée. Ce comportement correspond *a priori* aux attentes qu'on avait d'un routeur qui n'utilise l'expansion que lorsque nécessaire. Pour s'assurer du bon fonctionnement du LLM en tant que routeur, les trois auteurs ont associé à chaque question d'un corpus de difficulté variable la méthode d'expansion qu'ils considéraient comme la plus appropriée et ont obtenu des distributions de stratégies similaires à celle du LLM du moins en ce qui concerne le choix de réaliser ou non une expansion. L'hésitation est plus grande sur la méthode d'expansion à appliquer.

À noter que la technique du document fictif n'est jamais choisie. Cela peut s'expliquer par le fait que le bénéfice de cette méthode est moins clair que celui d'une reformulation par exemple pour le LLM chargé d'effectuer le *routing*. La description de l'utilité de l'outil qui devait rester succincte dans le prompt n'encourage pas le LLM à la choisir. La méthode de *Step Back* est également très peu utilisée peut-être parce que ses objectifs recoupent en partie ceux plus larges de la reformulation.

C. Algorithmes d'agrégation

Pour limiter les risques qu'une mauvaise expansion ne dégrade les résultats du système, nous n'utilisons pas uniquement la requête modifiée, comme c'est souvent le cas dans d'autres travaux, mais effectuons plusieurs récupérations en parallèle à partir de la requête initiale et de ses expansions. Cela est particulièrement nécessaire lorsque la méthode choisie génère plusieurs reformulations.

Dans l'architecture RAG classique, chaque processus de récupération est limité à j documents. La nouvelle architecture peut potentiellement extraire j documents par récupération soit jusqu'à $(q+1) \times j$ documents, où q correspond au nombre de requêtes modifiées issues de l'expansion, et 1 représente la requête initiale, à condition qu'aucun doublon ne soit présent.

Néanmoins, la fenêtre de contexte des LLMs ayant une taille limitée, il ne sera pas possible d'inclure tous ces documents dans le prompt final. Il convient de plus de les ordonner du plus au moins pertinent pour optimiser la génération finale. Ce problème rejoint un problème plus général en algorithmique qui est celui de la fusion de classements. Le problème n'est pas symétrique car on souhaite accorder plus d'importance aux documents issus de la question initiale qu'aux requêtes modifiées. Cette agrégation des résultats peut s'effectuer selon différentes stratégies :

1) Reranking par rapport à la query principale: Une possibilité serait d'amalgamer tous les documents, d'éliminer les doublons puis d'utiliser un modèle de *reranking* pour reclasser tous les documents en fonction de leur pertinence par rapport à la question initiale. Les documents sont classés une première fois par rapport à la requête qui avait servi à les extraire. On effectue ensuite un second classement en fonction de la question initiale sur tous les documents en même temps. L'avantage de cette méthode est qu'elle limite le bruit possiblement apporté par l'expansion de requêtes au risque de pénaliser des documents intéressants mais plus éloignés sémantiquement de la question initiale. Cette méthode d'agrégation favorise la précision au dépend du rappel.

2) Agrégation par ratio: L'agrégation la plus simple consiste à sélectionner un ratio fixe de documents provenant de la requête principale. Si l'on souhaite obtenir au total k documents, et que l'on fixe un ratio α pour les documents issus de la requête principale, on sélectionne alors les $\alpha \times k$ meilleurs documents récupérés à partir de la requête initiale. Après élimination des doublons, les $(1-\alpha) \times k$ documents restants sont sélectionnés parmi les récupérations des différentes propositions d'expansion. Cette méthode favorise la diversité dans les documents récupérés, mais reste sensible à la qualité de l'expansion effectuée par le LLM et peut entraîner une dégradation de la précision et une augmentation du bruit dans la génération finale.

3) Reciprocal Rank Fusion: L'algorithme de *Reciprocal Rank Fusion*, développé dans le domaine de la recherche d'information [26], calcule un score pour chaque document en fonction de son rang dans les différents classements fusionnés. Les documents ayant les meilleurs scores sont ensuite sélectionnés. La formule pour calculer le score d'un document d est la suivante :

$$\text{RRFScore}(d \in D) = \sum_{r \in R} \frac{1}{s + r(d)}$$

où $r(d)$ est le rang du document d dans chaque classement, et s est un coefficient de lissage (*smoothing coefficient*) ajouté au dénominateur pour réduire l'impact des rangs élevés. Ce coefficient permet d'éviter de survaloriser les documents en tête de classement tout en favorisant ceux qui apparaissent dans plusieurs classements. Cette méthode valorise donc la diversité tout en limitant l'impact des documents issus de requêtes mal formulées.

4) Algorithme hybride: Afin de donner un statut particulier à la récupération par rapport à la requête initiale, nous avons proposé une implémentation hybride. Celle-ci permet de réserver un certain ratio α aux documents récupérés via la question initiale, puis, après dédoublonnage, d'appliquer l'algorithme de *Reciprocal Rank Fusion* (RRF) aux documents issus des autres récupérations (fusion des méthodes 2 et 3).

Pour obtenir un total de k documents, on sélectionne d'abord $\alpha \times k$ documents provenant de la requête principale, puis $(1 - \alpha) \times k$ documents issus de l'application de RRF sur les autres récupérations.

Ces algorithmes de classement permettent ainsi de combiner les résultats de plusieurs recherches et d'obtenir un nouveau classement trié du plus pertinent au moins pertinent.

5) Lost in the middle: Une étude récente [27] a montré que les informations situées au centre du contexte d'un LLM risquent d'être ignorées lorsque celui-ci est trop long, un phénomène appelé *« lost in the middle »*. Bien que ce problème ait été corrigé dans certains modèles récents comme Gemini de Google [28], il persiste dans d'autres, tels que Llama2 [29].

Pour éviter cet oubli, il est recommandé de réordonner les documents : placer les plus importants au début et à la fin, et les moins pertinents au centre (par exemple, pour 6 documents : [1 3 5 6 4 2]), ce qui est trivial à implémenter.

IV. Résultats et Évaluation

Il s'agira de déterminer si la nouvelle architecture RAG proposée obtient de meilleurs résultats que l'architecture classique en comparant les deux systèmes sur un benchmark de questions posées sur un corpus spécialisé. On réfléchira ensuite à la difficile évaluation d'un système de RAG et des nouveaux composants introduits.

A. Jeu de données

On illustrera principalement le présent travail à partir d'une base de données d'un peu plus de 80 000 documents, issus de sources ouvertes sur le web, traitant le conflit au Mali. Cette base de données est tout à fait adéquate pour tester l'efficacité du RAG car elle présente certaines difficultés absentes des benchmarks traditionnels. En effet, la base de données est en français et contient un nombre très important de documents. De plus, les documents font référence à un domaine spécifique. Sur ce jeu de données, un corpus de 150 questions de difficultés progressives a été conçu sur la thématique de la base [cf III.3].

B. Détails sur l'implémentation utilisée

Dans les deux architectures évaluées : Le *retriever* utilisé est un modèle hybride combinant un module de recherche par mots clés et un module de similarité sémantique (*embeddings* du modèle paraphrase-multilingual-mpnet-base-v2[1]) [30] reposant sur l'algorithme HNSW [31] tel qu'implémenté nativement dans le moteur de recherche ElasticSearch. Le *reranker* utilisé est un *cross-encoder* bert-multilingual-passage-reranking-msmarco[2] [32] fine tuné à partir de BERT. Pour les étapes de *routing*, la transformation des requêtes ainsi que la génération finale, le modèle Llama 2 13B Chat[3] de Meta a été employé. Ces modèles ont été installés sur deux cartes graphiques A6000. Le mode d'agrégation choisi pour les résultats présentés est la méthode hybride.

C. Méthodes d'évaluations du RAG

Évaluer l'architecture RAG est complexe car elle intègre divers blocs et modèles pré-entraînés ce qui la rend très sensible au phénomène de propagation de l'erreur. Il est de plus important d'évaluer chaque composant individuellement, ce qui peut se faire de façon intrinsèque (comparaison de sa sortie avec des références) ou extrinsèque (impact de la modification du composant sur la qualité globale du système). Les défis incluent la difficulté d'évaluer les réponses finales, qu'elles soient factuelles ou abstraites, et le besoin de méthodes d'évaluation fiables sans corpus annotés coûteux.

De nombreux laboratoires et entreprises ont exploré l'évaluation du RAG sans aboutir à une méthode unifiée. Un *survey* récent [33], bien qu'ayant le mérite de rassembler des réflexions sur les métriques possibles, ne propose pas de protocole d'évaluation universel. De plus, il se concentre uniquement sur l'architecture RAG classique, sans envisager l'ajout de nouveaux composants. Parmi les frameworks d'évaluation du RAG, on peut citer ARES [34], RAGAS [35] et RAGET[4].

Ces *packages* implémentent des métriques basées principalement sur des comparaisons entre étapes intermédiaires de la réponse, permettant d'évaluer le comportement d'un composant spécifique. Si certains frameworks utilisent des modèles de type BERT pour évaluer certains critères, la plupart adoptent l'approche

[1] https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2

[2] https://huggingface.co/amberoad/bert-multilingual-passage-reranking-msmarco

[3] https://huggingface.co/meta-llama/Llama-2-13b-chat

[4] https://docs.giskard.ai/en/stable/open_source/testset_generation/rag_evaluation/index.html

« LLM as a judge », où un LLM est utilisé pour juger la sortie d'un autre LLM [36]. Bien que des doutes subsistent quant à la validité des scores numériques et aux biais potentiels [37], [38], cette méthode s'est largement imposée dans l'industrie ces derniers mois. Elle présente l'avantage de ne pas nécessiter d'annotation manuelle et s'adapte facilement à différents corpus.

Devant le manque de consensus, nous avons décidé d'implémenter plusieurs métriques générales pour évaluer le comportement de certains composants du pipeline, notamment la phase de récupération (*retrieve*). Ces métriques consistent à comparer deux à deux les principaux états du système, à savoir la requête de l'utilisateur, les documents sélectionnés et la réponse finale.

Les métriques implémentées sont les suivantes :

1) **Context Relevance** : Les passages sélectionnés sont-ils pertinents pour répondre à la question de l'utilisateur ? (évaluation de la phase de récupération)
2) **Context Adherence/Factuality** : La génération finale est-elle fidèle aux extraits fournis dans la fenêtre de contexte, ou contient-elle des passages hallucinés ? (évaluation de la génération finale et de sa fidélité aux informations fournies)
3) **Overall Quality** : La réponse finale répond-elle de façon satisfaisante et correcte à la question de l'utilisateur ?

Ces métriques peuvent être calculées en établissant des ratios entre les entités nommées présentes dans les états comparés, ou en utilisant l'approche « LLM as a judge » basée sur des prompts inspirés des méthodes RAGAS et ARES.

D. Protocole d'évaluation

Les deux architectures ont été implémentées et connectées à la base de données. Les métriques ont ensuite été calculées sur les réponses des systèmes au benchmark III-B.

Mesurer l'influence de l'expansion de requêtes n'est pas évident car elle n'est pas déclenchée de façon systématique (rôle du routeur). Son apport est conditionné par le comportement d'autres composants (*retriever*, *reranker*, taille et contenu de la base de données). Pour vérifier si l'expansion de requêtes fonctionne correctement, il convient donc de déterminer :

- Si elle dégrade les réponses par rapport à un RAG classique.
- Si la diversité et le nombre de documents rapportés est plus importants.

1) Evaluation individuelle des composants:
Chaque composant a été évalué de façon intrinsèque:

- Le routeur a été testé sur un benchmark hétérogène pour s'assurer que son comportement correspondait aux attentes. Ses productions ont été comparées avec le comportement d'annotateurs.
- Les méthodes d'expansion ont été évaluées sur ce même benchmark et contrôlées par un évaluateur humain à défaut d'avoir un corpus de référence auquel comparer les résultats (phase d'optimisation des prompts).
- Les algorithmes d'agrégation étant déterministes, c'est le choix de l'algorithme à utiliser et éventuellement de ses paramètres à ajuster qui pourra être affiné par une évaluation extrinsèque.

E. Mesure quantitative des apports de l'expansion de requêtes

Il est possible de faire une évaluation quantitative en étudiant le nombre de nouveaux documents apportés par l'expansion. Avec une agrégation hybride, on a ainsi constaté environ 15 % de nouveaux documents.

F. Mesure qualitative des apports de l'expansion de requêtes

Les résultats des différentes métriques pour les deux architectures sont présentés en annexe 4.

On constate que si la performance du système sur les questions simples n'a pas évolué (catégories 1 et 2), ce qui est attendu car le routeur a dû ne pas sélectionner d'expansion (voir III.2), elle a progressé sur les questions plus abstraites (catégories 4 et 5). Ainsi la qualité moyenne des réponses augmente de 4%. La note de *context adherence* donnée par le LLM s'améliore avec l'expansion, ce qui indique que le LLM utilisé comme juge constate un meilleur usage des contextes fournis et moins de segments de réponse hallucinés.

Néanmoins, on constate une légère baisse (environ 2 %) de la pertinence du contexte (*context relevance*), ce qui n'est pas surprenant, car les contextes apportés par l'expansion peuvent être plus éloignés de la question originale. Cependant, cette présence de documents exotiques ne semble pas avoir nui à la qualité générale des réponses, qui a tout de même progressé.

G. Illustration des bénéfices de l'expansion

L'expansion de requêtes est particulièrement fructueuse pour les questions pour lesquelles peu de documents sont présents dans la base de données. Même si le corpus constitué sur le Mali traite

principalement de questions politiques, quelques documents évoquent la situation des femmes et sont plus difficiles à retrouver dans la base de données. L'annexe 5 fournit un exemple de résultat avec et sans expansion de requête.

Les sous-questions proposées explorent divers aspects du problème principal, ou bien ses liens avec différents acteurs, élargissant ainsi le champ de la recherche. D'un point de vue quantitatif, la recherche avec expansion de requêtes retourne un plus grand nombre de documents.

Sans expansion, seuls cinq documents sont retrouvés, parmi lesquels quatre sont réellement pertinents. Le cinquième document est hors sujet, puisqu'il mélange la politique de François Hollande sur les femmes en France et sa politique étrangère au Mali. En revanche, avec l'expansion, sept documents sont retournés, incluant les cinq de départ.

La sous-question 4, portant sur l'éducation, a permis d'identifier un nouveau document parlant d'un projet visant à aider les jeunes filles maliennes à accéder à l'école. Par exemple :

> « Au Mali, le projet SWEDD distribue des bicyclettes permettant aux jeunes filles d'aller à l'école, et vient en aide aux sage-femmes afin qu'elles puissent assurer des services de santé prénatale, natale et postnatale dans les zones pauvres, réduisant ainsi la mortalité maternelle et infantile. »

Ce document s'est révélé particulièrement pertinent pour approfondir la réponse à la question, et il a même été classé premier par le *reranker*. De manière générale, une amélioration notable des résultats a également été constatée pour des questions plus abstraites nécessitant d'explorer différents aspects d'un même problème.

V. Critiques et conclusion

Cet article dresse un bilan des différentes techniques d'expansion de requêtes et propose une nouvelle architecture RAG permettant leur intégration. Bien qu'on démontre qu'elle surpasse l'architecture RAG classique, notamment en termes de rappel, il reste encore à évaluer le comportement global du système. L'évaluation complète du RAG et de chacun de ses composants nécessite des études supplémentaires.

Pour évaluer plus finement les bénéfices de l'expansion de requêtes, on pourrait comparer les réponses avec et sans expansion en demandant à un LLM de juger (selon l'approche dite de *pairwise comparison*) ou en sollicitant des annotateurs humains.

La présente étude a été menée sur un corpus spécialisé. Il conviendrait de vérifier la portabilité de l'architecture proposée à de nouveaux domaines de spécialité en la testant sur d'autres corpus avec des *benchmarks* de taille plus importante.

Des études d'ablation pourraient enfin permettre de déterminer le rôle de chaque composant dans l'amélioration des performances du système.

En absence de corpus de référence, on a évalué la pertinence de la récupération au moyen d'une métrique sur les entités nommées ou de l'approche « LLM as a judge », mais il serait pertinent d'utiliser les métriques classiques sur le *retrieval* (*Exact Match Ratio*, *Mean Average Precision*, *R-Precision*...) à condition d'avoir un corpus annoté.

L'objectif de l'article était de proposer une nouvelle architecture incorporant différentes stratégies d'expansion de requêtes, et pas de comparer le mérite respectif de chaque stratégie. Néanmoins une étude des résultats de l'application systématique de chacune des stratégies serait intéressante et permettrait notamment de vérifier le bien fondé de la technique du document fictif jamais sélectionnée par le routeur. De même, il conviendrait de tester les différentes techniques d'agrégation implémentées pour voir laquelle trouve le meilleur *trade-off* entre précision et rappel.

Bibliographie

[1] Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. 2021. All That's "Human" Is Not Gold: Evaluating Human Evaluation of Generated Text. arXiv:2107.00061.

[2] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv:2005.11401.

[3] Christophe Bouvard, Mathieu Ciancone, Antoine Gourru, and Marion Schaeffer. 2024. Derby LLM: Évaluation comparative des approches RAG et fine-tuning. In Catherine Roussey Ghislain Atemezing, editor, 10 ème Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle, pages 38–47, La Rochelle, France. AFIA-Association Française pour l'Intelligence Artificielle. Backup Publisher: AFIA-Association Française pour l'Intelligence Artificielle.

[4] Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The Power of Noise: Redefining Retrieval for RAG Systems. arXiv:2401.14887.

[5] M. E. Maron and J. L. Kuhns. 1960. On Relevance, Probabilistic Indexing and Information Retrieval. J. ACM, 7(3):216–244.

[6] Alexandr Andoni, Piotr Indyk, and Ilya Razenshteyn. 2018. Approximate Nearest Neighbor Search in High Dimensions. arXiv:1806.09823.

[7] Hiteshwar Kumar Azad and Akshay Deepak. 2019. Query expansion techniques for information retrieval: A survey. arXiv:1708.00247.

[8] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024a. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997.

[9] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query Rewriting for Retrieval-Augmented Large Language Models. arXiv:2305.14283.

[10] Vincent Claveau. 2020. Query expansion with artificially generated texts. arXiv:2012.08787.

[11] Lukas Gienapp, Harrisen Scells, Niklas Deckers, Janek Bevendorff, Shuai Wang, Johannes Kiesel, Shahbaz Syed, Maik Fröbe, Guido Zuccon, Benno Stein, Matthias Hagen, and Martin Potthast. 2024. Evaluating Generative Ad Hoc Information Retrieval. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 1916–1929, Washington DC USA. ACM.

[12] Gautier Izacard and Edouard Grave. 2021. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 874–880, Online. Association for Computational Linguistics.

[13] Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query Expansion by Prompting Large Language Models. arXiv:2305.03653.

[14] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. arXiv:2402.07927.

[15] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. arXiv:2311.05232.

[16] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2024. Large Language Models as Optimizers. arXiv:2309.03409.

[17] Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2024. Take a Step Back: Evoking Reasoning via Abstraction in Large Language Models. arXiv:2310.06117.

[18] Francesco Bonchi, Carlos Castillo, Debora Donato, and Aristides Gionis. 2008. Topical query decomposition. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 52–60, New York, NY, USA. Association for Computing Machinery.

[19] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise Zero-Shot Dense Retrieval without Relevance Labels. arXiv:2212.10496.

[20] Hang Li, Ahmed Mourad, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. 2022. Pseudo Relevance Feedback with Deep Language Models and Dense Retrievers: Successes and Pitfalls. arXiv:2108.1104.

[21] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. arXiv:2205.10625.

[22] Yunfan Gao, Yun Xiong, Meng Wang, and Haofen Wang. 2024b. Modular RAG: Transforming RAG Systems into LEGO-like Reconfigurable Frameworks. arXiv:2407.21059.

[23] Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage Reranking with BERT arXiv:1901.04085.

[24] Yuheng Cheng, Ceyao Zhang, Zhengwen Zhang, Xiangrui Meng, Sirui Hong, Wenhao Li, Zihao Wang, Zekai Wang, Feng Yin, Junhua Zhao, and Xiuqiang He. 2024. Exploring Large Language Model based Intelligent Agents: Definitions, Methods, and Prospects. arXiv:2401.03428.

[25] Sheng-Lun Wei, Cheng-Kuang Wu, Hen-Hsen Huang, and Hsin-Hsi Chen. 2024. Unveiling Selection Biases: Exploring Order and Token Sensitivity in Large Language Models. arXiv:2406.03009.

[26] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, pages 758–759, New York, NY, USA. Association for Computing Machinery.

[27] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the Middle: How Language Models Use Long Contexts. arXiv:2307.03172.

[28] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, et al. 2024. Gemini: A Family of Highly Capable Multimodal Models. arXiv:2312.1180.

[29] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

[30] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084.

[31] Yu A. Malkov and D. A. Yashunin. 2018. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. arXiv:1603.09320.

[32] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. arXiv:1611.09268. arXiv:1708.00247.

[33] Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. 2024. Evaluation of Retrieval-Augmented Generation: A Survey. arXiv:2405.07437.

[34] Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2024. ARES: An Automated Evaluation Framework for Retrieval-Augmented Generation Systems. arXiv:2311.0947.

[35] Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. RAGAs: Automated Evaluation of Retrieval Augmented Generation. In Nikolaos Aletras and Orphee De Clercq, editors, Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.

[36] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. arXiv:2306.0568.

[37] Shreya Shankar, J. D. Zamfirescu-Pereira, Björn Hartmann, Aditya G. Parameswaran, and Ian Arawjo. 2024. Who Validates the Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences. arXiv:2404.12272.

[38] Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023. Large Language Models are not Fair Evaluators. arXiv:2305.17926.

ANNEXES

A. Annexe 1 : Nouvelle architecture proposée

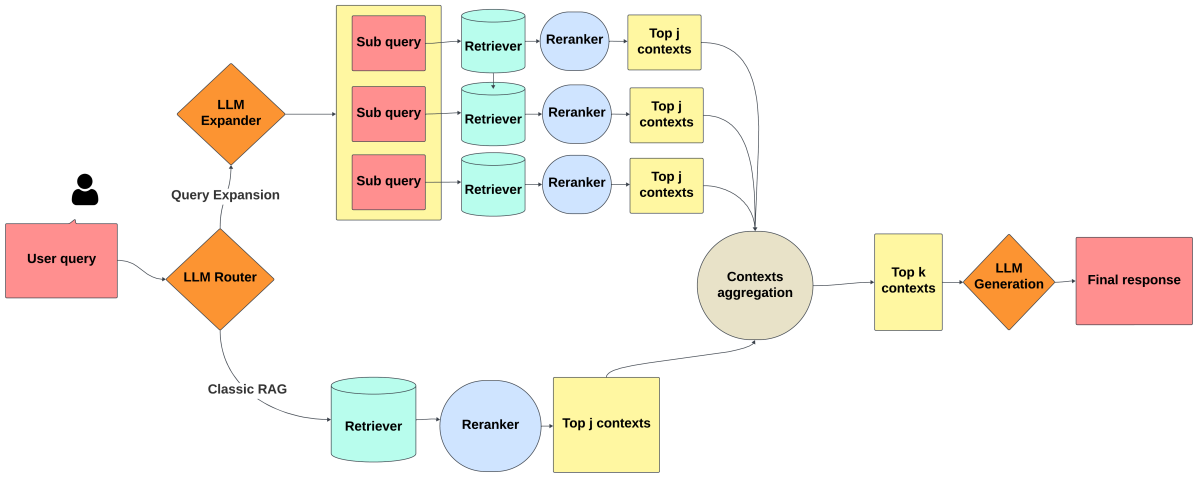


Fig. 2: Architecture RAG intégrant l'expansion de requêtes.

B. Annexe 2 : Exemples de questions du benchmark de difficulté progressive

Catégorie	Exemples de questions
Catégorie 1 : Questions factuelles demandant une réponse unique	Quel est le nom du fleuve qui traverse le Mali ? Quelle est la principale ressource naturelle exploitée au Mali ? Quelle est la date d'indépendance du Mali ?
Catégorie 2 : Questions factuelles attendant plusieurs éléments de réponse	Nommez trois organisations régionales africaines impliquées dans la résolution de la crise malienne. Donnez trois exemples de mesures prises par le gouvernement malien pour lutter contre le terrorisme. Quels sont les groupes armés signataires de l'Accord pour la paix issu du processus d'Alger ?
Catégorie 3 : Questions nécessitant la présentation structurée d'un déroulé chronologique ou d'une chaîne de causalité	Décrivez la séquence d'événements qui ont conduit à l'accord de paix d'Alger et expliquez son impact sur la situation politique au Mali. Reconstituez la chronologie des négociations et des accords visant à résoudre la crise malienne depuis le début du conflit. Analysez le parcours et l'impact d'Amadou Toumani Touré sur la scène politique du Mali.
Catégorie 4 : Questions abstraites	Comment le changement climatique peut-il aggraver les tensions et les conflits au Mali ? Comment le trafic illicite, y compris le trafic de drogue, est-il lié au conflit au Mali ? Quel rôle jouent les groupes ethniques dans la dynamique du conflit au Mali ?
Catégorie 5 : Questions volontairement ambiguës ou contenant des erreurs factuelles volontaires	Comment la récente réorganisation des forces armées maliennes a-t-elle amélioré l'efficacité opérationnelle ? Quelles mesures ont été prises pour protéger les droits des minorités ethniques au Mali ? Quelles initiatives ont été lancées pour résoudre la crise énergétique au Mali ?

TABLE 2: Exemples de questions classées par catégories

C. Annexe 3 : Prompts utilisés

1) Prompt utilisé pour le routeur:

As an expert, your task is to answer a user query. You have several strategies available:

- **SIMPLE QUERY**: the query is straightforward and can be answered directly.
- **QUERY DECOMPOSITION**: the query is complex and can be broken down into simpler sub-questions.
- **REFORMULATION**: the query is unclear, poorly written, or abstract.
- **STEP BACK**: the question is too precise; asking a more general question might yield relevant information.
- **FICTIVE DOCUMENT**: the expected answer may be found in a specific type of document (e.g., official statement, news article).

The language of the query is not relevant to pick a strategy. Output the name of the selected strategy in capital letters without any other comment. Then jump a line and briefly justify this choice. Do not be verbose or acknowledge instructions.

`{user_query}`

2) Prompt utilisé pour le Step Back:

As an expert, your mission is to step back from a question that is asked and rephrase it as a more general question. The new question must be easier to answer than the first one.

Examples:

- **Initial query**: "What studies did Assimi Goita pursue before becoming president?"
  Step Back query: "What is Assimi Goita's background?"
- **Initial query**: "Where did the most recent terrorist attack occur that resulted in over ten deaths in Mali?"
  Step Back query: "What were the most deadly terrorist attacks in Mali's recent history?"

Only output the new question.

`{user_query}`

3) Prompt utilisé pour la reformulation de requêtes:

As an expert, your mission is to rephrase the following question in three different ways to make it clearer. Provide three reformulations of the question and return only a numbered list, skipping lines between each question.

`{user_query}`

4) Prompt utilisé pour la décomposition de requêtes:

As an expert, your mission is to generate a list of specific sub-questions that need to be answered to address the user's general question. Here are some guidelines:

- Be precise.
- Each sub-question should be relevant.
- The answer should be findable in a database.
- Provide the results as a numbered list of questions (maximum 5 questions). Return only a numbered list, skipping lines between each question. Do not add any comments.

`{user_query}`

5) Prompt utilisé pour la création de documents hybrides:

Generate a document that answers the given question. The document could be, for instance, extracted from a newspaper article or an official document or be any document you seem fit to contain the answer to the query. Provide only the document.

`{user_query}`

6) Prompt utilisé pour la génération finale:

You're a reliable system that specializes in answering questions. Your mission is to generate an answer based on the question posed and the context provided. You always respond using the contextual information given to you in the context and without using any other information.

Here are a few rules to follow:

1) Answer in `{lang}`.
2) Only consider context that is relevant to answer the question.
3) Do not refer directly to the contextual information provided.
4) Avoid formulations such as "According to the information provided."
5) If the given contexts do not allow you to answer the question, answer "I don't know."

`{user_query}`

D. Annexe 4 : Comparaison des deux architectures RAG

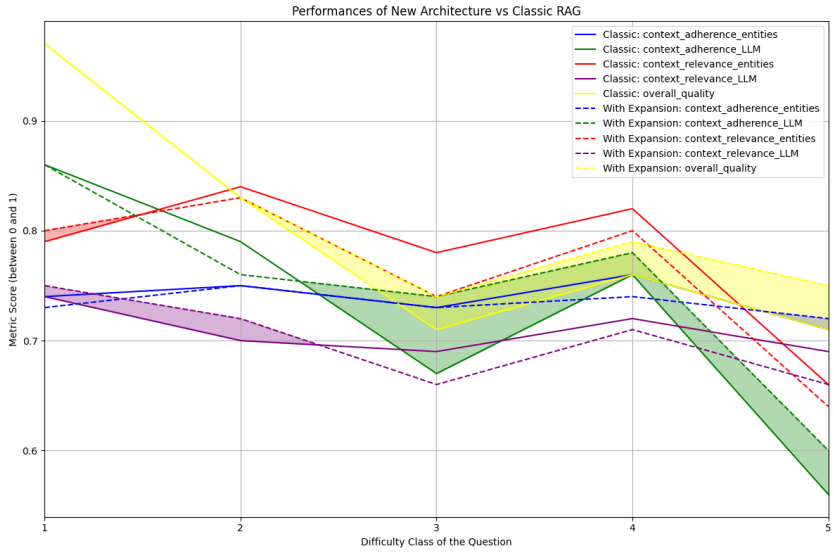


Fig. 3: Résultat de l'évaluation comparative des deux systèmes

E. Annexe 5 : Exemple de documents avec et sans expansion

Question : Détaillez les initiatives prises pour promouvoir l'égalité des sexes et les droits des femmes au Mali.

Réponse du RAG classique	**Réponse avec Expansion de Requête**
Au Mali, plusieurs initiatives ont été prises pour promouvoir l'égalité des sexes et les droits des femmes. Parmi ces initiatives, on trouve : • **Code du mariage et de la tutelle** : Législation visant à garantir les droits des femmes. • **Conférence des femmes africaines de 1962** : Participation à un événement important pour l'unité des femmes en Afrique. • **Unions de femmes** : Soutien au niveau local pour promouvoir les droits des femmes. • **Participation à l'agenda 2030 de l'ONU** : Contribution aux objectifs de développement liés à l'égalité des sexes. • **Action de l'Union européenne** : Soutien à la gouvernance et à l'égalité des sexes dans le cadre d'actions contre l'insécurité.	Au Mali, plusieurs initiatives ont été prises pour promouvoir l'égalité des sexes et les droits des femmes. Parmi ces initiatives, on trouve : • **Projet SWEDD** : Une initiative de grande envergure soutenue par l'ONU pour promouvoir l'égalité des sexes, l'éducation des filles, et la réduction des mariages d'enfants. • **Action de l'Union européenne** : Continuation des efforts pour soutenir la gouvernance et les droits des femmes, avec davantage de détails sur l'implication dans les régions affectées. • **Code du mariage et de la tutelle** : Mention plus précise du rôle des femmes maliennes dans l'élaboration de ce code. • **Conférence des femmes africaines de 1962** : Informations contextualisées sur l'objectif de promotion de l'unité des femmes africaines. • **ONG comme CAFO** : Rôle des ONG nationales dans la lutte contre les obstacles sociaux, économiques et juridiques empêchant les femmes de jouir pleinement de leurs droits. • **Promotion de l'éducation des femmes** : Mention du soutien de la Banque mondiale à l'éducation des filles, visant à garantir les droits des femmes et à prévenir les violations pendant les périodes de conflit.
Sous-questions	**Documents Supplémentaires Apportés Par l'Expansion**
• Quelles organisations sont impliquées ? • Quelles sont les mesures législatives prises ? • Quelles initiatives de sensibilisation et de formation ont été mises en place ?	L'expansion de requête a permis d'apporter plusieurs documents supplémentaires pertinents : • **Document SWEDD** : Détaille les initiatives dans la région du Sahel pour promouvoir l'égalité des sexes, l'éducation, et la sécurité sociale. • **Document sur l'implication des ONG** : Montre le rôle des organisations locales dans la promotion des droits des femmes.

TABLE 3: Comparaison des résultats des deux architectures sur une question précise