

UNIVERSITY OF DEUSTO



Deusto

Universidad de Deusto
Deustuko Unibertsitatea
University of Deusto

DOCTORAL THESIS

**Out-of-Distribution Detection in
Open-World Machine Learning:
New Algorithms, Learning
Scenarios, and Pathways towards
Safe Artificial Intelligence**

Author:

Aitor
MARTINEZ-SERAS

Supervisors:

Prof. Dr. Javier DEL SER
Prof. Dr. Pablo GARCIA-BRINGAS

*A Thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy in the*

Department of Mechanics, Design,
and Industrial Organization

October 31, 2024

*“Which is the most important step a person can take?
It is not the first one, is it?
It’s the next one. Always the next step.”*

Dalinar Kholin, in the book *Juramentada*.

*“You mostly failed. This is life. The longer you live, the more you fail.
Failure is the mark of a life well lived. In turn, the only way to live without
failure is to be of no use to anyone. Trust me, I’ve practiced.”*

Sagaz, in the book *Juramentada*.

*“I am grateful for all that life has brought me, both the joys and the sorrows,
as each experience has shaped me into who I am today, positioning me here,
exactly where I wish to be. Journey before destination. ”*

Aitor Martinez Seras, myself.

UNIVERSITY OF DEUSTO

Abstract

Faculty of Engineering
Department of Mechanics, Design,
and Industrial Organization

Doctoral Degree

Out-of-Distribution Detection in Open-World Machine Learning: New Algorithms, Learning Scenarios, and Pathways towards Safe Artificial Intelligence

by Aitor MARTINEZ-SERAS

As we witness the rapid ascent of artificial intelligence, concerns regarding the safety of AI models have significantly heightened. It encompasses AI alignment and robustness and resilience, the former being the ability to withstand unknown scenarios and the latter the ability to adapt to such new scenarios. All these attributes are becoming essential pillars for future models, which need to be trustworthy to gain acceptance in industries and among the general public. In fact, it is increasingly likely that models will not only need to be inherently trustworthy but may also soon be legally required to demonstrate this characteristic.

Despite the critical importance of safety, the current research landscape shows a notable deficiency in studies focused on it. However, since 2016, the Out-of-Distribution detection paradigm has been developed to enhance model robustness by endowing them with the ability to identify and handle inputs that are unknown or outside of their training distribution inputs, i.e., that carry semantics from a foreign data distribution.

Since its inception, this framework has gradually captured the attention of the scientific community for its applicability in real-world settings. Today, there is a substantial body of work in this field, with even official benchmarks established for testing various proposed OoD detectors. Yet, the literature on OoD detection predominantly focuses on a specific task—supervised image classification—often utilizing image datasets that are not particularly complex.

This Thesis aims to investigate robustness through the expansion of the OoD detection framework. Specifically, it explores other types of models such as Spiking Neural Networks, different tasks like Object Detection, and alternative learning paradigms such as Reinforcement Learning. Through this research, we seek to broaden the scope of the OoD detection framework beyond its traditional applications, whether by exploring uncharted

approaches or by tackling tasks that are closer to practical real-world applications.

This dissertation demonstrates that the OoD detection framework is a highly useful paradigm for Open World Machine Learning, which brings us closer to implementing models capable of learning, recognizing their knowledge gaps, consolidating new knowledge from these gaps, and integrating this knowledge into the model.

Acknowledgements

How treacherous the mind is, making me believe that it wasn't long ago that I embarked on this doctoral journey, when in reality it has already been three years. At that time, I was someone who had just lost their job due to the Covid pandemic, had begun studying programming, and had the wild idea of dedicating myself to something related to artificial intelligence. I was fortunate to cross paths with one of my current co-supervisors, Pablo Garcia-Bringas, who recognized that this young man, with little knowledge in programming and artificial intelligence, could put in the necessary work to change this with the right support. Thank you, Pablo, for that and for the help you've always provided whenever I needed it. And of course, thank you for bringing Javi on board as my other co-supervisor.

Javi has been the guiding light on this path, a person with an amazing ability to generate ideas and find potential solutions to problems that seem unbeatable, always with a smile, always welcoming challenges as new opportunities to pursue science. Thank you for being this way, for supporting me during difficult moments, and for guiding me towards becoming the person I am today.

I cannot forget to thank the Bikaintek Program and its participants. Thanks to Indra and Tecnalia for trusting me with the doctoral program, to Deusto for welcoming me, and to the Basque Government for the program and funding.

I also want to express my gratitude for the unwavering support of my colleagues at Tecnalia and the good times we've shared in day-to-day life: Danel, Bego, Ibai, Adriana, Eric, Iñigo (both), Pedro, Raquel, Andoni, Aritz, Marcos, Jonander, Oier... And especially to Alain, with whom I had the pleasure of collaborating on the synthesis of one of the articles that shaped a chapter of this thesis, and who provided invaluable help. Oh, and I must also thank you all for leaving the server free (Free GPUs!).

I am grateful to my "cuadrilla", Beni 1931, each and every one of them, for always lifting my spirits in difficult times and helping me forget, for a few hours, the overwhelm and stress that often accompany a PhD journey. In general, I appreciate how they have helped me grow as a person; without them, I would not be who I am today.

I also thank my "cuadrilla" from Noja, who have always accompanied me and managed to help me disconnect every summer from the whirlwind of emotions that comes with pursuing a PhD. Special thanks to Eder, who has had to endure many conversations about the doctoral journey from a distance and for helping me get through the final stretch of this path.

I wanted to take a moment to thank those who have accompanied me on this journey and throughout my life, my university friends; my Erasmus colleagues, especially the ones from "el pisito"; and finally, my crossfit mates from the 8 PM class.

Lastly, I want to express my heartfelt gratitude to my family. To my uncle and aunt, my cousins, and especially to my father, my mother, and my brother, who have always offered their help when I needed it,

supported me in every way possible, and stood by me at every moment on this beautiful yet challenging journey. Thank you. And thanks to my grandmother Julia for her invaluable affection and dedication. Finally, to the person who has celebrated every success with me and shared every failure, who has helped me rise when I felt overwhelmed by the PhD, who has always had a smile for me—my life partner, Nerea.

Contents

Abstract	v
Acknowledgements	vii
1 Introduction	1
1.1 Motivation	3
1.2 Outline and Contributions of the Thesis	4
1.3 Reading this Thesis	5
2 Background	7
2.1 Out-of-Distribution Detection	7
2.1.1 Logits-based Post-hoc OoD techniques	11
2.2 Spiking Neural Networks	13
2.3 Object Detection	16
2.4 Fundamentals of Reinforcement Learning	18
2.4.1 Reinforcement Learning Key Concepts	18
2.4.2 Proximal Policy Optimization	24
3 OoD Detection in Image Classification for SNNs	25
3.1 Related Work and Contribution	27
3.1.1 Explainability in Spiking Neural Networks	27
3.1.2 Contribution	28
3.2 Designed Method	28
3.2.1 Spiking Neural Network: Mathematical Notation	29
3.2.2 Spike Count Pattern based Out-of-Distribution De- tection	30
3.2.3 Local Relevance Attribution Method	34
3.3 Experimental Setup	37
3.3.1 Considered SNN Architectures	37
3.3.2 In-Distribution and Out-of-Distribution Data	39
3.3.3 Post-hoc Methods	40
3.3.4 Out-of-Distribution Detection Metrics	41
3.3.5 Evaluating the Quality of the Relevance Attribution Method	42
3.4 Results and Discussion	43
3.4.1 RQ1: Does the SCP-based detector perform competitively when compared to post-hoc OoD de- tectors from the literature adapted to SNN architec- tures?	43
3.4.1.1 Gray-scale Datasets	43

3.4.1.2	Color Datasets	46
3.4.1.3	Neuromorphic Datasets	48
3.4.2	RQ2: How does the SCP-based detector perform when fused with other adapted OoD detection methods?	49
3.4.3	RQ3: Does the local relevance attribution method yield informative explanations about the features that make a given sample be detected as OoD? Which are the limitations of this explainability technique?	51
3.5	Conclusions	55
4	OoD Detection in Object Detection	57
4.1	Related Work and Contribution	58
4.1.1	Out-of-Distribution Detection	58
4.1.2	Open-World Object Detection	58
4.1.3	Contribution	60
4.2	Open-World Object Detection by Feature Map Characterization	61
4.2.1	High-level description	62
4.2.2	Low-level description	63
4.2.3	Supervised Dimensionality Reduction	65
4.2.4	Enhanced Unknown Object Localization	66
4.3	Experimental Setup	67
4.3.1	Evaluation Benchmark	68
4.3.2	Evaluation Metrics	68
4.3.3	Implementation Details	70
4.3.4	Fusion Strategies	71
4.3.5	Post-hoc Methods	72
4.4	Results and Analysis	73
4.4.1	Preamble	73
4.4.2	RQ1: Which is the optimal configuration for our detection method?	74
4.4.3	RQ2: How does our method perform when compared to output-based post-hoc OoD detection methods in single-stage object detectors?	78
4.4.4	RQ3: Does a fusion of feature-based methods with logits-based methods outperform other potential ensemble configurations?	78
4.4.5	RQ4: How does the performance of unknown object detection algorithms implemented on single-stage models compare to that of the SOTA?	80
4.5	Conclusions	82
5	OoD Detection in RL with Sparse Rewards	85
5.1	Related Work and Contribution	86
5.1.1	Sparse Reinforcement Learning	86
5.1.2	Continual Reinforcement Learning and Generalization	87

5.1.3	Out-of-Distribution Detection in Reinforcement Learning	87
5.1.4	Contribution	88
5.2	OoD Detection Techniques and Adaptation Strategies . . .	88
5.2.1	Exploration with Intrinsic Motivation	88
5.2.2	Out-of-Distribution Detection Formulation in Reinforcement Learning	89
5.2.3	Out-of-Distribution Detection Techniques	90
5.2.4	Out-of-Distribution Adaptation Strategies	91
5.3	Experimental Setup	93
5.4	Results and Discussion	95
5.4.1	RQ1: Out-of-Distribution Detection Performance . .	95
5.4.2	RQ2: Adaptation Strategies	98
5.5	Conclusions	100
6	Concluding Remarks	103
6.1	Research Outcomes	106
6.1.1	Other Publications	107
6.2	Future Research Lines	108
A	Results of Ensembles in SNNs	111
A.1	Results of ensembles in color datasets	111
A.2	OoD detection results for SNN with increased depth	113
B	Results on Unknown Object Detection Benchmark	115
	Bibliography	117

List of Figures

1.1	A model deployed in an Open-World facing unknown data.	2
1.2	Block diagram illustrating the structure of the Thesis . . .	6
2.1	Difference between semantic and covariate shift	8
2.2	Detailed mathematical explanation of the OoD framework .	9
2.3	Selection and application of the OoD threshold	13
2.4	Leaky Integrate-and-Fire neuron operation	14
2.5	Schematic architecture of one- and two-stage object detectors with FPN	18
2.6	RL framework	19
2.7	Actor-Critic framework	23
3.1	Workflow of the SCP based detector for an image classification task	29
3.2	Graphical description of the methodology of the SCP based detector	31
3.3	Schematic description of the local attribution method . . .	34
3.4	Datasets for the experiments related to the quality of relevance attribution heatmaps	42
3.5	Critical distance diagrams FC-SNN and CNN-SNN architectures	46
3.6	Bayesian signed test of ODIN and SCP-based detector for FC-SNN and CNN-SNN architectures	47
3.7	Critical distance diagram for color datasets	47
3.8	Bayesian signed tests for ODIN, Energy and SCP-based detector	48
3.9	Fusion of OoD detection methods and voting criterion for their predictions	49
3.10	Critical distance diagram of ensembles, ODIN and Energy evaluated over color datasets	50
3.11	Bayesian signed tests for <i>E-ODIN-SCP</i> ensemble, the <i>E-ODIN-Energy</i> ensemble and the Energy method	50
3.12	Heatmaps showing the relevance attributed to different digits for FC-SNN and CNN-SNN architectures for 1 fully connected layer	53
3.13	Heatmaps showing the relevance attributed to different digits for FC-SNN and CNN-SNN architectures for 2 fully connected layers	54

4.1	Description of the operation of the FMap detector	62
4.2	Enhanced Unknown Localization graphically depicted with an example	67
4.3	Intersection over Union (IoU) explained	69
4.4	Illustration of the AND and OR fusion strategies	71
4.5	Illustration of the SCORE fusion strategy	72
4.6	Pareto front for vanilla FMap method	74
4.7	Pareto front for SDR FMap method	75
4.8	Pareto front for EUL FMap method	76
4.9	Pareto front for non-dominated FMap parameter combinations of the three versions of the algorithm (vanilla, SDR and EUL)	77
4.10	Precision vs Recall plot for non-dominated FMap versions (vanilla, SDR and EUL)	78
4.11	Pareto front for FMap against post-hoc methods	79
4.12	Pareto front for fusion methods	80
4.13	Pareto front for best FMap, post-hoc and fusion methods	81
5.1	OoD detection in RL with our three proposed methods (FwD , AP and $AP-A$)	89
5.2	Three distinct levels illustrating environments that the lava, balls and sand OoD elements	94
5.3	Average AUROC of best performing FwD , AP and Energy versus the L_1 distance between the OoD object and the agent	97
5.4	Average performance returns of various adaptation strategies employed by an agent initially trained on MN5S8 when faced with distinct OoD environments	99
A.1	Bayesian signed tests computed over the paired AUROC scores attained by ODIN and the SCP-based detector with the FC-SNN and the CNN-SNN architecture.	114

List of Tables

3.1	SNN architectures considered in the experiments for gray-scale datasets	39
3.2	SNN architectures considered in the experiments for color datasets	39
3.3	OoD detection scores obtained for the gray-scale FC-SNN architecture	43
3.4	OoD detection scores obtained for the gray-scale CNN-SNN architecture	44
3.5	OoD detection scores obtained for the RGB CNN-SNN architecture	47
3.6	OoD detection scores obtained for the RGB CNN-SNN architecture in neuromorphic datasets	48
3.7	Scores for the SCP-based detector in the experiments devised to address RQ3	51
4.1	Number of images and annotated known and unknown objects per subset in the Unknown Object Detection benchmark	68
4.2	Comparison of best FMap, post-hoc and fusion methods against the SOTA	82
4.3	Hyperparameters for our methods	82
5.1	Test performance in terms of average return and success rate of an agent trained on MN5S8 and a Random agent over different PCG environments	96
5.2	AUROC performance over each OoD environment for different detection techniques	97
A.1	OoD detection scores obtained by the individual methods and the ensembles over the color datasets using a convolutional SNN architecture.	112
A.2	OoD detection scores corresponding to the FC-SNN architecture with $L = 2$ fully connected layers.	113
A.3	OoD detection scores corresponding to the CNN-SNN architecture with $L = 2$ fully connected layers.	113
B.1	Results from best method of RQ1, RQ2 and RQ3	116

List of Abbreviations

General

AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
SL	Supervised Learning
RL	Reinforcement Learning
XAI	eXplainable Artificial Intelligence
NN	Neural Network
CNN	Convolutional Neural Network
SNN	Spiking Neural Network
NIST	National Institute of Standards and Technology
SOTA	State Of The Art
RQ	Research Question
SR	Success Rate
OWOD	Open World Object Detection
UOD	Unknown Object Detection
SDR	Supervised Dimensionality Reduction
EUL	Enhanced Unknown object Localization

Out-of-Distribution detection

ID	In-Distribution
OoD	Out-of-Distribution
MSP	Maximum Softmax Probability

Spiking Neural Networks

STDP	Spike-Time Dependent Plasticity
LI	Leaky Integrator
LIF	Leaky Integrate-and-Fire
SCP	Spike Count Pattern

Object Detection

NMS	Non Maximum Supression
IoU	Intersection over Union
SPPNet	Spatial Pyramid Pooling Networks
RPN	Region Proposal Network
FPN	Feature Pyramid Network

YOLO	Y ou O nly L ook O nce
SSD	S ingle S hot M ultibox D etector

Metrics

TP	T ru e P osi tive
TN	T ru e N ega tive
FP	F al se P osi tive
TPR	T ru e P osi tive R ate
FPR	F al se P osi tive R ate
AUROC	A re a U nder the R OC curve
AUPR	A re a U nder the P re cision-R ecall curve
AP	A ve rage P re cision
mAP	m ea n A ve rage P re cision
A-OSE	A bsol ute O pe n-S e t E rror
AUPR	A re a U nder the P re cision-

Reinforcement Learning

DRL	D eep R ein force ment L e arn ing
MDP	M arkov D e ci sion P ro ce ss
POMDP	P ar ti ally O bservable M arkov D e ci sion
AC	A cto r-C ri ti c
IM	I n tri n si c M oti va ti o n
PCG	P ro ce dural C o n t e nt G e ne ra ti o n
PPO	P ro xi mal P o li cy O pti mi za ti o n
RIDE	R e wa r di ng I mpact D ri ve n
	E xploration
GAE	G e ne re ali zed A dva n tage E sti ma tor
SSL	S e lf S up e rvised L oss

Chapter 1

Introduction

The rapid advancements and widespread adoption of Artificial Intelligence (AI) systems for real world applications have underscored the urgent need for these models to be trustworthy. But what trustworthiness means for AI is a matter of present concern and has become a highly debated topic in the recent years, attracting significant interest from scientists, institutions, businesses, and governments alike. A major step in elucidating it was the publication of the “Ethical Guidelines for Trustworthy AI” [1] in 2019 by the European Union, an actor with relevance in the matter. The text outlines seven key requirements for trustworthy AI and identify three essential pillars for fulfilling these requirements. The robustness of an AI system is recognized as a crucial element, since one of the seven requirements is the “*technical robustness and safety*” and the robustness itself is underlined as one of the three pillars.

As evidenced, robustness is closely related to safety. In fact, AI safety can be defined as an interdisciplinary field concerned with preventing accidents, misuse, or other harmful consequences that may result from AI systems. It encompasses AI alignment (including machine ethics), robustness and resilience, which need monitoring and inspection to ensure them [2]. In other words, the robustness of a model contributes to greater safety and, therefore, better trustworthiness by the users (*audience*) of the AI-based system. The current challenge, as highlighted by many influential authors in this field his 2024 work [3], is the lack of sufficient research in this area.

In summary, the scientific community is not paying enough attention to the safety and robustness of AI models, the latter being vital for deploying models reliably in the open-world. But what defines a robust model? According to the European Commission, a robust AI model is one that “*does not cause any unintentional harm, working in a safe and reliable fashion from both technical (performance, confidence) and social (usage, context) perspectives*”. Focusing more on the technical aspect of robustness, which is the primary concern of this Thesis, we refer to the definition [4] provided by the National Institute of Standards and Technology (NIST) in the United States AI Risk Management Framework:

“Robustness or generalizability is defined as the ability of a system to maintain its level of performance under a variety of circumstances. [...] Robustness requires not only that the system perform exactly as it does

under expected uses, but also that it should perform in ways that minimize potential harms to people if it is operating in an unexpected setting.”

Therefore, a robust AI model must be capable of withstanding malicious actors attempting to alter its behavior and be able to handle situations where the initial data distribution changes during operation. It can be caused due to non-stationarities in the modeled concepts (concept drift [5]) or the arrival of new unknown classes that may not necessarily be outliers to the model’s training. Ultimately, this means having a model prepared to effectively face the open-world machine learning paradigm, which should follow the following pipeline:

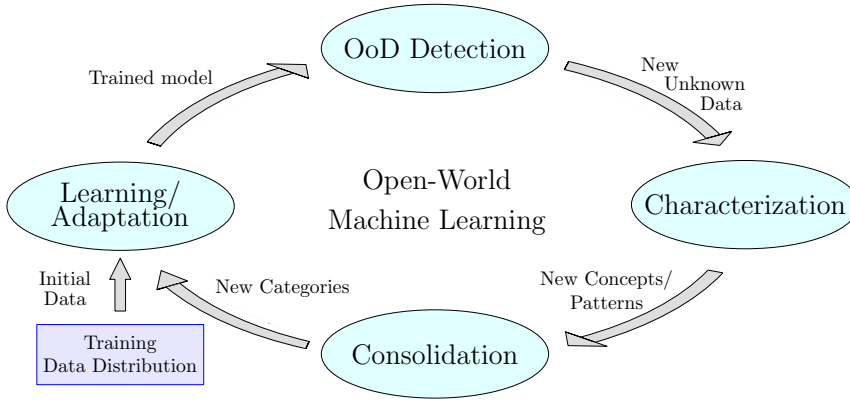


FIGURE 1.1: A model deployed in an Open-World facing unknown data.

These different components are currently being addressed in various ways:

- **Out-of-Distribution (OoD) Detection.** The model must first be capable of detecting the presence of new elements or dynamics. In this context, open set detection, novelty detection, or, more generally, Out-of-Distribution detection are responsible for this task.
- **Characterization.** Departing from a number of different unknown elements, they must be grouped into several concepts by identifying instances with similar features.
- **Consolidation.** The next step involves consolidating the new concepts into new knowledge, i.e., being able to differentiate groups or classes among the new concepts. Unsupervised clustering algorithms or topological studies can be particularly useful in this regard.
- **Learning/adaptation.** Finally, the newly detected and correctly identified and consolidated knowledge must be incorporated into the model. Techniques from the paradigm of class incremental learning [6] can be especially beneficial for this purpose, as well as data stream learning [7] techniques for the case where data comes in a continuous flow.

1.1 Motivation

In this Thesis, our objective is to advance the detection component of the presented workflow, specifically in OoD detection. Since the inception of OoD detection in 2016 with Dan Hendrycks' work [8], this field has garnered considerable attention from the scientific community.

OoD detection aims to develop AI models capable of recognizing when they are being queried to predict an example that does not belong to the learned knowledge during training, i.e., a sample outside the statistical distribution of the training data. Simultaneously, these models must maintain acceptable performance on known data.

Since its inception, the OoD detection community has predominantly focused on image classification due to the ease of implementation of image classification models and the simplicity of obtaining OoD data for experiments. In this context, it is relatively straightforward to find datasets for training a model on a specific domain, such as classifying animal species, and then test the model on samples outside its training distribution, such as images of vehicles. Without an OoD detector implemented, the model would misclassify each vehicle image as one of the learned animal species. The desired behavior, however, is for the model to recognize that the query falls outside its knowledge and to alert the human accordingly.

Therefore, most OoD detectors have been designed for the Supervised Learning (SL) task of image classification, with the preferred option for models being the traditional Neural Networks (NNs). Despite recent shifts in focus towards other possibilities, we believe there is insufficient research in new areas. Thus, we envision the need to extend OoD detectors towards:

- **New models.** Lately, the research community has increasingly focused on highly efficient bio-inspired models, termed Spiking Neural Networks (SNNs). The main reason for this interest is the computational efficiency offered by these networks, making them particularly attractive for the future, where the massive deployment of AI systems will make the efficiency of them a significant concern. Moreover, their low power consumption makes them especially suitable for use in embedded AI systems. Nonetheless, unlike for traditional NNs, there is not yet any research done for SNNs in the field of OoD detection.
- **Unexplored tasks and learning scenarios.** While image classification is a useful task, it is not so commonly implemented in real-world systems. Object detection, a closely related task, is widely used in numerous applications, such as robotics and autonomous driving. This task presents new challenges due to the increased complexity of the models and the difficulty of obtaining OoD or unknown data, which the community is just beginning to address. In terms of other learning scenarios, Reinforcement Learning (RL) has received minimal attention in the OoD community, despite its potential to be revolutionary in the near future.

In light of the aforementioned objectives, this Thesis aims to expand the boundaries of the OoD detection paradigm, covering a type of models

not covered yet, the SNNs, and tasks or learning scenarios that are not tackled enough in the literature, namely, object detection and RL.

1.2 Outline and Contributions of the Thesis

In accordance with the research objectives stated above, the Thesis is structured in chapters, each addressing one specific goal: OoD detection in SNN, OoD in object detection tasks, and OoD in RL scenarios. A brief summary of each of these chapters is presented below.

Chapter 2: Background

Before proceeding with the original technical contributions of this Thesis, this second chapter aims to introduce and condense all the needed information to understand them thoroughly. First, the problem of the OoD detection is defined in a mathematical fashion, together with the methods that will be used for the benchmarks in the different chapters. At the same time, an extensive introduction on the operation of the SNNs is developed, given that they are not mainstream neural networks and their inner workings being considerably different to the traditional ones. Moreover, knowing how they operate will be important to understand the technique developed for them. In addition, object detection framework is explained and a brief summary of the evolution of the field is provided, alongside the explanation of the workings of one- and two-stage object detectors. Finally, the RL learning paradigm or scenario is introduced, together with the mathematical notation that will be used for the corresponding technical contribution.

Chapter 3: Out-of-Distribution Detection in Image Classification for Spiking Neural Networks

In this chapter we introduce a pioneering method for OoD detection tailored to the previously unexplored realm of SNN models. This approach capitalizes on the internal mechanisms of SNNs. The technique is evaluated against and integrated with post-hoc OoD detection methods, which are detailed in Chapter 2. We assess its performance across various types of image classification datasets, including black and white, colored, and neuromorphic datasets. Additionally, we enhance our detection method with the capability to provide explanations for its detections. This feature not only augments the model’s transparency, but also significantly boosts its overall trustworthiness in a current European regulatory context where robustness is a pillar for AI trustworthiness [9].

Chapter 4: Out-of-Distribution Detection in Object Detection

Throughout this chapter we present an study on the performance of the single-stage object detectors for the unknown object detection. We design

our own OoD detector for this models with the premise of not entailing re-training and test it together with the post-hoc OoD methods to obtain the optimal configurations. Driven by the conclusions of the previous chapter, we fuse our method with the post-hoc methods and finally compare all against the state-of-the-art (SOTA) techniques, mainly implemented in two-stage object detectors with the need of retraining the model.

Chapter 5: Out-of-Distribution Detection in Reinforcement Learning with Sparse Rewards

This chapter delves into the design and leverage of the OoD detection framework in the RL scenario, specifically for sparse reward environments. Our aim is double in this technical contribution. On the one hand, we define some OoD detection methods exploiting the RL agent’s architecture and the modules (e.g., intrinsic motivation modules) typically required to solve this type of environments. Subsequently, after the detection, we study the possible adaptation strategies that could be utilized, regarding to the knowledge transmission of the agent to the new scenario. We also test if adding new modules to elicit the learning of the representations could be beneficial.

Chapter 6: Concluding Remarks

Finally, Chapter 6 encapsulates the key insights garnered from several years of research. It also lists contributions made to conferences and journals. Additionally, this chapter outlines future research directions that are consistent with the thematic interests of this Thesis.

1.3 Reading this Thesis

The chapters of this Thesis do not necessarily need to be read in sequence. While the organization follows a logical progression, readers already versed in the various models, tasks, and learning scenarios may prefer to start with the experimental chapters (3, 4, 5). They can consult the background chapter (2) as needed, especially if the related work discussed within the chapters does not provide enough context to fully grasp the contributions.

Chapter 6 concludes with a summary of the findings and possible extensions based on the outcomes from each chapter. It also outlines potential avenues for future research.

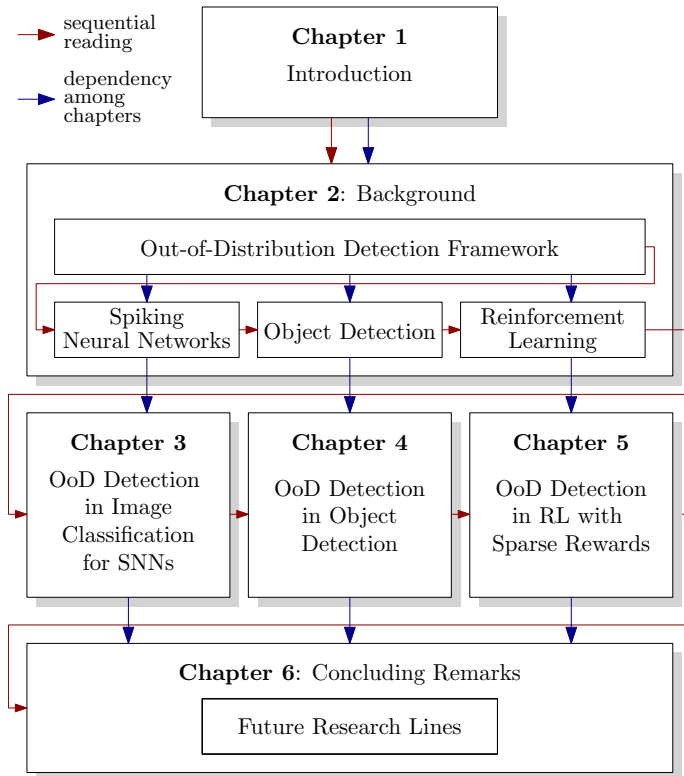


FIGURE 1.2: Block diagram of the structure and reading flow of the Thesis.

Chapter 2

Background

In this section, an extensive presentation of the concepts fundamental for this Thesis provided. We start defining the guiding thread of all the chapters, the OoD detection framework in Section 2.1. We include a detailed description of the various approaches that will be used throughout the Thesis in Subsection 2.1.1. Subsequently, the specific contextualization for each technical contribution chapter is given. For Chapter 3, Section 2.2 undertakes an extensive introduction of the SNNs operation. In Section 2.3, the essential concepts of the object detection tasks are explained alongside a concise review of the history of the models developed, to finally delve deeper into the inner workings of the single- and two-stage models. To conclude, Section 2.4 presents a comprehensive overview on the RL learning paradigm and the techniques needed to solve the environments used for the experiments in Chapter 5.

2.1 Out-of-Distribution Detection

Before formulating the problem in a mathematical fashion, it is important to clearly define what the OoD detection encompasses and to establish the similarities and differences with other related tasks, such as outlier, anomaly and novelty detection. This is precisely the central matter tackled in the comprehensive survey recently contributed in [10], which also points out several misconceptions existing in the literature. This work distinguishes between two different types of shift in data: the *semantic shift* and the *covariate shift*. The latter stands for the situation where features of the input data vary without any change in the class they belong to. Stated differently, the statistical dependence between variables change, but not their relationship with the output variable. In contrast, semantic shift refers to the case when the new data belongs to a class different than the ones learned by the model during its training stage. This being said, it is important to note that OoD detection addresses only semantic shift.

To clarify the application domain in which OoD detection can be formulated, we start by differentiating it from the above-mentioned tasks. The most distinct task is outlier detection, as it does not follow the classic train-test schema. Instead, a set of observations is provided, and the task is to detect significantly different samples from this contaminated dataset

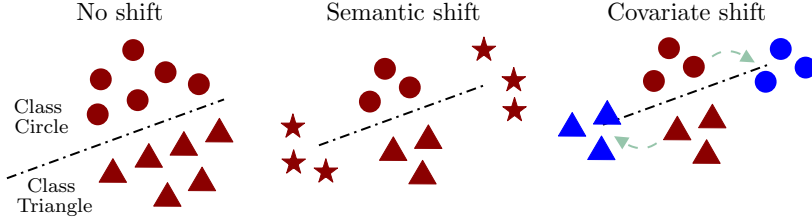


FIGURE 2.1: Difference between semantic and covariate shift. The semantic one occurs when a new class (star) emerges. In the covariate shift, the features of the input change (the color of the figures, from red to blue) while maintaining the semantics (the label), what makes the model incorrectly classify some instances.

of observations. Both semantic and covariate shifts can occur in this scenario. Conversely to outlier detection, OoD detection is assumed to occur under a train-test schema. As concluded in [10], OoD detection can be regarded as a super-category that includes semantic anomaly detection, multi-class novelty detection and open set recognition [11]. Summarizing, in OoD detection a model is learned from a training set that is sampled from a given ID dataset, whereas test samples are drawn evenly from either the in-distribution or from a semantically different distribution, namely, the aforementioned OoD dataset. The objective is to correctly classify test samples as belonging to the ID or OoD distributions, while maintaining the performance of the model in the original classification task. In almost all cases, detection is ultimately achieved by crafting a scoring function that assigns different scores for the in- and out-distribution samples.

Mathematically, the OoD detection task aims to distinguish whether a query instance $\mathbf{x}_{query}$ belongs to the data distribution of the data used to train the model, or alternatively can be thought to belong to another different distribution. Without loss of generality, let the *in-distribution* dataset be denoted as $\mathcal{D}_{tr}$, which is drawn from a probability distribution $P_{\mathbf{X}}(\mathbf{x})$ defined over a space $\mathcal{X} \in \mathbb{R}^K$. This dataset is used to learn a classifier $F_{\theta}(\mathbf{x})$ by adjusting the model's parameters θ so as to maximize its generalization capability over a set of labels or classes $\{1, \dots, C\}$. Once the model has been learned, the classifier can infer the class $\hat{y} = F_{\theta}(\mathbf{x}) \in \{1, \dots, C\}$ of any new sample $\mathbf{x} \in \mathbb{R}^K$. We further assume another distribution $Q_{\mathbf{X}}(\mathbf{x})$, the out-distribution, which is different than $P_{\mathbf{X}}(\mathbf{x})$ but defined over the same space $\mathcal{X}$. New query examples can be sampled from a mixture distribution $M_{\mathbf{X}|Z}(\mathbf{x}|z)$ such that $M_{\mathbf{X}|Z}(\mathbf{x}|z)$ is equal to $P_{\mathbf{X}}(\mathbf{x})$ if $z = 1$, and $Q_{\mathbf{X}}(\mathbf{x})$ if $z = 0$. If Z follows a uniform binary probability distribution, an OoD detector $G(\mathbf{x})$ aims to distinguish which query instances $\mathbf{x}_{query}$ follow the in-distribution from the ones which not, without observing variable Z , and by only resorting to the trained model $F_{\theta}(\mathbf{x})$. This explanation is graphically presented in Figure 2.2.

Recalling to [10] and following its proposed taxonomy, OoD detection methods can be divided in three types: classification-based, density-based or distance-based. The techniques that are designed in this Thesis for the

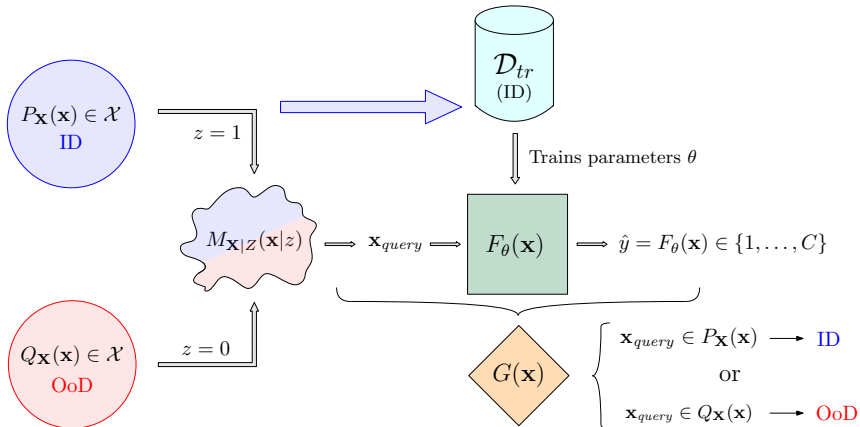


FIGURE 2.2: Detailed mathematical explanation of the OoD framework.

different uses cases fall within the latter category. Nevertheless, we herein focus on the literature related to some of the classification-based methods, specifically those which can be labeled as *post-hoc* methods. These detectors can be adapted to the mainstream neural networks with no to minor changes to their cores.

The first paper coining the *Out-of-Distribution detection* term was actually a classification-based method, presented in [8] and often referred to as the *baseline* method. This approach utilizes the so-called Maximum Softmax Probability (MSP) to detect OoD samples, based on the observation that samples belonging to the ID dataset tend to have a higher MSP value than OoD instances. By simply defining a threshold on this score, their method achieved acceptable results in a wide variety of datasets. The authors in [12] went one step further by applying a temperature scaling strategy to the softmax computation, which pushes softmax scores of in- and out-distribution samples further apart from each other. In addition, they included input preprocessing by adding small gradient and softmax dependent perturbations to the data, slightly improving the performance. They named their method ODIN. More recently, the work in [13] assumed that the feature space of the penultimate layer of a classifier follows a Gaussian distribution, allowing for the estimation of mean and variance statistics from the features of every class. A posterior fit of a multivariate Gaussian distribution and the usage of Mahalanobis distance to gauge the closest class-conditional distribution is used for OoD detection. Following the strategy posed in [12], the performance is improved by adding small noise perturbations; this time depending on the Mahalanobis distance rather than softmax.

Contrarily to previous approaches, in [14] a technique called Outlier Exposure is presented for improving the performance of existing detectors. Authors propose to leverage the enormous quantity of data available nowadays by modifying the training process of neural networks with the addition of an additional term to the original loss function. This term

depends on the original task (classification, density estimation, etc.) and on the detector in use. This term helps the model to learn heuristics that will improve the detector’s performance. Later, an energy-based detector was derived in [15] by adapting the concept of *Helmholtz free-energy* to deep NNs, expressing it in terms of the denominator of the softmax activation. Therefore, each sample is assigned a score based on its logits (its *energy*), which is shown to be higher for OoD samples than for ID instances. This was proven to yield even better detection performance than previous softmax-based scores.

Thereupon, the study presented in [16] improved the computational efficiency at inference time of the energy-based detector by training multiple intermediate classifiers operating at different depths of the trained neural network. The intermediate activations of the neural networks are extracted at certain depths of its internal layered structure depending on its complexity, leveraging the intuition that less complex OoD samples can be detected by using low-level activation statistics of the network. Extending the success of the energy-based detectors, authors in [17] came up with a new scoring function (*JointEnergy*), which exploits the joint uncertainty across labels by aggregating the label-wise free energy for multiple labels.

In contrast to presented methods, *GradNorm* [18] explores the model’s parameters and the gradient space to compute an OoD score. Specifically, GradNorm backpropagates the Kullback-Leibler (KL) divergence between the softmax output and the uniform distribution, computing the OoD score by extracting the gradients of the last fully connected layer and calculating its L1 norm. Subsequently, [19] proposed a simple post-hoc method forged as *Rectified Activations* (ReAct), which is applied to the penultimate layer of a network to truncate the activations at inference time. Its effectiveness is supported by the finding that applying the BatchNorm statistics of ID training data to the OoD data at inference time leads to abnormally high activations in all layers, and therefore to the model’s output, deteriorating the detection performance. Basically, ReAct limits the activation’s value, yielding better detection scores than the previously presented techniques hinging on the logits of the network. Another interesting detector is *RankFeat* [20], which relies on subtracting the rank-1 matrix from the high-level feature, and using logits derived from this perturbation to compute the OoD score. In their work, authors demonstrate that in-distribution samples are less affected by this perturbation, enabling the detection of OoD samples in the logit space.

More recently, in [21] a novel hyperparameter-free and computationally efficient OoD detection method is designed by leveraging Modern Hopfield Networks. The approach utilizes the penultimate layer outputs of a neural network as representations of ID data, which are then stored as patterns. These stored patterns serve as anchors to measure the discrepancy of unseen data, facilitating OoD detection. The authors derive a simplified version of the Hopfield energy function, termed SHE (Simplified Hopfield Energy), which uses a single stored pattern per class, obtained by averaging the penultimate layer outputs of training samples within each class. Authors in [22] introduce a novel, straightforward method for OoD detection

called Activation Shaping (ASH). ASH is a post-hoc, on-the-fly technique that prunes a significant fraction (e.g., 90%) of a sample’s activations at a late layer of a neural network, simplifying or lightly adjusting the remaining activations. Similar to this, the work in [23] focuses on activation shaping. Specifically, their analysis reveals that while scaling significantly enhances OoD detection, pruning has a detrimental effect. Building on this insight, they introduce SCALE, a simplified post-hoc enhancement method that achieves state-of-the-art OoD detection without compromising ID accuracy. Additionally, they propose Intermediate Tensor SHaping (ISH), a training-time method that incorporates scaling concepts to improve OoD detection performance. ISH demonstrates substantial improvements, emphasizing the critical role of activation distributions in distinguishing ID data.

It is also important to highlight that, in response to the lack of standardized benchmarks—which led to unfair comparisons and inconclusive results—some authors introduced OpenOOD [24], a well-structured code-base that implements over 30 methods from the aforementioned related fields such as anomaly detection and open set recognition. OpenOOD provides a comprehensive benchmark under the generalized OOD detection framework for image classification tasks, facilitating rigorous and fair evaluations of OOD detection methods. Building on this foundation, OpenOOD v1.5 [25] introduces significant improvements to ensure accurate, standardized, and user-friendly evaluations. It extends capabilities to large-scale datasets like ImageNet and explores full-spectrum OOD detection, which considers both semantic and covariate distribution shifts. New features include an online leaderboard and an easy-to-use evaluator.

In conclusion, the OoD detection is an important framework that has received significant attention since its inception. Nevertheless, the majority of research has predominantly focused on the image classification task within traditional neural networks, as demonstrated by the presented works and the OpenOOD benchmark.

2.1.1 Logits-based Post-hoc OoD techniques

To evaluate an OoD detection method on a specific model, compatibility between the method and the model is essential. The technical contributions of this Thesis aim to extend the OoD detection framework to tasks and models that have been less explored. Consequently, we must select methods that are compatible with these applications, hence focusing on post-hoc detection methods. However, adapting some widely acknowledged post-hoc detection methods to the models and tasks discussed in subsequent chapters—such as spiking neural networks (Chapter 3) and the object detection task (Chapter 4)—proves challenging or even unfeasible.

Therefore, we will employ post-hoc techniques that are completely model-agnostic, being that the model is any neural architecture under choice for any classification task. In fact, these methods depend solely on the logits from the model, i.e., the activations of the final layer before the

application of the Softmax function, which is typically used to generate a probability distribution over the established categories.

The most widely used logits-based OoD detection methods include MSP [8], ODIN [12] and Energy [15]. These methods derive a score for each input sample $\mathbf{x}$ by computing a function $f_{OoD}(\cdot)$ using the logits obtained from the model's F_θ forward pass. The MSP method calculates the maximum softmax probability as the score. ODIN enhances separation between OoD and ID samples by incorporating a temperature scaling T into the softmax calculation. Meanwhile, the Energy method defines the free energy function in terms of the denominator of the softmax activation. To align with the conventional definition where positive (in-distribution) samples have higher scores, the negative of the free energy function is used. The equations for these methods are as follows:

$$\text{MSP} : f_{OoD}(\mathbf{x}) = \max_i \frac{\exp(F_{\theta i}(\mathbf{x}))}{\sum_{i=1}^C \exp(F_{\theta i}(\mathbf{x}))}, \quad (2.1)$$

$$\text{ODIN} : f_{OoD}(\mathbf{x}) = \max_i \frac{\exp(F_{\theta i}(\mathbf{x})/T)}{\sum_{i=1}^C \exp(F_{\theta i}(\mathbf{x})/T)}, \quad (2.2)$$

$$\text{Energy} : f_{OoD}(\mathbf{x}) = -T \cdot \log \sum_{i=1}^C \exp(F_{\theta i}(\mathbf{x})/T), \quad (2.3)$$

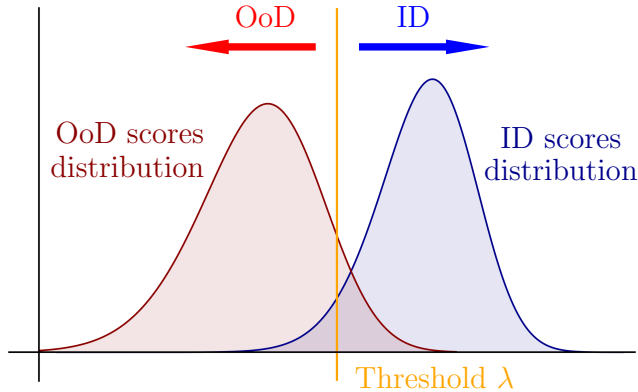
where $F_{\theta i}(\mathbf{x})$ refers to the logit value of the i^{th} class or category, obtained by processing the input instance $\mathbf{x}$ utilizing the model F_θ . The temperature T utilized in ODIN varies and is specified in each respective chapter, whereas for the Energy method, the temperature is consistently set to 1, in accordance with recommendations from the original study.

A threshold λ is subsequently imposed on the score to discriminate between the ID and OoD samples, as:

$$G(\mathbf{x}) = \begin{cases} 1 \text{ (ID)} & \text{if } f_{OoD}(\mathbf{x}) > \lambda, \\ 0 \text{ (OoD)} & \text{if } f_{OoD}(\mathbf{x}) \leq \lambda. \end{cases} \quad (2.4)$$

Typically, this threshold is set to ensure a 95% True Positive Rate (TPR), i.e., the proportion of in-distribution images correctly classified as in-distribution. The application and effects of this threshold are visually illustrated in Figure 2.3.

It is of utmost importance to clarify that while ODIN also incorporates a perturbation technique to enhance OoD detection, this approach is not readily adaptable for SNNs or object detection models discussed in this Thesis. Consequently, this aspect will not be implemented in these cases, and further details are omitted.

FIGURE 2.3: Selection and application of the OoD threshold λ .

2.2 Spiking Neural Networks

Often referred to as *the third generation of artificial neural networks*, this family of neural networks has garnered significant attention since their inception. Not only because they can represent more faithfully the inner stimuli of the human brain, but also due to the technical advantages that their particular information processing strategy entails in terms of spatio-temporal pattern learning and energy efficiency. In the brain, information flowing between neurons is conveyed through synapses by trains of short electric pulses called *spikes*, so that neurons receive and accumulate them as potential in their membrane. When a potential threshold is met, the neuron emits an pulse (spike) of its own to the subsequent neuron. The neuron emitting a spike is the pre-synaptic neuron, whereas the receiver neuron is referred to as post-synaptic neuron. Mimicking this dynamic behavior offers an inherent way to deal with temporal data, and the sparsity and binary nature of spikes makes it possible to reduce dramatically the energy consumption of these models when deployed on specialized hardware, as chips like IBM’s TrueNorth [26], SpiNNaker Chip [27] and Intel’s Loihi [28].

A major drawback of SNNs is the fact that spiking neurons are not differentiable due to the discontinuity of the spike. Therefore, gradient backpropagation-based learning schemes are not directly applicable. To overcome this issue, plenty of learning methods have been proposed over the years, ranging from new bio-inspired learning rules to adaptations of the gradient descent algorithm to deal with the aforementioned discontinuity, among others. To top it all off, each learning method comes with its own limitations in terms of input encoding scheme, neuron models, network architectures and network running mode, among others [29]. In this subsection we briefly revisit them as a referential knowledge base for concepts later referred to in the remainder of the article.

Encoding methods for SNNs can be divided into two big groups: rate encoding schemes and temporal encoding schemes [30]. In both cases, their purpose is to transform input samples into spike trains that can be

processed by the network. As such, rate encoding schemes embeds input data information in the instantaneous or averaged rate of generated spikes in a single or group of neurons, i.e., in the count of spikes emitted within a time encoding window. On the other hand, temporal encoding methods conveys the input information in the precise timing of the spikes.

As for the neuron models, a great variety exists, each with its own biological plausibility and processing efficiency. Basically, they describe the evolution of the membrane potential of the neuron and the spike generation at different levels of detail. The Leaky Integrate-and-Fire (LIF) model, for instance, is the most implemented model in the SNN community due to its low computational cost and simplicity when describing neuron dynamics [31]. Basically, the neuron integrates the input spikes as currents applied to a resistor–capacitor (RC) circuit that increases its voltage until a threshold is exceeded. At this point in time, a spike is emitted and the voltage of the neuron is reset to the resting value. Sometimes it is also modeled with the so called refractory period, a interval of time after the reset in which the neuron can not generate any output spike. This whole process is illustrated in Figure 2.4. The LIF neuron abstracts away the shape and profile of the output spike: it is simply treated as a discrete event. As a result, information is not stored within the spike, but rather the timing (or frequency) of spikes. The Spike-Response Model (SRM) behaves similarly to the LIF model, with the main difference being that it has an adjustable threshold that depends on the time since the last post-synaptic spike occurred. While LIF models are usually defined in terms of differential equations, the SRM expresses the membrane potential at a certain time as an integral over the past [32]. More biologically plausible models include the Hodgkin-Huxley or the Izhikevich models [33], among others.

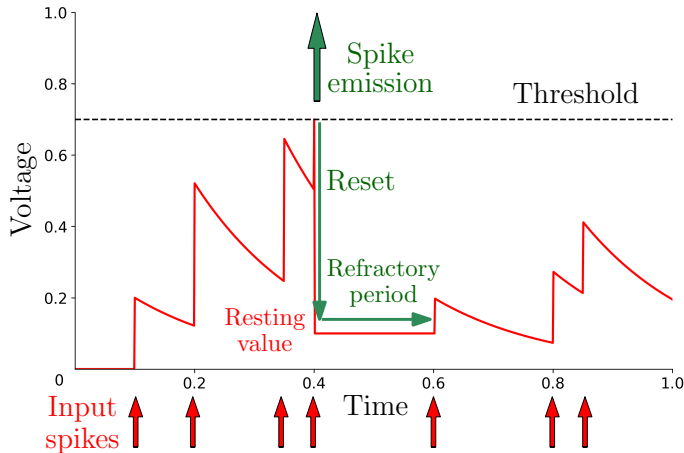


FIGURE 2.4: Leaky Integrate-and-Fire (LIF) neuron operation.

As in traditional neural networks, SNNs can run in an offline or on-line fashion. SNNs have shown their ability to build spike-time learning

rules that capture temporal associations by leveraging spike information representation. Therefore, not only they are well suited for learning from spatio-temporal data in a batch learning setting, but they are also inherently effective for handling these temporal associations in streaming data [34].

After the above short primer on concepts related to SNNs, we proceed by describing several learning algorithms relevant to our study. It is not the aim of this research to comprehensively review all existing learning methods for SNNs. Therefore, some few insights about the most relevant approaches for this purpose are next provided. To begin with, the Spike-Time Dependent Plasticity (STDP) [35] is a strongly bio-inspired unsupervised learning rule that strengthens the connection between pre- and post-synaptic neurons that fire one after the other and that has inspired many other learning algorithms. SpikeProp [36] was the first to train SNNs by backpropagating errors, done on a single hidden layer architecture, with the SRM neuron model and the limitation that neurons were limited to emitting only one spike. Works thereafter refined this method to enable multiple spike firing at desirable times [37]. The main idea of the Spike Pattern Association Neuron (SPAN) [38] is to transform the input and the desired output spike trains into analog signals by convolving the spikes with a kernel function. In this manner, the computation of error signals is simplified, allowing for the application of gradient descent to optimize the synaptic weights.

However, learning methods discussed so far have difficulties to effectively train deep (multi-layered) SNNs. Surrogate Gradient (SG) methods overcome the discontinuity of the spiking neuron model by substituting it with a smooth continuous relaxation. Therefore, it does not impose any restrictions in the learning algorithm that can be used to train the model. Consequently, SG methods can be combined with Back-Propagation Through Time (BPTT) to give rise to recurrent flavors of SNNs. Nevertheless, this workaround poses stringent limitations for SNNs in terms of computational and memory resources when deployed over traditional hardware (GPUs). Moreover, its implementation over neuromorphic hardware may impose locality requirements that can complicate its deployability [39]. A remarkable method that solves the latter and utilizes a surrogate gradient approach is *SuperSpike*, a biologically plausible nonlinear Hebbian three-factor rule with individual synaptic eligibility traces [40]. Basically, the objective of the algorithm is to minimize the van Rossum distance between the desired and the actual spike train through gradient descent. Hence, the derivatives of the spikes with respect to the weights of the network are needed. To make this computation feasible, the spike train is approximated by replacing it with a continuous auxiliary function that depends on the difference between the membrane potential of the LIF neuron and its firing threshold.

The mathematical description and notation of an SNN is provided within the Chapter 3 (Section 3.2.1) as it is essential to fully grasp the operation of the OoD detector designed in the chapter.

2.3 Object Detection

To better understand the technical contributions of Chapter 4, a brief contextualization of the object detection task will be provided. We will take a concise journey through the history of this task, focusing on the technical milestones that have shaped the highly capable models we have today and expanding on the differences between one- or single-stage and two-stage object detection models.

Object detection stands as a critical task in computer vision, focusing on identifying instances of specific visual categories such as humans, animals, or vehicles within digital images. This task forms the foundation for numerous advanced computer vision applications, including instance segmentation, image captioning, and object tracking. The swift advancement of DL methodologies—since the rebirth of Convolutional Neural Networks (CNNs)—has significantly accelerated progress in object detection, leading to groundbreaking achievements and positioning it as a focal point of intense research activity [41]. We now detail the two main types of object detector architectures.

Two-stage object detectors. Initiated by [42] in 2014 with the Regions with CNN features (R-CNN), the landscape of object detection began to transform. This methodology introduced a “coarse-to-fine” approach, starting with the extraction of object proposals using selective search algorithm, followed by feature extraction through CNNs pre-trained on ImageNet, and finally, the classification of these features using linear support vector machines. Building on the foundations of R-CNN, [43] proposed the Spatial Pyramid Pooling Networks (SPPNet). The key innovation was the spatial pyramid pooling layer that allowed CNNs to process images of varying sizes without needing to resize them. Subsequently, [44] in 2015 improved upon R-CNN and SPPNet with Fast R-CNN, which integrated the training of the detector and bounding box regressor within a single DL network. Despite improvements in the detection process and the speed and efficiency of the model, it still was limited by the proposal generation process. Therefore, later that year [45] introduced Faster R-CNN, which incorporated a Region Proposal Network (RPN) that generated region proposals with a CNN, substituting the costly selective search algorithm and enabling near-real-time detection. This model unified most components of an object detection system, including proposal detection, feature extraction, and bounding box regression, into a cohesive end-to-end DL framework.

One-stage object detectors. Also in 2015, You Only Look Once (YOLO) model was proposed by [46] and marked the advent of one-stage detectors in the realm of DL. YOLO is distinguished by its remarkable speed; a streamlined version operates at 155 frames per second (fps), while a more robust variant performs at 45 fps. Unlike two-stage detectors, YOLO employs a single neural network that processes the entire image at once, dividing the image into regions and concurrently predicting bounding boxes and probabilities for each region. Despite its significant enhancements in speed, YOLO experiences reduced localization accuracy

compared to two-stage detectors, particularly in detecting smaller objects. Subsequent versions of YOLO object detector have been developed tackling these problems and improving overall efficiency and accuracy by incorporating new elements proposed in the object detection community [47]. The most recent version is YOLOv10 by [48]. Another remarkable one-stage detectors are Single-Shot Multibox Detector (SSD) [49], RetinaNet [50], CenterNet [51] and DETR [52]. The latter is a transformer based architecture, a trend that has gained attention in recent years.

One big leap on performance was achieved in 2017 by [53] with the introduction of the Feature Pyramid Network (FPN). Prior to the introduction of FPN, many DL-based detectors relied solely on the feature maps from the topmost layers of the network for detection tasks. While deeper layers in a CNN offer rich features beneficial for classifying categories, they are less effective for object localization. To address this, FPN introduced a top-down architecture complemented by lateral connections, which enables the construction of high-level semantic features at various scales. Leveraging the natural formation of a feature pyramid through CNN forward propagation, FPN significantly enhances the detection of objects across a broad range of scales.

The FPN was first introduced for the Faster R-CNN architecture, and then a similar concept was included for one-stage models like YOLO. The key idea is the same, but with differences in the implementation. The most relevant dissimilarity for the purpose of the Chapter 4 is the output size of FPN. When combined with two-stage architectures, the final features coming out from the different levels or scales of the FPN have the number of channels fixed to be the same, since all levels of the pyramid use shared classifiers/regressors (the so called detection heads). In contrast, in the adaptation for YOLO architectures the channels do not need to be fixed as each level has its own detection head.

Another two essential concepts or mechanisms of the object detectors are the confidence threshold and the Non-Maximum Supression (NMS). The former, is set on the prediction confidence p_c , which indicates the likelihood of an object's presence within the bounding box. This threshold is established to eliminate predictions with low confidence. The latter is a post-processing technique that refines results by removing redundant overlapping bounding boxes generated around the same object with varying confidence scores, i.e., filter outs the redundant irrelevant predictions.

To summarize, we illustrate in Figure 2.5 the presented concepts and the two main types of object detector architectures. Following the detection heads, each detection is represented by a vector that includes the confidence on the detection p_c , bounding box coordinates $\{b_1, b_2, b_3, b_4\}$ and class probabilities $\{p_{c_1}, p_{c_2}, \dots, p_{c_n}\}$ for that box. By using the mentioned threshold on the confidence and the NMS, the final predictions $\hat{\mathbf{y}}$ are obtained. They are composed by the bounding box coordinates $\hat{\mathbf{b}} = \{b_1, b_2, b_3, b_4\} \in \mathbb{R}^4$ and the predicted category $\hat{c}$, i.e., they are defined as $\hat{\mathbf{y}} = \{\hat{\mathbf{b}}, \hat{c}\}$.

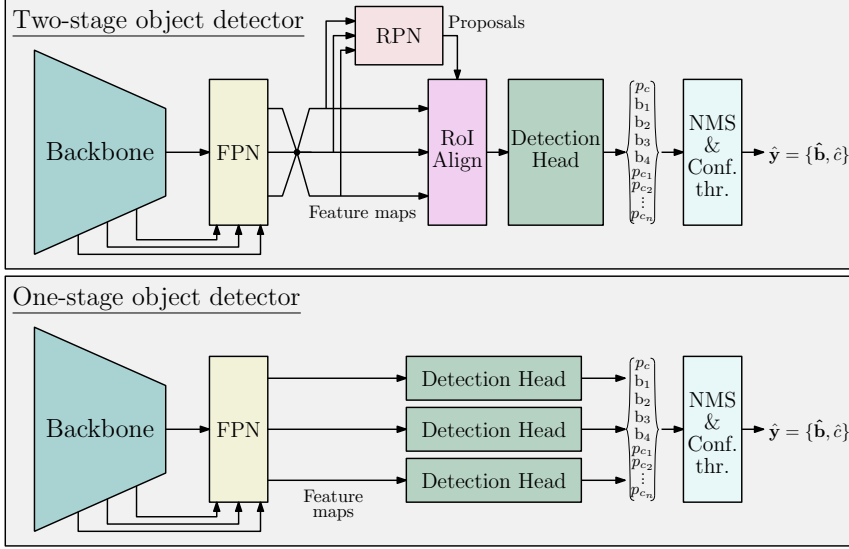


FIGURE 2.5: Schematic architecture of one- and two-stage object detectors with FPN. In the two-stage model, the union of the arrows of the feature maps symbolizes that they all share the same number of channels, not that there is an aggregation between them.

2.4 Fundamentals of Reinforcement Learning

Given that this Thesis focuses on OoD detection, this chapter provides a brief overview of the field of RL, describing some key mathematical formulas and concepts and the PPO algorithm, all necessary to understand the technical contributions of Chapter 5.

RL is a learning paradigm similar to SL. However, instead of learning to generalize to new inputs from labeled data, it learns through interaction with an environment via a reward system that guides learning in RL. Unlike in SL, where data are assumed to be independent and identically distributed (i.i.d.), in RL, previous decisions influence future inputs (i.e., data are not independent; it follows a sequential paradigm), and the ground truth answer is not known (correct actions/labels are not provided).

Although the RL field has been under study since the 20th century, it did not come to the fore until the last decade due to advances in Deep Learning (DL), and computational capabilities that ease their application. By the virtue of this progress, RL can leverage NNs to handle more complicated and diverse problems unapproachable in the past, which gives name to the field of Deep Reinforcement Learning (DRL), the RL paradigm we utilize in Chapter 5.

2.4.1 Reinforcement Learning Key Concepts

This subsection details the most important concepts of the RL framework necessary to understand the algorithms and methods used in Chapter 5.

Markov Decision Process

Commonly, RL problems are formalized as Markov Decision Processes (MDP), which is used to represent sequential decision making problems in which an agent interacts with a stochastic environment. Some key concepts are mentioned in this section and further explained in Subsection 2.4.1

Formally an MDP is defined by the tuple $\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma$. Here, $\mathcal{S}$ denotes the set of states, whereas $\mathcal{A}$ stands for the set of possible actions that the agent can perform over the environment. The function $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the transition probability function that sets the probability of transitioning between states given an action, while $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ represents the reward function that assigns a bonus to state-action pairs in the form of rewards. The discount factor, represented by $\gamma \in \mathbb{R}[0, 1)$ reduces the importance of future rewards.

At each discrete time step t , the agent perceives a state s_t from the state space $\mathcal{S}$ and makes a decision by selecting an action a_t according to its policy $\pi(a_t|s_t)$. This is the basic RL framework, illustrated in Figure 2.6. As a consequence, the environment is updated and the agent transitions to a new state according to $\mathcal{P}(s_{t+1}|s_t, a_t)$. Moreover, the agent decision is evaluated by means of a feedback signal, commonly referred as *reward* $r_t = \mathcal{R}(s_t, a_t, s_{t+1})$.

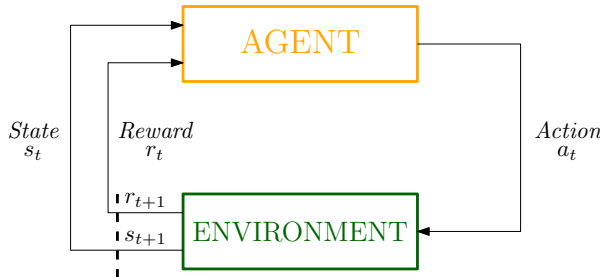


FIGURE 2.6: RL generic framework.

Nevertheless, in some scenarios the agent is not capable of perceiving the whole state information. In such situations, also referred as Partially Observable Markov Decision Process (POMDP), information about the state is limited by a observation function $f_o : \mathcal{S} \times \mathcal{A} \times \Omega \rightarrow [0, 1]$, where $f_o(s, a) \in \Omega$ and Ω representing the whole observation space. The agent receives observations $o_t \sim f_o(s_t, a_t)$. The ultimate goal of RL is to maximize the sum of rewards over time, which is conceptualized through the discounted return as $\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$, where the aforementioned discount factor γ is used to balance immediate rewards with longer-term gains. The return and reward concepts are further detailed below.

Episode and Rollout

The sequence or number of interactions between the agent and the environment can be divided into subsequences referred to as trajectory, rollouts or

episodes, with their meanings differing slightly depending on the boundaries used. In this Thesis, we follow a taxonomy commonly adopted in the literature. An episode concludes when a maximum number of steps $\mathcal{T}$ is taken or when the agent reaches the goal. Once an episode ends, the environment resets, and the agent returns to an initial state to attempt solving the environment again. In contrast, a rollout is not dependent on the episode's termination and consists of a fixed number of steps. As a result, a rollout may contain fewer experiences than an episode or even multiple episodes, with the number of experiences—the rollout size $\mathcal{T}$ —defined by the user independently of the environment. Trajectory is the most general term and can be used to refer to either of the previous two concepts.

Reward and Return

In RL, a reward is a scalar value provided to the agent by the environment following an action, serving as a signal to guide its learning process. Consequently, designing a reward function that offers meaningful feedback is crucial to the agent's success in achieving its goal. This reward reflects how effectively the agent's action aligns with the overall objective. The fundamental aim of the agent is to make decisions that maximize the cumulative reward, known as the return. It can be expressed as:

$$G_t = r_{t+1} + r_{t+2} + r_{t+3} + \dots + r_{\mathcal{T}}, \quad (2.5)$$

where t and $\mathcal{T}$ represent the current and final time steps of an episode, respectively. In this form, all decisions are treated equally, without considering their position in time. To balance the importance of current rewards w.r.t. rewards that are collected in the future, the already mentioned discount factor $\gamma \in [0, 1)$ was introduced, leading to:

$$G_t = r_{t+1} + \gamma^1 r_{t+2} + \gamma^2 r_{t+3} + \dots + \gamma^{\mathcal{T}-1} r_{\mathcal{T}}. \quad (2.6)$$

This form of the return not only enables controlling the relative importance of future decisions in the current one, but also solves the challenges that (2.5) poses in continuous tasks where no episode boundaries exist. In this scenarios, the return becomes an infinite sum. Thanks to the discount factor, the calculation remains finite by reducing the importance of future rewards after a sufficient number of steps, as these rewards become negligible over time.

Policy and Value Function

As previously explained, the agent interacts with the environment by taking actions. A policy, denoted as $\pi : \mathcal{S} \rightarrow \mathcal{A}$, is a function that maps the agent's current state to an action. Policies can be either deterministic or stochastic. A deterministic policy assigns a single action to each state, whereas a stochastic policy assigns a probability distribution over the possible actions an agent can take in each state.

The value function estimates the long-term reward an agent can expect from a given state or state-action pair under a specific policy π . The state value function, $V_\pi(s)$, estimates the expected return starting from state s and following policy π . Similarly, the action value function, $Q_\pi(s, a)$, estimates the expected return starting from state s , taking action a , and subsequently following policy π .

The key difference between these two value functions lies in the fact that Q_π calculates the expected return based on the assumption that the immediate action taken will be a_t , which determines the next state $s_{t+1} \sim \mathcal{P}(s_t, a_t)$ and the corresponding reward $r_{t+1} = \mathcal{R}(s_t, a_t, s_{t+1})$. In contrast, V_π does not assume any specific action, as the action selection depends on the behavior defined by policy π .

In addition to these value estimators, there is the advantage function, $A_\pi(s, a)$, which measures how beneficial a particular action a is in state s compared to the expected value $V_\pi(s)$ in that state. It is defined as:

$$A_\pi(s_t, a_t) = Q_\pi(s_t, a_t | s_t = s, a_t = a) - V_\pi(s_t | s_t = s). \quad (2.7)$$

Finally, a policy π is considered better than another policy π' if its expected return is higher, i.e., $\pi \succ \pi'$ if and only if $V_\pi(s) > V_{\pi'}(s)$. In this context, there exists an optimal policy π^* that is at least as good as any other policy. Correspondingly, the optimal value functions $V^*(s_t)$ and $Q^*(s_t, a_t)$ represent the best possible returns from each state s , assuming the agent follows the optimal policy π^* from that point onward.

On-policy or Off-policy

A common way to categorize RL algorithms is by examining how they utilize training data, which generally places them into two categories: on-policy and off-policy methods.

On-policy methods focus on improving the policy currently being used to interact with the environment. As a result, they can only leverage data that is representative of the current policy, π_t , and cannot use data collected under different policies, including previous policy states such as π_{t-1} , π_{t-2} , and so on. This restriction often makes on-policy algorithms less sample-efficient, meaning they require more data, but they tend to offer more stability during the learning process. Examples of on-policy algorithms include SARSA [54] and Trust Region Policy Optimization (TRPO) [55].

In contrast, off-policy methods learn a target policy using data generated by a different policy, referred to as the behavior policy. This means that learning occurs from experiences that are “off” the target policy. Off-policy algorithms are typically more sample-efficient, but they are more susceptible to overestimation and instability during training. Notable off-policy algorithms include Q-learning [56], its deep learning extension DQN [57], and Deep Deterministic Policy Gradient (DDPG) [58].

Value-based or Policy-based

RL algorithms can be classified into value-based and policy-based methods based on how they derive the policy.

Value-based methods focus on learning a value function that estimates the utility of each state (i.e., $V_\pi(s)$) or state-action pair (i.e., $Q_\pi(s, a)$). The goal is to minimize the difference between the predicted return for each state ($V_\pi(s_t)$ or $Q_\pi(s_t, a_t)$) and the actual return obtained. Since this return is based on experiences collected by the agent, it may not accurately reflect the optimal return. Furthermore, the trajectories gathered during training are diverse, as they depend on the evolving policy, π . To address the resulting high variance and instability in return estimates, several estimation techniques have been developed, though they are not discussed in this Thesis as they are not relevant to Chapter 5. Prominent algorithms in this category include Q-learning [56], SARSA [54], and members of the DQN family, such as [57].

In contrast, policy-based methods directly parameterize and optimize the policy without the need for a value function. These policies can be learned using derivative-free approaches like genetic algorithms or through policy gradient techniques. The objective in these approaches is to maximize performance, either by optimizing a fitness score or directly maximizing the expected return.

Both value-based and policy-based methods can result in deterministic or stochastic policies. In value-based methods, the agent typically learns the value of each action and selects the action with the highest expected outcome, leading to a deterministic policy. However, this deterministic behavior can be modified if needed. In policy-based methods, the agent learns a probability distribution over the available actions in discrete action spaces (or a distribution for each action in continuous spaces) and samples from this distribution to choose an action.

Policy Gradient Methods and Generalized Advantage Estimation

As previously mentioned, policy gradient methods are a type of policy-based approach that optimize the expected total reward by estimating the gradient. The loss function or objective they optimize has the following expression:

$$L^{PG} = \hat{\mathbb{E}}_t [\psi_t \log \pi_\theta(a_t | s_t)], \quad (2.8)$$

which results in the gradient:

$$\hat{g} = \hat{\mathbb{E}}_t \left[\sum_{t=0}^{\infty} \psi_t \nabla_\theta \log \pi_\theta(a_t | s_t) \right], \quad (2.9)$$

where $\hat{\mathbb{E}}$ represents the expected value and ψ is a variable that needs to be estimated.

One of the most widely used estimators for this purpose, since its introduction, is the Generalized Advantage Estimation (GAE) [59]. GAE is

an exponentially-weighted estimator of the advantage function, A_π , and is governed by two parameters: the aforementioned discount factor γ , which controls the influence of past rewards on the current state, and $\lambda \in [0, 1]$, which adjusts the bias-variance trade-off. Higher values of λ lead to a more evenly exponentially-weighted sum of advantages. In practice, λ is often set within the range of $[0.9, 1]$ to balance this trade-off effectively.

Actor-Critic

Actor-Critic (AC) methods are a class of algorithms that simultaneously learn both a policy and a value function. In this framework, the actor (decision-maker) selects actions, while the critic evaluates those actions by estimating the expected return. Both components are optimized together, with the actor receiving feedback from the critic in the form of a numerical value that indicates how beneficial or detrimental the chosen action was. The actor then adjusts its policy based on this feedback, while the critic refines its return estimates based on the actor's behavior, gradually improving overall performance. This is graphically depicted in Figure 2.7.

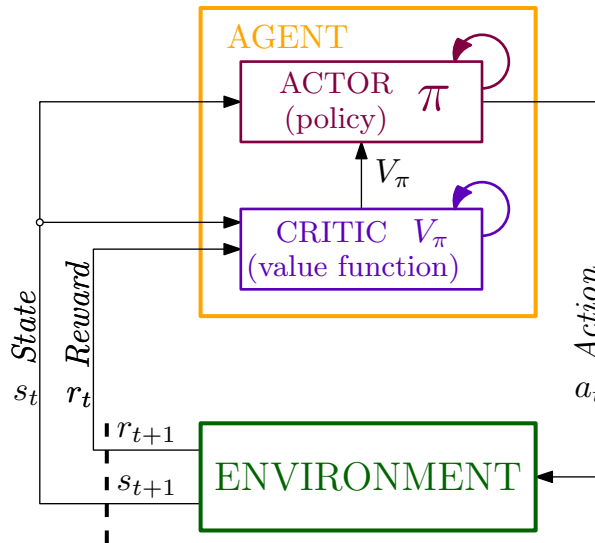


FIGURE 2.7: Actor-Critic (AC) method's workflow.

This framework blends elements of both value-based and policy-based (specifically, policy gradient) approaches, leveraging the advantages of each to counteract their respective limitations. For example, value-based methods can suffer from poor convergence, as even small changes in the value estimate may lead to significant policy shifts. In contrast, policy gradient methods, which are typically on-policy, offer more stable and smoother training. However, policy gradient methods tend to exhibit high variance and are often sample-inefficient, while value-based approaches are generally more sample-efficient and experience less variance.

In essence, AC methods represent a hybrid approach, combining the stability of policy gradients (actor) with the sample efficiency and reduced variance of value-based methods (critic). Prominent examples of AC algorithms include Deep Deterministic Policy Gradient (DDPG) [58] and Proximal Policy Optimization (PPO) [60], which are discussed in the next subsection.

2.4.2 Proximal Policy Optimization

PPO [60] is the algorithm that is used in the Chapter 5 to train the RL agent. It is an AC method that collects data in an on-policy manner and incorporates several concepts from an aforementioned on-policy algorithm, TRPO [55]. It stands out for being simpler than the latter, more general, and offering better sample efficiency. One of the key features of PPO is that it performs multiple gradient updates using the same batch of sampled data before discarding it. This differs from traditional on-policy policy gradient methods, which typically update the gradient only once per collected dataset.

PPO also introduces a modified objective (or loss) function inspired by TRPO’s formulation. Without delving into the mathematical formulations of PPO, since this thesis does not focus on RL, it is essential to highlight a key aspect of this objective. Specifically, it relies on the ratio between the new and old policies. Without proper constraints, this ratio can lead to overly large policy updates between consecutive iterations (π_t and π_{t+1}), which can destabilize training.

To address this issue, PPO offers two main approaches: one is to add a penalty term to the loss function based on the Kullback-Leibler (KL) divergence, and the other is to apply clipping directly to the objective function. The clipping method, which prevents the new policy from deviating too far from the old one, is the most widely adopted and is also the approach used in this thesis. The clipping mechanism depends on a hyperparameter, the clipping factor ϵ , which is commonly set to 0.2.

In addition to the core objective, PPO often includes an entropy term in the objective function. This entropy regularization encourages sufficient exploration during training [61], and its influence is controlled by a user-defined hyperparameter, ξ . As for the critic, the value function can be learned either as a separate module or by sharing parameters with the policy.

Lastly, PPO typically utilizes GAE as the ψ estimator. GAE is used to compute both the advantage estimates for the policy update and the return targets for the critic’s loss function.

Chapter 3

Out-of-Distribution Detection in Image Classification for Spiking Neural Networks

In the last decades, Machine Learning (ML) has been widely proven to outperform human capabilities in tasks that were previously thought to be intractable for machines, such as image/video classification, natural language processing or generative modeling. The unprecedented levels of performance achieved over these tasks have spurred an ever-growing proliferation of practical applications exploiting the capabilities of ML models. As introduced in Chapter 1, the sensitive nature of practical decisions that can be made as per the output elicited by ML models has lately motivated a flurry of research around the need for ensuring the trustworthiness of the audience consuming their inputs. This issue is particularly concerning when it comes to models whose internal structure and/or learning algorithm cannot be easily understood by non-expert users. This is the case of DL, arguably the field where the most outstanding breakthroughs have been reported in manifold domains [62, 63]. Concerns with the trustworthiness of DL models are even more acute as a result of their inherent black-box nature. In fact, in the already mentioned ethical guidelines document [1], the transparency stands as one of the requirements for a trustworthy AI system. Consequently, a new field of study has developed over the last decade, coined as eXplainable Artificial Intelligence (XAI), which focuses on making the decisions issued by ML models understandable for humans [64, 65, 66].

Besides the lack of interpretability of ML models, other factors may hinder the trustworthiness of ML models when deployed in real-world contexts. In such scenarios, models can face more varying and complex circumstances than the ones encountered in the controlled environments where their parameters are usually learned. As discussed in Chapter 1, using robust models is crucial to face these challenges, and the OoD detection framework is key in enabling such capabilities on new and existing

models. The OoD methods endow the ML models with the ability of discriminating between known and unknown inputs, i.e., inputs sampled from the training or ID and the out-distribution.

As previously mentioned in Section 2.1, the literature has tackled the OoD detection problem by focusing mainly on traditional or classic neural networks for image classification, using convolutional architectures for the purpose. Hence, more efficient models like SNNs have received no attention by the OoD community, despite the advantages they present for certain scenarios. They exhibit the ability to model complex temporal dynamics thanks to their event-based nature and huge energy gains when implemented in neuromorphic hardware [26, 67]. In fact, they are the best suited models for sensors like the Dynamic Vision Sensor (DVS), i.e., a bio-inspired vision sensor or camera that outputs pixel-level brightness changes instead of standard intensity frames, offering advantages like a very high dynamic range, no motion blur, and a latency in the order of microseconds [68]. Another concern of the current OoD methods is their lack of transparency, missing to attribute which parts of the input test instance are most influential for the detection of the whole sample as an OoD instance. In other words, OoD detection must be also explained for the audience of the model; otherwise, it could jeopardize the completion of a fully trustworthy ML pipeline.

In light of the aforementioned challenges, our approach utilizes information produced by a SNN model learning algorithm during the training process to build a representation of its typicality when processing ID data. A second contribution of this chapter is a local attribution method that highlights the parts of the input instance that push the developed detector towards classifying a sample as OoD. The attribution information produced by this method can improve the acceptability of the audience of the model when being notified about an *unknown* input, connecting clearly to the trustworthiness of ML pipelines sought in risk-sensitive domains as the ones exemplified previously. An extensive experimental setup over several image classification tasks is designed i) to compare the devised OoD detection method to existing alternatives that can be adapted to SNNs; ii) to verify whether a fusion of baseline OoD detection techniques with our designed detector can be synergistic and attain even better detection scores than those achieved by the detectors on their own; and iii) to assess whether the generated attributions succeed at identifying artificially induced image artifacts on the considered datasets. Results are promising, exposing a very competitive detection performance and attributions that conform to expectations.

The rest of the chapter is structured as follows: first, Section 3.1 mentions recent advances made in XAI for SNNs. Next, Section 3.2 presents the devised detector by describing its overall workflow, followed by a detailed description of the detection algorithm and the procedure to extract local explanations in the form of attribution maps. The experimental setup is described in Section 3.3, whereas results are discussed in Section 3.4. Finally, conclusions are drawn in Section 3.5, together with several technical extensions that could be targeted for future investigations.

3.1 Related Work and Contribution

This chapter lies in the intersection between several fields of research, namely, SNNs, the OoD detection problem, and XAI in SNNs. Regarding OoD detection and SNNs, a comprehensive contextualization is provided in Sections 2.1 and 2.2 of Chapter 2 respectively, citing the most important works in the corresponding research domains. In what follows, the research done in XAI related to SNNs is presented in Subsection 3.1.1 and the contribution of this chapter to the literature is explained in Subsection 3.1.2.

3.1.1 Explainability in Spiking Neural Networks

Concerning XAI, as in traditional neural networks, SNNs are inherently opaque models that need to be complemented with explanations and justifications of their issued predictions, particularly in critical applications involving a non-expert audience. Currently it is widely acknowledged that SNNs are not applied in practical modeling problems as often as their non-spiking counterparts. However, their event-based working procedure is making SNNs gradually replace non-spiking neural architectures in problems and tasks driven by computational efficiency. This progressively increasing prevalence of spiking neural computation approaches makes it of utmost necessity to derive XAI methods specific for this family of models, over as many tasks as possible, so that the time to their acceptance and adoption in real-world settings is effectively narrowed.

Despite this noted relevance, efforts invested so far in this direction are scarce. To begin with, the work in [69] proposed a new knowledge encoding method that enables the attribution of the input features in the classification decision in a Multi-Class Synaptic Efficacy Function based leaky-integrate neuRON (MC-SEFRON) classifier, a model without hidden layers based on the one presented in [70]. MC-SEFRON leverages a time-varying weight neuron model and trains it by using a modified STDP rule previously designed in preceding contributions. The result is a classifier with great accuracy when tested over the MNIST image classification task, and capable of eliciting explanations in the form of attribution maps for input images. The downside is that explanations are tightly bounded to the crafted model, and therefore are not generalizable to other spiking neuron architectures in more frequent use by the related community. Methods for rule extraction from SNNs are presented in [71] and also in [72, 73], where the NeuCube architecture [74] is used.

When searching for a more general explainability method, the contribution [75] exploited the fact that spikes with short inter-spike intervals highly contribute to the neural decision process. This observation gave rise to the Spike Activation Map (SAM) proposed in this study. For each timestep and neuron, a Neural Contribution Score (NCS) aggregates the Temporal Spike Contribution Score (TSCS) of all the pre-synaptic spikes previously processed by every neuron. These NCS are calculated at every time step for the feature maps of a convolutional neural network, and are summed pixel-wise across all channels, obtaining the aforementioned

SAMs. Hence, one heatmap is generated by this technique for every time step and SNN layer.

When shifting the focus on time series data, the contribution in [76] combines three components to design a local feature-based explanation called Temporal Spike Attribution (TSA): i) the influence of spike times, modeled by the NCS introduced above; ii) the influence of the model’s parameters, extracted from the weights of the connections between layers; and iii) the influence of the classification decision, inferred from the softmax values. To the best of our knowledge, this threefold approach is the more recent of the short series of contributions dealing with explainability techniques specifically devised for SNNs.

3.1.2 Contribution

After examining the current status of knowledge of the research fields tackled in this chapter, we note that multi-layered SNNs are lately becoming a viable option to implement and test in practical settings, performing competitively against non-spiking neural counterparts with notably improved energy efficiency. For instance, spiking-based CNN models comprising several neural processing layers have been proposed recently for different image classification tasks [77, 78, 79, 80]. This is the reason why the community is slowly expressing interest in addressing practical issues of these models when used in real-world environments, including the detection of OoD samples. However, as far as our revision of the literature can tell, there is no prior research in which OoD detection is studied for SNNs, nor have explanations been fabricated for modeling tasks besides classification. This chapter steps beyond the current SOTA by designing a specific method to detect OoD samples for SNNs, validating that not only it can be done by examining information produced during the training process of the SNN, but also that it attains better detection statistics than post-hoc methods from the literature that can be directly applied to these models. Further along this line, OoD predictions issued by the developed detector can be combined with those produced by other OoD detection methods so as to yield an ensemble of OoD techniques that improve further their individual performance scores. This exposes that predictions produced by the devised approach are complementary, coherent and diverse enough w.r.t. to other techniques for its fusion to enhance the overall detection figures. In addition, we propose an attribution method aimed to make the OoD detection process more understandable and trustworthy for the user, which spans further the narrow gamut of modeling tasks for which XAI methods have so far been developed in related studies.

3.2 Designed Method

We now describe the method designed to detect OoD samples in SNNs, which we hereafter denote as *Spike Count Pattern* (SCP) based detector. Its workflow is graphically sketched in Figure 3.1. First, a mathematical formulation of the network characteristics is posed in Subsection 3.2.1.

Next, an in-depth explanation of the detector is provided in Subsection 3.2.2. Finally, Subsection 3.2.3 describes the steps to produce an attribution map based on the information generated during the OoD detection performed by the SCP-based detector.

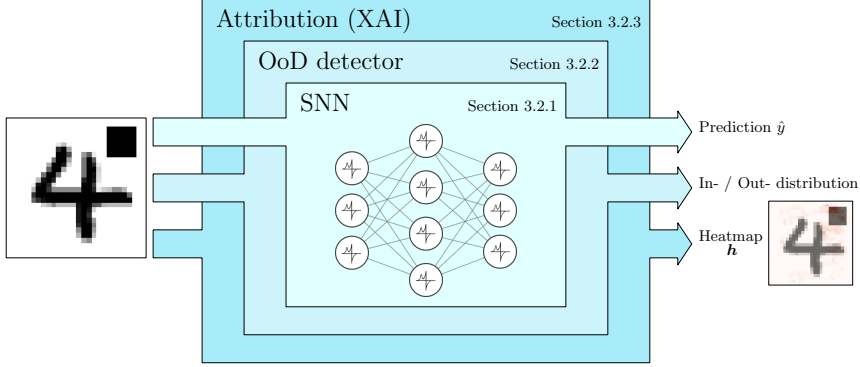


FIGURE 3.1: Workflow of the SCP based detector for an image classification task.

3.2.1 Spiking Neural Network: Mathematical Notation

Building on the definition and context of SNNs provided in Section 2.2, we now present the mathematical background necessary to understand the underlying concepts of these models. Specifically, our method focuses on detecting OoD instances by leveraging the internal mechanisms of SNNs that utilize rate-based encoding strategies. We assume a training dataset $\mathcal{D}_{tr}$ with samples $\{\mathbf{x}^{(i)}\}_{i=1}^M$, where $\mathbf{x}^{(i)} \in \mathbb{R}^D$, M denotes the number of samples in the training set, and D is the dimension of every input instance. Each of such training samples belong to a class $c \in \{1, 2, \dots, C\}$, where C denotes the number of classes in $\mathcal{D}_{tr}$. The SNN model requires spike trains $\mathbf{s}^{(i)}(t) = \{s_d^{(i)}(t)\}_{d=1}^D$ at their input which, when implemented in discrete-time processing hardware, are represented by binary sequences $\mathbf{s}^{d,(i)} = \{s_1^{d,(i)}, s_2^{d,(i)}, \dots, s_{T/\Delta T}^{d,(i)}\}$, where $s_t^{d,(i)} \in \{0, 1\}$ and a one represents the occurrence of a spike. Each sample will be processed during a certain amount of time, named the simulation time T , that is divided in time steps of width ΔT . The number of time steps within a simulation period is easily obtained as $T/\Delta T$, which establishes the length of the binary sequence in which spikes are represented in the discrete domain.

Hence, inputs must be converted in spike trains by an encoding method $g(\mathbf{x}^{(i)}) : \mathbb{R}^D \mapsto \{0, 1\}^{t \times D}$, where the addition of the time dimension is done if the data is static. Here a rate-based coding scheme that converts each feature to a spike train using an homogeneous Poisson process is employed [81]. Basically, for each feature, the probability of a spike $s_t^{d,(i)}$ being emitted at time $t \in \{1, \dots, T/\Delta T\}$ for feature $d \in \{1, \dots, D\}$ and instance

$\mathbf{x}^{(i)}$ is given by:

$$Pr\{s_t^{d,(i)}\} = r\Delta T = x_d^{(i)} r_{max} \Delta T, \quad (3.1)$$

where ΔT is the interval width, and r denotes the firing rate whose value depends on the value of each normalized feature $x_d^{(i)} \in \mathbb{R}[0, 1]$ and a maximum firing rate threshold r_{max} . The value of r_{max} is upperbounded by $1/\Delta T$ to avoid generating more than one spike per interval. To generate spike trains, at each timestep a sample $z_t^{d,(i)}$ is drawn uniformly at random from a continuous distribution $\mathcal{U}(0, 1)$, and is compared to the probability in Expression (3.1), such that a spike is generated if $z_t^{d,(i)} \leq r\Delta T$. As a result, the obtained spike train $\mathbf{s}^{(i)}(t)$ comprises randomly distributed spikes, such that the number of spikes in the simulation period T follows a Poisson distribution.

To process an instance $\mathbf{x}^{(i)}$, the SNN converts each of its features $x^{d,(i)}$ into a spike train $\mathbf{s}^{d,(i)}$. At each time step t , the corresponding part of all the spike trains $\{s_t^{1,(i)}, \dots, s_t^{D,(i)}\}$ are fed to the model. Assuming a LIF neuron model, the state of the LIF neurons of each SNN layer is updated depending on the occurrence of a spike in the previous layer and the previous state of the neuron. If an activation threshold is surpassed, the potential/voltage of the neuron is reset to its resting state value for the next simulation time step $t+1$. It is important to note that spikes affect the next layer's neuron states through the so-called *synaptic current*, which impacts on the membrane voltage of the neuron. This synaptic current generated by a spike depends only on the weight of the connection between the actual neuron (post-synaptic) and the neurons of the previous layer (pre-synaptic). The Figure 2.4 of Chapter 2 illustrates this behavior.

When the SNN is used for classification tasks, the output layer is not composed by LIF neurons, but rather by Leaky Integrator (LI) neurons matching the number of classes in $\mathcal{D}_{tr}$. This type of neurons only accumulates voltage through time, i.e., it lacks any mechanisms to fire a spike. The maximum voltage levels achieved by these neurons during the simulation period T are interpreted as the logits input to a softmax function, from where the corresponding predicted class is inferred under a *winner-takes-all* decision strategy (as in classical non-spiking neural classifiers).

3.2.2 Spike Count Pattern based Out-of-Distribution Detection

The design of our OoD detection approach departs from the intuitive observation that similar samples of the same class tend to have similar neuron activation patterns in the hidden layers. Indeed, the SCP-based detector is largely inspired by this observation to yield a novel OoD detection method, which is graphically depicted in Figure 3.2. In essence, the detection hinges on comparing the spike count pattern of every new test sample to the representative (archetypical) spike count patterns of its predicted label, which can be characterized during the training stage of the SNN. Therefore, the method requires augmenting the training algorithm of the SNN in order to characterize and produce such spike count archetypes, so that they can

be used at inference time to detect whether any test instance is out of the training distribution. Here, a spike count $q(\mathbf{s})$ corresponding to a spike train $\mathbf{s} \in \{0, 1\}^{T/\Delta T}$ is given by the number of spikes fired during the simulation period, namely, $q(\mathbf{s}) = \sum_{t=1}^T s_t$, where we recall that $s_t \in \{0, 1\}$ and value 1 represents a fired spike.

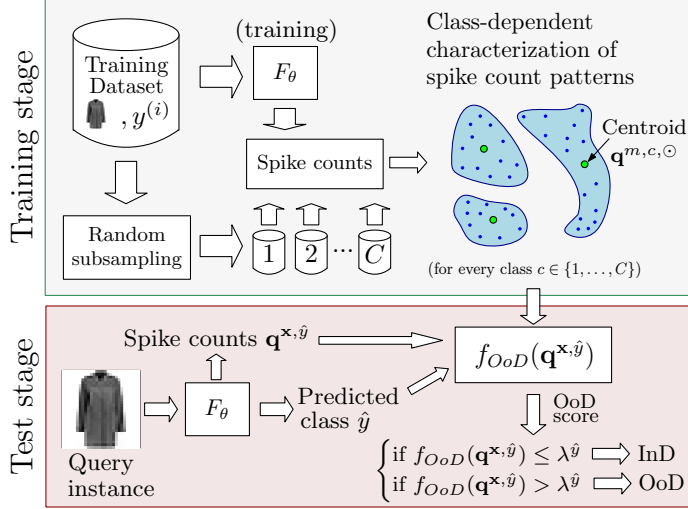


FIGURE 3.2: Graphical description of the methodology of the SCP based detector.

In doing so, and for the sake of a reduced computational complexity added to the training process, the SCP-based detector selects uniformly at random a specific number of training examples of every class and collects the spike counts of the layer prior to the softmax (for reasons disclosed later). Once spike counts have been computed, a class-conditional clustering is performed over the spike counts, using a distance metric to gauge the dissimilarity between spike counts. After computing all pairwise distances, the centroid of each class-conditional cluster resulting from the clustering stage is computed, which can be regarded as the aforementioned spike count archetypes of the samples inside the cluster of the class to which they belong. It should be noted that every class in $\mathcal{D}_{tr}$ may span more than one cluster, depending on the inherent intra-class variability of the spike counts produced by the training instances of the class at hand. Finally, at inference time, a test instance $\mathbf{x}'$ is processed through the already trained SNN, yielding a predicted class $\hat{y}$. A score is then computed to determine whether a new sample belongs to the in- or out-distribution, which is given by the closest L_1 distance to the archetypes of its predicted class $\hat{y}$. If this distance is greater than a predefined threshold λ , $\mathbf{x}'$ is declared to be an OoD sample. Otherwise, the SCP-based detector determines that $\mathbf{x}'$ belongs to the distribution of the training data.

We proceed by describing step by step the SCP-based detector. As schematically described in Algorithm 1, our developed technique requires a training stage for generating archetypal spike count patterns for every

Algorithm 1: Designed SCP detector.

Data: Training data $\mathcal{D}_{tr}$, size E of training samples $\mathcal{D}_{tr}^c$, last layer index L , spikes $\mathbf{s}^{n,L,(i)}$ associated to example $\mathbf{x}^{(i)}$, distance measure $d(\cdot)$, clustering algorithm, aggregation function $f_{agg}(\cdot)$, OoD score function $f_{OoD}(\cdot)$, thresholds $\lambda^{\hat{y}}$, query instance $\mathbf{x}$

Result: Decision on the OoD nature of $\mathbf{x}$

```

1 //Training stage
2 Train SNN over  $\mathcal{D}_{tr}$ 
3 for  $c \in \{1, \dots, C\}$  do
4    $\mathcal{D}_{tr}^c \leftarrow E$  sized  $c$ -class sample randomly selected from  $\mathcal{D}_{tr}$ 
5   Collect the spike trains  $\mathbf{s}^{n,L,(i)} = \{s_t^{n,L,(i)}\}_{t=1}^{T/\Delta T}$ 
6   Compute the spike counts  $q(n, L, (i), c)$  as per Expression (3.2)
7   Aggregate spikes counts into a vector  $\mathbf{q}^{(i),c} = \{q(n, L, (i), c)\}_{n=1}^{N_L}$ 

8   Compute distances  $d(\mathbf{q}^{(i),c}, \mathbf{q}^{(j),c}) \forall \mathbf{q}^{(i),c}, \mathbf{q}^{(j),c} \in \mathcal{D}_{tr}^c$ 
9   Extract  $M^c$  clusters  $\{\mathbf{Q}^{m,c}\}_{m=1}^{M^c}$ , where  $\mathbf{Q}^{m,c} \subseteq \{\mathbf{q}^{(i),c}\}_{i \in \mathcal{D}_{tr}^c}$ 
10  for  $m \in \{1, \dots, M^c\}$  do
11    | Compute archetype  $\mathbf{q}^{m,c,\odot}$  applying  $f_{agg}()$  to  $\mathbf{Q}^{m,c}$ ;
12  end
13 end
14 //Detection stage
15 Let  $\hat{y} = F_{\theta}(\mathbf{x})$  (prediction of query sample)
16 Collect spike trains  $\mathbf{s}^{n,L,\mathbf{x}}$ 
17 Compute spike count  $\mathbf{q}^{\mathbf{x},\hat{y}} = \{q(n, L, \mathbf{x}, \hat{y})\}_{n=1}^{N_L}$  Assign an OoD score
    by computing  $f_{OoD}(\mathbf{q}^{\mathbf{x},\hat{y}})$  as per Expr. (3.4)
18 Declare  $\mathbf{x}$  to be OoD if  $f_{OoD}(\mathbf{q}^{\mathbf{x},\hat{y}}) > \lambda^{\hat{y}}$ , otherwise  $\mathbf{x}$  is declared to
    be ID

```

class in the dataset. For simplicity we assume L stacked layers in the SNN architecture, comprising N_L LIF neurons. By slightly modifying the previous notation, we denote the time slot of the spike train occurring at the n -th neuron of the last layer L (the layer prior to the LI neurons) as $\mathbf{s}^{n,L,(i)} = \{s_t^{n,L,(i)}\}_{t=1}^{T/\Delta T}$. Based on this definition, the SCP-based detector starts by extracting E spike trains $\{\mathbf{s}^{1,L,(i)}\}_{i \in \mathcal{D}_{tr}^c}$ for each of the labels in the training dataset (line 5 of Algorithm 1). To this end, a E -sized subset $\mathcal{D}_{tr}^c \subset \mathcal{D}_{tr}$ of the training instances predicted to belong to label c is sampled at random (line 4), such that $|\mathcal{D}_{tr}^c| = P \forall c \in \{1, \dots, C\}$. Once such spike trains are collected, counts can be computed for each spike train at neuron n and instance $\mathbf{x}^{(i)}$ (line 6) as:

$$q(n, L, (i), c) = \sum_{t=1}^{T/\Delta T} s_t^{n,L,(i)}, \text{ for } i \in \mathcal{D}_{tr}^c, \quad (3.2)$$

which can be aggregated into a spike count vector $\mathbf{q}^{(i),c} = \{q(n, L, (i), c)\}_{n=1}^{N_L}$

(line 7). The SCP-based detector extracts the spikes of the *last layer* L of the SNN architecture for two reasons. The first is that the deeper the selected layer is along the hierarchy of neural layers, the higher the level of the features handled is, and the lower the number of spike count trains will be. High-level semantics are better for OoD sample discriminability, whereas low-dimensional spaces are preferred for characterizing the space of archetypical behaviors of the activations of instances of a given class. Second is the fact that the softmax layer is bounded to represent the probability of the input belonging to a certain class, which may not be well suited to characterize OoD samples.

Based on the E spike count vectors $\{\mathbf{q}^{(i),c}\}_{i \in \mathcal{D}_r^c}$ computed for every class c as per Expression (3.2), a clustering algorithm can be used over such spike counts to group them in terms of their inter-vector similarity (line 9). This clustering process yields M^c clusters $\{\mathbf{Q}^{m,c}\}_{m=1}^{M^c}$, where $\mathbf{Q}^{m,c} \subseteq \{\mathbf{q}^{(i),c}\}_{i \in \mathcal{D}_r^c}$ denotes the m -th cluster arising from the spike count activations of the random subsample drawn from the training instances predicted to belong to class c . To compute the dissimilarity between spike count vectors we select the L_1 norm (also referred to as *Manhattan distance*) between two vectors (line 8), namely:

$$d(\mathbf{q}^{(i),c}, \mathbf{q}^{(j),c}) = \sum_{n=1}^{N_L} |q(n, L, (i), c) - q(n, L, (j), c)|, \quad (3.3)$$

as its behavior in high-dimensional spaces is known to behave better than other alternatives, e.g., the Euclidean norm [82]. Finally, all samples in every cluster $\mathbf{Q}^{m,c}$ are aggregated to obtain an archetype $\mathbf{q}^{m,c,\odot}$ (line 11), which is considered the representative spike count pattern of all samples in the corresponding cluster. We do so by devising a function $\mathbf{q}^{m,c,\odot} = f_{agg}(\mathbf{Q}^{m,c})$, which can be chosen to be any aggregation strategy that allows for a variable-size set of spike count patterns in their argument, e.g. the mean, the median or any other statistic alike.

After computing the archetypes of every cluster and class $\mathbf{q}^{m,c,\odot}$, the OoD detection process is ready to process each new query or test sample $\mathbf{x}$ fed to the model. First, its prediction $\hat{y}$ is output by the trained SNN model (line 13). Along the inference process, spike trains $\mathbf{s}^{n,L,\mathbf{x}}$ are collected at the last layer of the network (line 14), from where spike counts vectors $\mathbf{q}^{\mathbf{x},\hat{y}} = \{q(n, L, \mathbf{x}, \hat{y})\}_{n=1}^{N_L}$ are computed (line 15). Next, an score is assigned to the test instance by using a scoring function (line 16):

$$f_{OoD}(\mathbf{q}^{\mathbf{x},\hat{y}}) = \min_{m \in \{1, \dots, M^{\hat{y}}\}} d(\mathbf{q}^{\mathbf{x},\hat{y}}, \mathbf{q}^{m,\hat{y},\odot}), \quad (3.4)$$

namely, the minimum L_1 distance between the spike count pattern $\mathbf{q}^{\mathbf{x},\hat{y}}$ of test instance $\mathbf{x}$ and the archetypes $\mathbf{q}^{m,\hat{y},\odot}$ for $m \in \{1, \dots, M^{\hat{y}}\}$ computed for its predicted class $\hat{y}$. Finally, the value of $f_{OoD}(\mathbf{q}^{\mathbf{x},\hat{y}})$ is used to decide whether instance $\mathbf{x}$ belongs to the in- or the out-distribution (line 17). If the score is greater than a certain class-conditional threshold $\lambda^{\hat{y}}$, the sample is considered as an OoD sample, declaring the sample from the ID

otherwise:

$$\mathbf{x} \text{ belongs to } \begin{cases} \text{ID} & \text{if } f_{\text{OoD}}(\mathbf{q}^{\mathbf{x}, \hat{y}}) \leq \lambda^{\hat{y}}, \\ \text{OoD} & \text{if } f_{\text{OoD}}(\mathbf{q}^{\mathbf{x}, \hat{y}}) > \lambda^{\hat{y}}. \end{cases} \quad (3.5)$$

3.2.3 Local Relevance Attribution Method

In case the query instance $\mathbf{x}$ is declared not to belong to the ID, the SCP-based detector is complemented by a local attribution-based explanation technique. This explanatory technique is local, as it operates over a given test instance towards generating a relevance attribution vector $\mathbf{h}(\mathbf{x}) \in \mathbb{R}^D$, where each component $h_d(\mathbf{x})$ quantifies the importance of each feature of the input when detecting it as an OoD sample.

The steps of this local attribution method are schematically illustrated in Figure 3.3 and detailed in Algorithms 2 and 3. In the following, we refer as *input layer* to the first fully-connected layer of the SNN network. In the case a feature extractor (e.g. a convolutional layer) is preceding that fully connected stage, the input layer will refer to the output of the feature extractor.

The core ideas driving the design of this method are two: on one hand, we aim to reproduce the detector’s procedure to detect OoD samples in the input space of the SNN rather than in the latent space where the detection is done (namely, the L -th layer prior to the softmax); on the other hand, since an attribution is sought, the method must account for the influence of each input feature on the aforementioned latent space. The latter is achieved by *propagating* the spike counts used for OoD detection at layer L back to layer 1 by using the weight matrices $\mathbf{W}^{[l]} : l \in [1, L - 1]$ of the network; and the former is achieved using a L_1 distance over the backpropagated spike counts. Here, l is the layer index.

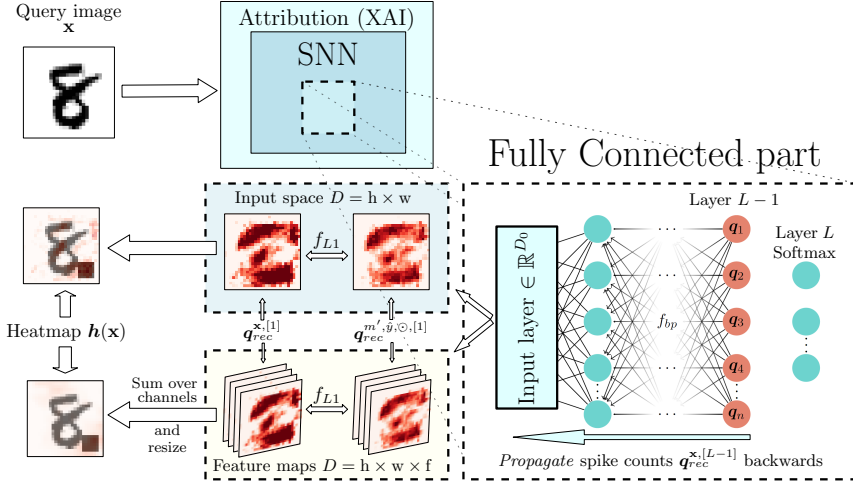


FIGURE 3.3: Schematic description of the local attribution method. Here, h , w and f refer to height, width and filters respectively.

The local attribution mechanism proceeds as follows: first, we back-propagate the spike counts of both the input $\mathbf{q}^{\mathbf{x},\hat{\mathbf{y}}}$ and the centroid $\mathbf{q}^{m,\hat{\mathbf{y}},\odot}$ to the input layer using the weight matrices $\mathbf{W}^{[l]}$. This is accomplished by recurrently applying a function f_{bp} defined as:

$$\mathbf{q}^{\mathbf{x},\hat{\mathbf{y}},[l-1]} = f_{bp}(\mathbf{W}^{[l]}, \mathbf{q}^{\mathbf{x},\hat{\mathbf{y}},[l]}) = \mathbf{W}^{[l]\top} \cdot \mathbf{q}^{\mathbf{x},\hat{\mathbf{y}},[l]}, \quad (3.6)$$

where we extend previous notation¹ with superindex $[l]$ to indicate that spike counts correspond to layer $l \in \{1, \dots, L-1\}$. This function can be viewed as the mapping of the spike counts of one layer to the space of the previous layer. Specifically, the spike counts are multiplied by the transpose $\mathbf{W}^{[l]\top}$ of the weight matrix at layer l . This operation simultaneously transforms the spike counts at the output of the layer back to its inputs, and implicitly implements a branching process similarly to what occurs in the backward pass of gradient calculus over computational graphs. Therefore, recurrently applying this function over all layers between the penultimate layer ($L-1$) and the input layer ($l=1$), we end up with the spike counts mapped to the input space, obtaining what we have called the *reconstruction* of the spike counts in the input space. This operation is described in Algorithm 2, which performs this mapping.

Algorithm 2: Reconstruction of spike counts at the input layer.

Data: Weight matrices $\mathbf{W}^{[l]}$, spike counts $\mathbf{q}^{\mathbf{x},\hat{\mathbf{y}},[L]}$, function to map spike counts to previous layer $f_{bp}(\cdot)$

Result: Spike counts mapped to input space $\mathbf{q}_{rec}^{\mathbf{x},[1]} \in \mathbb{R}^D$

```

1  $\mathbf{q}_{rec}^{\mathbf{x},[L-1]} = f_{bp}(\mathbf{W}^{[L]}, \mathbf{q}^{\mathbf{x},\hat{\mathbf{y}},[L]})$ 
2 for  $l \in \{L-2, \dots, 1\}$  do
3   |  $\mathbf{q}_{rec}^{\mathbf{x},[l]} = f_{bp}(\mathbf{W}^{[l+1]}, \mathbf{q}_{rec}^{\mathbf{x},[l+1]})$ 
4 end
5 Let  $\mathbf{q}_{rec}^{\mathbf{x},[1]} = \{q_{rec,d}^{\mathbf{x},[1]}\}_{d=1}^D$ 
6 for  $d \in \{1, \dots, D\}$  do
7   |  $q_{rec,d}^{\mathbf{x},[1]} = \max(q_{rec,d}^{\mathbf{x},[1]}, 0)$ 
8 end
```

It is important to remark the reason behind the maximum or ReLU operation done in the last step of Algorithm 2. To obtain a reasonable attribution, our goal is to reproduce the operating conditions of our detector, but in the input space rather than in the latent space. As LIF neurons are in use (whose output can be either a spike or nothing), when we integrate them in time (computing the spike counts) we will always obtain a positive or a 0 value, even if the weights arriving a neuron are all negative i.e. inhibitory. Therefore, to imitate this behavior when backpropagating static values (counts), negative values obtained from Expression (3.6) should be set to zero.

¹Accordingly, the spike counts $\mathbf{q}^{\mathbf{x},\hat{\mathbf{y}}}$ over which OoD detection is done as per Algorithm 1 should be denoted as $\mathbf{q}^{\mathbf{x},\hat{\mathbf{y}},[L]}$.

Algorithm 3: Construction of the relevance attribution vector.

Data: Weight matrices $\mathbf{W}^{[l]}$, spike counts $\mathbf{q}^{\mathbf{x},\hat{y},[L]}$ and the closest centroid $\mathbf{q}^{m',\hat{y},\odot}$ among $\{\mathbf{q}^{m,\hat{y},\odot}\}_{m=1}^{M^{\hat{y}}}$ resulting from Algorithm 2

Result: Relevance attribution vector $\mathbf{h}(\mathbf{x}) \in \mathbb{R}^D$

- 1 Use Algorithm 2 to obtain s ;
 - 2 Repeat Algorithm 2 to extract the reconstructed centroid $\mathbf{q}_{rec}^{m',\hat{y},\odot,[1]}$ from $\mathbf{Q}^{m',\hat{y}}$ for the closest centroid $m' \in \{1, \dots, M^{\hat{y}}\}$ to $\mathbf{q}^{\mathbf{x},\hat{y},[L]}$;
 - 3 **for** $d \in \{1, \dots, D\}$ **do**
 - 4 $h_d(\mathbf{x}) = |q_{rec,d}^{\mathbf{x},[1]} - q_{rec,d}^{m',\hat{y},\odot,[1]}|$;
 - 5 **end**
-

Second, after obtaining the reconstructions $\mathbf{q}_{rec}^{\mathbf{x},[1]}$ and $\mathbf{q}_{rec}^{m',\hat{y},\odot,[1]}$ (where m' stands for the closest centroid to the spike counts corresponding to sample $\mathbf{x}$), we seek how to reproduce the OoD detector's operation in this input space. To this end we resort to a L_1 distance as in Expression (3.3) so as to compute the absolute difference of each feature of both reconstructions, namely:

$$h_d(\mathbf{x}) = \left| q_{rec,d}^{\mathbf{x},[1]} - q_{rec,d}^{m',\hat{y},\odot,[1]} \right|, \quad \forall d \in \{1, \dots, D\}, \quad (3.7)$$

i.e., the relevance attribution vector results from the application of the same L_1 -based OoD detection criterion used in the SCP-based approach, yet in the reconstruction of the spike counts in the input space.

To end with the description of the devised relevance attribution technique, it is important to note that, depending on how the input data is encoded to the hierarchy of fully-connected LIF neural layers, an additional step would be needed to map $\mathbf{h}(\mathbf{x})$ (namely, a D -dimensional vector) to the dimensions of the original data. We exemplify this additional operation by describing two cases:

- Let us assume that $\mathbf{x}$ is an image, so that it is input to the network by means of a fully-connected input layer with as many LIF neurons as the number of pixels and channels of the image. In this case, $\mathbf{h}(\mathbf{x})$ should be *unflattened* in the same order as the one followed to serialize the input image into the network.
- If a convolutional layer is used to extract features from the input image, the dimension D of the relevance attribution network depends on the size of the convolutional filters and their cardinality. In this case, the rearrangement of the relevance attribution vector operates first by rearranging its components to the shape of the feature maps of the convolutional layer, yielding a three-dimensional tensor-like structure with width and height equal to the size of the convolutional filters, and with depth equal to the number of convolutional filters. Then, we perform a depth-wise aggregation of this tensor. Finally, the obtained map is

scaled up by interpolation to generate a relevance attribution heatmap with the size of the original input image.

3.3 Experimental Setup

In order to examine the performance of the developed detection and attribution techniques, we design a comprehensive experimental setup using SNNs for image classification, aimed to inform with empirical evidence the responses to the following research questions:

- RQ1: Does the SCP-based detector perform competitively when compared to post-hoc OoD detectors from the literature adapted to SNN architectures?
- RQ2: How does the SCP-based detector perform when fused with other adapted OoD detection methods?
- RQ3: Does the local relevance attribution method yield informative explanations about the features that make a given sample be detected as OoD? Which are the limitations of this explainability technique?

The design of the experiments devised to answer the above questions include the definition of the SNN architectures (Subsection 3.3.1), the ID and OoD datasets under consideration (Subsection 3.3.2), the OoD methods selected for comparison and the scores used to measure their performance (Subsection 3.3.3), and the methodology followed to qualitatively validate the relevance attribution vectors generated for OoD samples by the SCP-detector (Subsection 3.3.5). In what follows, details about these design choices are given. All scripts and notebooks producing the results reported in this chapter have been made publicly available in a GitHub repository (https://github.com/aitor-martinez-seras/OoD_on_SNNs).

3.3.1 Considered SNN Architectures

We start describing the experimental setup by the specific implementation of the SNN under consideration. SNNs comprise a very wide research field that lies in between Neuroscience and ML. Hence, some SNN variants are designed departing from the bio-plausibility perspective, attempting to make a machine simulate faithfully the biological brain, neurons and its synapses. Other variants, however, pursue SNNs capable of solving complex modeling tasks due to their inherent advantages rather than the biological plausibility of the dynamics and neurons used. The latter design criterion has dominated this area in recent times due to the need for more efficient ML algorithms. As a result, a great variety of SNN frameworks are nowadays available [83]. Although some of them seek a realistic simulation of the human brain, others offer simpler albeit more effective SNN implementations for data-based modeling, in terms of the the depth of the networks that can be trained and the complexity of the tasks they can tackle.

As stated in the introduction, our experiments focus on practical fully-spiking SNN implementations, namely, networks that are trained and run entirely on data encoded as temporal spikes. Specifically, we use 2^{nd} order LIF neurons that account for synaptic conductance. By denoting the membrane voltage as ϑ and the synaptic current as ζ , these decay over time at rates τ_{mem} and τ_{syn} , respectively. The ordinary differential equations that are integrated over time (in discrete time steps) are [84]:

$$\dot{\vartheta} = 1/\tau_{mem}(\vartheta_{leak} - \vartheta + \zeta), \quad (3.8)$$

$$\dot{\zeta} = -1/\tau_{syn}\zeta, \quad (3.9)$$

together with the jump condition:

$$z = \Theta(\vartheta - \vartheta_{th}), \quad (3.10)$$

and transition equations:

$$\vartheta = (1 - z)\vartheta + z \cdot \vartheta_{reset}, \quad (3.11)$$

$$\zeta = \zeta + \zeta_{in}, \quad (3.12)$$

where ϑ_{th} defines the membrane voltage that, when exceeded, leads to the emission of a spike; z is a variable taking value 1 if the neuron voltage surpasses ϑ_{th} (0 otherwise); ϑ_{leak} refers to a constant voltage value that is leaked from the neuron; $\Theta(\cdot)$ stands for the Heaviside step function that returns value 1 when the threshold is surpassed; and ϑ_{reset} is the value of voltage that is set when the latter happens. A SNN framework that features this LIF model is Norse [85], which runs on top of PyTorch and provides rate-based encoding strategies, LIF-based and Izhikevich neuron models, and the SuperSpike training algorithm mentioned in Section 2.2.

We have tested our method over two different types of SNN architectures for image classification: a Fully Connected (FC) architecture and a Convolutional Neural Network (CNN) architecture. In both cases, the encoder used to translate pixel values to spike trains is the so-called Poisson encoder, whereas neuron models are LIF (intermediate neurons) except for the output layer, which utilizes LI neurons followed by SoftMax. LI cells are similar to LIF neurons, but never fire: they only accumulate voltage. Parameters for both types of cells are $\vartheta_{leak} = 0$, $\vartheta_{reset} = 0$, $\tau_{mem} = 0.005 \text{ ms}^{-1}$, and $\tau_{syn} = 0.01 \text{ ms}^{-1}$. These parameters are identical for all cells in the SNN, except for the voltage threshold ϑ_{th} , which is specified layer-wise in Tables 3.1 (gray-scale datasets) and 3.2 (color datasets) along with the architectures of the networks in use for every case. Information in these tables can be interpreted as explained in their captions.

All gray-scale models are trained for 5 epochs using an Adam solver with learning rate equal to 0.002. The simulation time T is 50 ms, whereas the time step duration ΔT is 1 ms, hence yielding 50 time steps per sample. For the model used over color datasets, the simulation time is expanded to 64 time steps per sample, whereas AdamW optimizer was used for a total of 75 epochs.

TABLE 3.1: SNN architectures considered in the experiments for gray-scale datasets. Each layer is defined by a number and some letters, which respectively define the number of neurons and the type of connections. Each layer is separated by a hyphen. **E** denotes Poisson encoding neurons, **fc** stands for fully connected neurons, **conv** means convolutional filters and **Flt** refers to a flatten. The subscript of the last two defines the voltage threshold ϑ_{th} of the LIF neurons. In the last layer only LI neurons are used, followed by a SoftMax activation, being the number of output neurons equal to the number of classes in the ID dataset.

Network Type	Reference	Architectures
Fully Connected SNN (FC-SNN)	FC_1	784E – 200fc _{0.25} – LI
	FC_2	784E – 300fc _{0.25} – 200fc _{0.25} – LI
Convolutional SNN (CNN-SNN)	CNN_1	784E – 20conv _{0.2} avgpool – 50conv _{0.2} – Flt – 300fc _{0.1} – LI
	CNN_2	784E – 20conv _{0.2} avgpool – 50conv _{0.2} – Flt – 500fc _{0.1} – 300fc _{0.05} – LI

TABLE 3.2: SNN architectures considered in the experiments for color datasets, following the same structure and legend as in Table 3.1. The new keyword **cpa** refers to a convolutional layer with padding equal to 1 and average pooling.

Network Type	Reference	Architectures
Convolutional SNN (CNN-SNN)	CNN_{rgb}	784E – 32cpa _{0.25} – 64cpa _{0.2} – 128cpa _{0.1} – Flt – 2048fc _{0.1} – 100fc _{0.1} – LI

3.3.2 In-Distribution and Out-of-Distribution Data

In order to evaluate the performance of the SCP-based detector, we have followed common practice in the literature related to OoD detection. Consequently, datasets of well-known image classification tasks have been considered as ID datasets. Specifically, for the gray-scale datasets we have used 28×28 centered images: the MNIST dataset [86], composed by 60,000 images of handwritten images with 10 classes (digits from 0 to 9); Fashion MNIST [87], which comprises 60,000 images of 10 different classes of clothes; KMNIST [88], consisting of 60,000 images of Japanese characters (*Kanjis*), each belonging to one among 10 classes; and finally, EMNIST Letters [89], which provides 145,000 images of 26 types of letters of the Latin alphabet. In the case of color images, CIFAR10 and SVHN are used, composed by 32×32 images and 10 classes each, the former comprising animals and vehicle classes and the latter house numbers from 0 to 9 obtained from Google Street View images. In addition, the techniques are evaluated over neuromorphic image classification datasets, i.e., datasets whose examples are composed by event-based data coming, for example, from a DVS. The selected ID datasets are NMNIST [90], a spiking version of the original frame-based MNIST dataset; and DVSGesture [91], a dataset created for developing a gesture recognition system with 11 hand gestures from 29 subjects under 3 illumination conditions.

When it comes to OoD datasets, we consider all possible combinations of pairs of the ID datasets defined above, establishing one of them as the dataset over which the SNN is trained, and querying the trained SNN

with the others, independently done for the gray-scale, color and neuro-morphic datasets. For the sake of a wider experimental coverage, we have added several additional OoD datasets for both gray-scale and RGB cases. For the first case, we have considered **notMNIST** [92], which gathers font glyphs extracted for the letters A through J; **omniglot** [93], which amounts to 1,623 different handwritten characters from 50 different alphabets; and **CIFAR10-BW**, the black-and-white rescaled version (to 28×28 pixels) of the CIFAR10 dataset [94]. As a result, the SNN trained for every ID dataset will be queried with images drawn from 6 different OoD datasets. In all cases, the OoD dataset is reduced to a uniform stratified sample of 10,000 samples. For the case of colored datasets, we add **Food101**, consisting of 101 food categories; **LSUN (crop)** and **LSUN (resize)**, 2 small subsets of the LSUN scene classification dataset, the former composed by randomly cropping image patches of size 32×32 and the latter by downsampling each image to size 32×32 ; **Flowers102**, 102 categories of commonly occurring plants of United Kingdom; **Caltech101**, a collection of pictures of objects belonging to 101 classes; **GTSRB**, which comprises 43 classes of German traffic signs; **TinyImageNet**, a smaller version of ImageNet with 200 classes and images downsampled to 64×64 pixels; **EuroSAT**, a land cover classification dataset using Sentinel-2 satellite images; and finally, **DTD**, which is composed by images of 47 classes of different textures. Finally, we consider as OoD neuromorphic datasets **CIFAR10DVS** [95], the neuromorphic version of CIFAR10; and **DVSLip** [96], an event-based lip-reading dataset.

3.3.3 Post-hoc Methods

For comparison purposes, to the best of our knowledge there is unfortunately no OoD detector specifically designed for SNNs. Consequently, our benchmark considers post-hoc OoD techniques that essentially rely on the output of the neural network being monitored, either the logits or the output of the softmax activation. In the case of the SNN architectures selected for our experimentation, logits correspond to the maximum voltage reach by every LI cell during the simulation time. The methods against which the SCP-based detector is compared include the MSP method [8], ODIN [12] and the energy-based approach presented in [15].

The MSP method does not require any parameter tuning, whereas ODIN and the energy-based method are governed by several parameters such as the temperature or the input preprocessing (in the case of ODIN). Values of these parameters must be properly set in order to obtain good results. To this end, as stated in Section 2.1.1 of Chapter 2, for the energy-based detector we follow their study [15] that states that can be used parameter-free by defining $T = 1$. For the temperature parameter of ODIN, as there is not such option, we have instead evaluated several temperature values for each (ID, OoD) dataset combination, reporting on the configuration that yields the best performance in each case.

Regarding the configuration of the SCP-based detector, the subset of instances of each class used for the characterization of the spike count patterns is formed by $E = |\mathcal{D}_{lr}^c| = 1,000$ samples per class. Agglomerative

hierarchical clustering was used to extract clusters from the spike counts of all training instances used for characterization. As for the creation of the centroids, a median statistic was selected as the aggregation function $f_{agg}(\cdot)$, which is robust against spike count outliers. Finally, the threshold for each class λ^c is selected utilizing a subset of 1,000 samples per class extracted from the training data and different from the ones used for the creation of the centroids. To do so, the OoD scores are computed and the threshold λ^c is defined as the value that yields a target percentage of that class-wise subset that will be correctly classified as ID.

3.3.4 Out-of-Distribution Detection Metrics

To quantify the capability of the methods in the benchmark to detect OoD samples, the scores used are the mainstream metrics below presented.

But first, is important to notice that the OoD detection task, in essence, is as a binary classification task where the outcome is that the evaluated sample is an ID or an OoD sample. Also, for clarification, in what follows a True Positive (TP) refers to an ID sample correctly classified as such, whereas a True Negative (TN) stands for an OoD sample detected correctly by the OoD detector at hand. Thus, False Positive (FP) is an OoD sample incorrectly detected as ID. The metrics that are defined in this section are based on the True Positive Rate (TPR) and the False Positive Rate (FPR), defined as:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (3.13)$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \quad (3.14)$$

Based on this definitions, we can now specify the metrics that are used to asses the capabilities of different OoD detectors:

- **FPR95** (FPR at 95% TPR). Is the rate of OoD samples wrongly classified as positive (ID) when the TPR is 95%.
- **AUROC** (*Area Under the Receiver Operation Characteristic curve*). It is a threshold-independent score for binary classification that can be regarded as the probability that the model ranks a random positive example with higher score than a random negative example. It is defined as TPR/FPR .
- **AUPR** (*Area Under the Precision-Recall curve*). Is another threshold-invariant score closely related to AUROC. The precision-recall curve is given by the relationship between $\text{TP}/(\text{TP} + \text{FP})$ (*precision*) and TPR (*recall*).

3.3.5 Evaluating the Quality of the Relevance Attribution Method

When aiming to inform an answer to RQ3, there is no standardized way to measure the quality of explanations provided by a relevance attribution technique like the one developed in this chapter. To overcome this, we opt for evaluating our method against OoD datasets whose instances are built from ID samples with imprinted visual artifacts. Since such artifacts are a priori known to be the parts of the image that *push most* the input sample towards the out-distribution, the relevance attribution technique should highlight such artifacts in the produced heatmap for the image at hand. In other words, if the difference between two samples is a certain extra and/or missing feature, our relevance attribution method should focus on them when performing OoD detection, and therefore should be clearly distinguishable in the heatmap.

In doing so, MNIST will be used as ID dataset, whereas OoD datasets used for the qualitative attribution assessment include i) **MNIST-Square**, that is, the MNIST dataset with a square of 5×5 pixels placed randomly in one of the corners of the image (with a separation to the border of the image of 2 pixels); and ii) **MNIST-C** [97], another MNIST-based dataset that includes several modifications made to the digits. Specifically, we use the *zig-zag* corruption pattern in our experiments. An example showing the three datasets used in our experiments related to RQ2 is depicted in Figure 3.4.

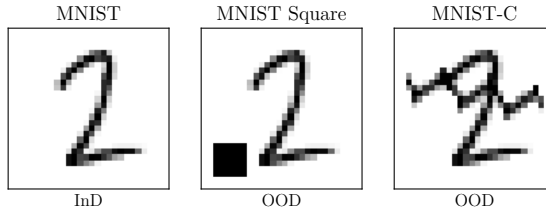


FIGURE 3.4: Datasets for the experiments related to the quality of relevance attribution heatmaps.

To correctly visualize the heatmaps, an appropriate plot range must be selected previously. To that end, a random sample is extracted from the training set and its heatmaps are computed. Then, from the pixel values of all the heatmaps, the maximum value and the q -th quantile value are obtained, where q is the TPR selected to define the threshold. The maximum of the visualization range will be the calculated maximum value and the minimum will be the q -th quantile value. The size of the random sample used to compute these maximum and quantile values is set to 100 examples.

3.4 Results and Discussion

This section presents the results obtained from the experiments, along with an analysis that leads to answers to the previously introduced research questions: RQ1 (Subsection 3.4.1), RQ2 (Subsection 3.4.2) and RQ3 (Subsection 3.4.3). The latter includes a short analysis of the limitations that the relevance attribution maps produced by our detector undergo in regards to the complexity of the SNN network structure, as well as the visual features observed by the trained SNN.

3.4.1 RQ1: Does the SCP-based detector perform competitively when compared to post-hoc OoD detectors from the literature adapted to SNN architectures?

This question will be answered differentiating three types of datasets, each with its own dedicated subsection: Subsection 3.4.1.1 for gray-scale datasets, Subsection 3.4.1.2 for color datasets and Subsection 3.4.1.3 for neuromorphic datasets.

3.4.1.1 Gray-scale Datasets

Scores obtained to address *RQ1* are shown in Tables 3.3 and 3.4 for the gray-scale datasets and in Table 3.5 for the color datasets. In all tables, dark gray shaded cells indicate, for each dataset configuration and score, the winning approach among the four OoD detection methods under comparison. Likewise, light gray shaded cells correspond to the second best method in terms of AUROC score (in general the ranking of the methods in the benchmark as per this score is maintained for the rest of scores).

TABLE 3.3: OoD detection scores obtained for the gray-scale FC-SNN architecture. The arrow next to every score indicates whether higher ($\uparrow$) or lower ($\downarrow$) values correspond to a better detection performance.

ID dataset	OoD dataset	AUROC $\uparrow$				AUPR $\uparrow$				FPR95 $\downarrow$			
		SCP (ours)	MSP	ODIN	Energy	SCP (ours)	MSP	ODIN	Energy	SCP (ours)	MSP	ODIN	Energy
MNIST (96% acc)	FMNIST	92.91	74.19	74.19	39.71	91.41	65.67	65.67	41.12	23.15	76.34	76.34	93.00
	KMNIST	95.26	80.97	80.97	60.83	95.47	76.68	76.68	55.30	21.84	72.40	72.40	89.06
	Letters	88.38	84.25	84.25	69.17	88.83	83.31	83.31	65.79	50.33	71.15	71.15	89.40
	notMNIST	99.29	64.42	64.42	20.67	99.25	57.59	57.59	34.39	02.59	86.29	86.29	97.62
	omniglot	72.91	93.61	95.89	95.80	75.06	94.07	95.98	95.91	88.54	44.10	24.81	24.90
	CIFAR10-BW	99.60	80.13	80.13	16.92	99.58	78.94	78.94	33.34	00.96	80.51	80.51	99.68
FMNIST (84% acc)	MNIST	92.73	86.77	96.02	97.34	93.72	85.64	95.77	97.66	36.62	46.19	17.18	15.90
	KMNIST	86.23	78.79	88.37	87.29	85.90	81.75	88.76	87.73	50.72	79.00	57.01	65.11
	Letters	92.64	81.27	92.34	93.15	93.37	80.06	91.80	93.92	34.79	62.97	34.68	44.99
	notMNIST	98.53	61.80	61.80	43.30	98.97	62.28	62.28	44.74	02.67	93.22	93.22	95.99
	omniglot	79.25	92.16	98.07	98.70	81.49	91.61	98.16	98.98	78.62	30.63	08.01	06.00
	CIFAR10-BW	97.10	55.22	55.22	33.38	97.64	60.05	60.05	40.33	12.62	94.70	94.70	98.99
KMNIIST (87% acc)	MNIST	63.24	84.27	85.68	82.44	63.51	87.37	88.50	85.76	86.84	57.08	49.65	63.59
	FMNIST	72.11	76.58	76.58	46.86	67.25	78.31	78.31	49.50	63.47	70.38	70.38	90.74
	Letters	75.64	85.16	86.51	83.09	73.47	88.26	89.29	86.10	74.25	54.76	47.28	60.32
	notMNIST	96.66	67.35	67.35	38.70	96.72	67.66	67.66	43.82	08.95	80.90	80.90	96.04
	omniglot	54.15	94.43	96.66	96.38	57.38	95.57	97.57	97.32	97.21	18.64	05.19	08.02
	CIFAR10-BW	95.01	79.94	79.94	45.68	95.35	82.60	82.60	55.28	16.55	65.71	65.71	99.41
Letters (86% acc)	MNIST	72.58	78.58	81.31	81.34	72.57	77.01	79.94	80.35	85.30	74.11	66.35	67.72
	FMNIST	88.78	71.64	71.64	34.47	87.66	72.07	72.07	39.68	37.67	87.27	87.27	92.98
	KMNIST	87.53	77.76	77.76	66.62	87.40	76.22	76.22	61.94	49.27	78.00	78.00	82.41
	notMNIST	98.48	58.33	58.33	24.81	98.32	54.19	54.19	35.51	06.12	93.57	93.57	98.01
	omniglot	77.47	94.86	97.72	97.88	80.72	95.22	97.83	97.92	86.92	28.63	12.27	10.51
	CIFAR10-BW	98.81	53.28	53.28	07.80	98.88	53.48	53.48	32.55	03.94	97.97	97.97	99.89

We start by discussing on the results for the gray-scale datasets. Several observations can be done by examining the results corresponding to the FC-SNN model (Table 3.3). First, the ours SCP-based OoD detection method achieves in general better detection scores, specially for the MNIST and Letters datasets. Only results obtained for omniglot are worse for the SCP-based approach than for the rest of methods in the benchmark. In the case of the KMIST dataset, the SCP-based detector dominates the comparison only when notMNIST and CIFAR10-BW are used as OoD datasets. Another interesting fact is that the scenarios where our devised method achieves the best results, all other logit-based OoD detectors fall apart, scoring significantly worse. Conversely, when any logit-based method performs better than our method, our proposal is never ranked second within the benchmark. This suggests that OoD predictions issued by the SCP-based detector succeed at rectifying those produced by logits-based approaches when they are fused together.

Another observation is that there are some cases where ODIN and MSP methods perform equally. The reason is that, as explained in Subsection 3.3.3, several values of the temperature parameter for each combination of datasets have been tested, reporting in all cases the best score. This could occur for a temperature value equal to 1, which would effectively convert ODIN into the MSP method. Actually, this occurs in the cases where our method clearly dominates the benchmark, supporting the idea that logits are not suitable for detecting OoD samples in some cases.

TABLE 3.4: OoD detection scores obtained for the gray-scale CNN-SNN architecture.

ID dataset	OoD dataset	AUROC $\uparrow$				AUPR $\uparrow$				FPR95 $\downarrow$			
		SCP (ours)	MSP	ODIN	Energy	SCP (ours)	MSP	ODIN	Energy	SCP (ours)	MSP	ODIN	Energy
MNIST (97% acc)	FMNIST	92.83	91.64	93.00	91.51	94.53	92.86	93.38	91.92	49.60	54.75	37.87	44.60
	KMNIST	97.23	89.64	90.23	88.33	97.75	90.25	90.51	88.67	13.36	57.13	50.88	58.52
	Letters	92.03	86.04	87.35	86.06	91.19	85.42	86.85	85.75	34.75	63.65	57.05	62.21
	notMNIST	97.10	87.42	88.48	88.04	97.56	86.54	87.60	87.57	17.57	61.43	54.74	58.59
	omniglot	96.91	89.16	90.67	90.06	97.46	88.51	89.78	89.21	15.83	55.73	45.24	48.25
	CIFAR10-BW	94.05	94.83	96.86	95.04	95.84	96.16	97.53	96.19	49.57	45.15	22.29	41.23
FMNIST (77% acc)	MNIST	97.45	49.83	65.76	82.50	97.97	52.79	65.84	80.07	09.34	95.10	88.00	51.39
	KMNIST	97.57	68.51	79.60	88.70	98.09	70.75	80.63	88.06	09.92	87.96	79.97	48.09
	Letters	96.16	53.63	66.31	78.39	96.72	52.99	63.30	74.03	17.64	91.64	85.06	58.87
	notMNIST	95.95	65.24	72.27	73.41	95.52	63.03	68.60	70.38	15.00	87.33	87.46	85.95
	omniglot	97.68	61.10	74.49	86.62	97.67	61.21	72.73	83.80	06.35	91.50	82.96	44.55
	CIFAR10-BW	92.14	83.39	91.23	87.03	92.79	85.41	92.73	89.22	36.89	70.20	61.87	75.06
KMIST (87% acc)	MNIST	83.96	84.85	86.49	86.43	86.07	87.97	88.91	88.82	56.50	61.79	51.07	53.44
	FMNIST	60.07	74.17	76.94	73.53	63.41	70.51	72.70	70.29	89.34	76.99	66.97	78.09
	Letters	82.26	84.26	87.30	87.01	82.34	86.75	88.98	88.57	55.23	61.34	46.21	48.33
	notMNIST	83.06	84.53	86.99	84.72	84.29	85.52	85.99	83.99	55.66	58.34	42.38	52.86
	omniglot	92.93	84.43	87.78	87.35	93.89	86.33	89.11	88.47	25.60	59.33	43.72	46.16
	CIFAR10-BW	47.35	85.66	92.24	85.36	54.90	89.30	94.10	89.43	99.27	64.13	32.33	69.18
Letters (82% acc)	MNIST	75.90	75.36	75.55	75.49	77.47	79.47	78.71	78.41	81.93	88.74	90.63	91.35
	FMNIST	79.06	76.23	76.23	64.47	84.18	82.24	82.24	69.43	84.80	88.46	88.46	90.42
	KMNIST	86.15	77.05	79.11	74.91	88.61	79.43	81.07	77.74	68.37	82.22	86.67	92.81
	notMNIST	83.35	65.74	65.74	62.31	84.62	66.67	66.67	67.12	66.84	91.88	91.88	96.61
	omniglot	91.57	78.36	84.53	85.18	92.61	80.94	85.64	86.06	43.95	79.14	69.93	68.46
	CIFAR10-BW	67.49	77.56	77.56	49.36	76.24	84.09	84.09	64.09	99.69	93.68	93.68	99.96

We now shift the focus of the discussion on the results for the CNN-SNN architecture shown in Table 3.4. In this second experiment, our method outperforms the rest of competing OoD detectors in a wider number of cases. Exceptions are Letters and KMIST datasets, especially the latter where the SCP-based detector only performs best in 1 out of 6 OoD datasets. Furthermore, performance gaps in those cases where it dominates the rest of the benchmark are small. It is also worth noting that,

although our proposal attains superior results in less cases than in the FC-SNN architectures, the differences to the winning OoD technique are not that large in those cases.

The smaller performance gaps observed in this second set of experiments call for a further examination of their statistical significance. To this end, we resort to two different methodologies widely used by the community for the purpose, and apply them to the AUROC scores reported in Tables 3.3 and 3.4:

- A critical distance diagram [98], which employs a post-hoc Nemenyi test to compute a critical distance value; if the average rankings of two methods as per the paired score values achieved across different cases are separated by more than the critical distance value, they can be considered to perform significantly different to each other. Otherwise, if their average ranks differ in less than the critical distance, they are declared to be statistically equivalent.
- A Bayesian posterior analysis of the differences between the paired performance scores achieved by the techniques under comparison. The work in [99] exposed that standard hypothesis testing was not suitable for assessing the significance of performance differences for a number of issues. Alternatively, they proposed to perform a Bayesian sign test of the performance scores of two different techniques, yielding an estimation of the probability that one technique outperforms the other using the scores obtained by each of them over all cases. The probability resulting from this analysis can be carried out via Monte Carlo sampling and depicted in a system of barycentric coordinates, in which three regions are distinguished: one where the first OoD technique outperforms the second, a second one where the second outperforms the first, and a third region reflecting *practical equivalence*. This latter region delimits the probability that techniques can be concluded to perform equivalently from the statistical point of view, and depends on a parameter (*rope*) that establishes the minimum difference between the scores of the OoD techniques for considering their performance as significantly different. The value of *rope* depends on the task being solved: in our case, *rope* is set equal to $\text{AUROC} = 1\%$.

Figure 3.5 depicts the result of the first statistical significance study over the results obtained for the (a) FC- and (b) CNN-SNN architectures. It is straightforward to note that for both SNN architectures, our method ranks first on average, yet its dominance over the rest of methods cannot be concluded to be statistically significant. Therefore, it is necessary to perform a closer study between our SCP-based detector and the second best one, namely, ODIN.

This closer examination is done by performing a Bayesian analysis of the differences following the methodology described above using the AUROC scores of ODIN and our SCP-based detector. Figures 3.6.a and 3.6.b show the results of this second significance study. When it comes to the scores obtained for the FC-SNN architecture (Figure 3.6.a), almost all

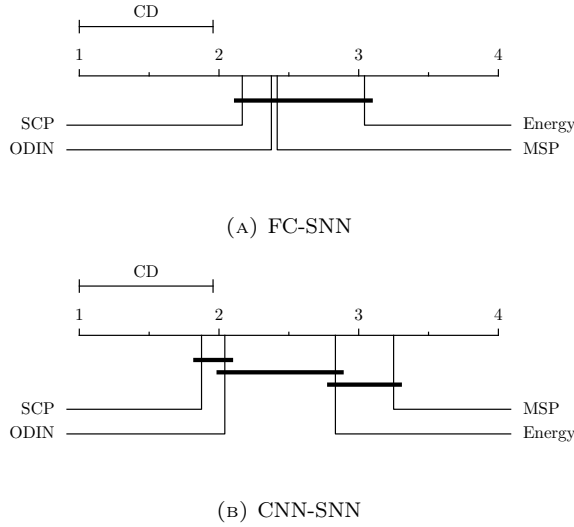


FIGURE 3.5: Critical distance diagrams computed over the AUROC results across the cases considered in the experiments of BW datasets for (a) FC- and (b) CNN-SNN architectures. The value of the critical distance is shown on top of every diagram. Horizontal lines connecting the average rank markers of several detectors indicate that their average rankings cannot be claimed to be different to each other with statistical significance.

Monte Carlo samples drawn from the adjusted Bayesian posterior probability lie in between the regions where ODIN or the SCP-based detector achieve better results. This means that none of the methods perform equivalently, and one always outperforms the other with statistical significance. This goes in line with the observation made in the discussions on Table 3.3, where it was concluded that either a logit-based approach or the SCP-based detector performs best in the majority of cases, being very few the scenarios where both perform similarly. The Bayesian analysis reported in Figure 3.6.a informs that the probability that our method performs better than ODIN is approximately 73%, whereas ODIN is expected to perform better for the remaining percentage.

We follow this second significance analysis by exploring the output of the Bayesian analysis over the results obtained for the CNN-SNN architecture. Figure 3.6.b clearly reveals that either our method performs better (approximately 94%) or ODIN does (6%), whereas the probability that both perform equivalently (*rope* region) is almost negligible.

3.4.1.2 Color Datasets

Regarding the color datasets, the results are more evenly matched as exposed by the scores reported in Table 3.5. However, conclusions obtained are similar to the ones drawn for the experiments with gray-scale datasets.

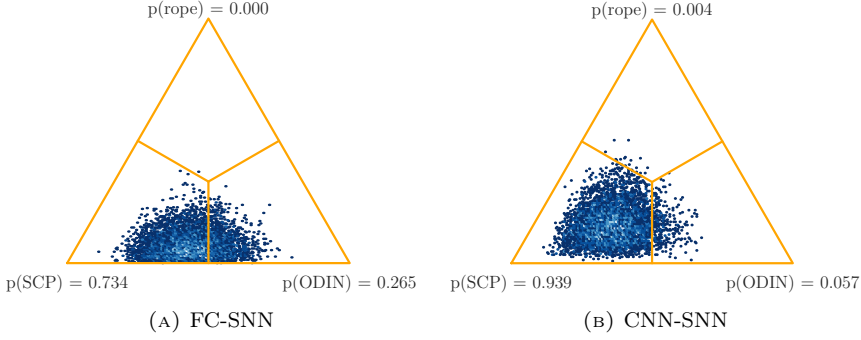


FIGURE 3.6: Posterior probability resulting from the Bayesian signed tests computed over the paired AUROC scores attained by ODIN and the SCP-based detector with (a) the FC-SNN architecture; (b) the CNN-SNN architecture.

TABLE 3.5: OoD detection scores obtained for the RGB CNN-SNN architecture.

ID dataset	OoD dataset	AUROC $\uparrow$				AUPR $\uparrow$				FPR95 $\downarrow$			
		SCP (ours)	MSP	ODIN	Energy	SCP (ours)	MSP	ODIN	Energy	SCP (ours)	MSP	ODIN	Energy
CIFAR10 (65% acc)	Food101	71.65	62.69	68.10	67.45	70.30	64.06	70.12	69.80	81.56	91.49	89.99	92.31
	LSUN (crop)	87.63	65.91	74.09	74.67	86.97	70.84	77.93	78.40	49.06	93.14	90.15	92.42
	LSUN (resize)	69.54	65.65	72.80	74.46	68.95	67.01	74.21	74.97	84.18	89.11	86.81	84.59
	Flowers102	75.77	62.79	68.71	67.75	75.04	64.71	70.93	70.32	75.59	92.47	90.54	91.84
	Caltech101	67.02	61.79	65.88	66.47	64.47	63.75	67.90	68.58	83.81	92.52	91.53	91.79
	GTSRB	78.76	67.03	71.81	69.12	78.10	70.41	75.42	73.61	69.49	92.11	91.38	91.71
	SVHN	62.70	66.73	70.10	69.72	61.21	69.28	72.27	72.06	88.51	90.50	89.41	90.17
	TinyImagenet	54.03	70.89	70.89	69.61	55.63	75.39	75.39	74.70	95.34	90.44	90.44	93.02
	EuroSAT	65.88	66.49	70.00	69.56	64.91	68.73	71.43	70.70	85.69	90.04	88.96	89.60
	DTD	61.43	76.86	81.76	73.19	63.91	80.22	85.12	83.92	90.60	84.28	83.95	80.71
SVHN (75% acc)	Food101	61.73	61.95	63.97	65.64	59.24	61.54	64.94	67.58	84.89	91.49	92.71	92.55
	LSUN (crop)	84.74	76.35	74.84	80.20	85.82	78.02	81.40	83.76	50.59	90.97	94.20	80.90
	LSUN (resize)	73.20	77.39	81.73	82.80	76.87	90.41	85.49	86.29	73.95	86.09	86.75	77.49
	Flowers102	80.47	74.29	77.92	81.32	81.66	79.00	82.68	84.66	60.79	90.28	93.20	78.76
	Caltech101	80.94	73.85	77.24	78.39	82.75	78.50	81.86	82.42	60.35	89.45	91.97	84.31
	GTSRB	81.82	73.74	77.02	79.88	82.23	78.34	81.85	83.45	56.24	93.60	90.09	82.10
	CIFAR10	83.11	78.02	81.01	78.90	83.80	77.85	81.10	81.79	55.59	86.42	88.64	81.76
	TinyImagenet	80.23	73.73	77.80	80.90	81.23	78.67	82.71	84.46	60.81	90.93	93.08	81.21
	EuroSAT	79.06	74.02	77.73	80.47	79.38	78.53	82.25	83.87	60.84	89.79	92.32	80.48
	DTD	81.16	74.11	77.69	81.01	92.85	78.78	82.39	84.42	56.75	81.61	81.61	78.89

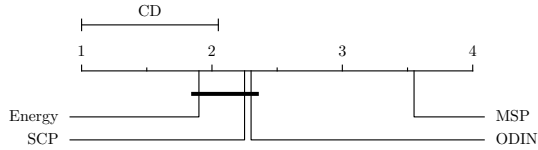


FIGURE 3.7: Critical distance diagram computed over the AUROC results across the cases considered in the experiments for color datasets.

In general, the detection performance of all methods is worse than in the previous set of experiments, due to the fact that the accuracy of the model is significantly lower (approximately 65% and 70% for CIFAR10 and SVHN, respectively). In these two ID datasets, SCP outperforms the other detection methods over 5 out of the 10 OoD datasets. This can be further observed in Figures 3.7 and 3.8. In the critical distance diagram, where obtained average rank is taken into account, the winning method is Energy, but without statistical significance as the distance of its average

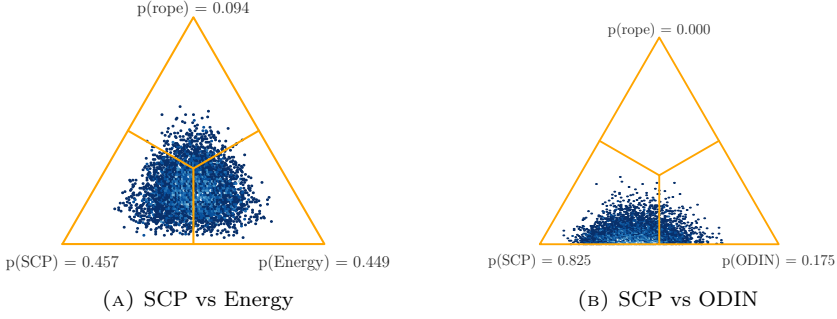


FIGURE 3.8: Posterior probability resulting from the Bayesian signed tests computed over the paired AUROC scores attained by ODIN, Energy and the SCP-based detector.

rank to that of ODIN and the SCP is smaller than the critical distance. Therefore, further exploration is carried out by means of a Bayesian analysis, concluding that our method works significantly better than ODIN (approximately 82% of the times), while performing on par with Energy, as both perform better than the other with equal likelihood (45%).

3.4.1.3 Neuromorphic Datasets

Finally, we analyze the results obtained for the neuromorphic datasets, which are summarized in Table 3.6. It can be observed that the SCP-based detector clearly outperforms its competitors in all the ID vs OoD combinations, except for the **DVSGesture** vs **NMNI**, where all datasets attain an optimal detection performance. In fact, our SCP based detector obtains near-perfect scores (more than 99% AUROC) when **NMNI** is the ID dataset.

TABLE 3.6: OoD detection scores obtained for the RGB CNN-SNN architecture in neuromorphic datasets.

ID dataset	OoD dataset	AUROC $\uparrow$				AUPR $\uparrow$				FPR95 $\downarrow$			
		SCP (ours)	MSP	ODIN	Energy	SCP (ours)	MSP	ODIN	Energy	SCP (ours)	MSP	ODIN	Energy
NMNI (97% acc)	DVSGesture	99.76	83.29	83.29	81.96	99.80	87.39	87.39	85.18	00.00	84.77	84.77	80.50
	CIFAR10DVS	99.89	88.86	92.28	92.17	99.90	90.16	92.9	92.96	00.00	66.15	42.51	43.01
	DVSLip	99.12	92.05	92.05	89.81	99.42	93.44	93.44	90.56	01.12	54.13	54.13	50.07
DVSGesture (70% acc)	NMNI	100.0	100.0	100.0	90.72	100.0	100.0	100.0	95.25	00.00	00.00	00.00	43.01
	CIFAR10DVS	85.74	53.89	66.56	57.85	87.35	56.36	65.95	58.63	24.62	82.20	71.21	85.98
	DVSLip	99.20	81.27	95.02	83.41	98.32	77.84	93.40	84.08	00.76	36.36	11.74	58.33

Results are in line with expectations, as SNNs are by design well suited for processing and modeling temporal data. Therefore, samples of these neuromorphic datasets can be characterized more clearly in the activations space than in the logit space, giving rise to a better latent embedding for the spike count characterization performed by SCP.

3.4.2 RQ2: How does the SCP-based detector perform when fused with other adapted OoD detection methods?

To inform this research question with evidence, different ensembles have been built by fusing the OoD detection methods considered for the benchmark. At this point it is important to note that the fusion of several techniques is only worth when i) they yield similar levels of performance for the datasets being evaluated; and ii) when their decisions for a given input are *diverse*. Otherwise, poorly performing techniques can introduce noise to the final fused decision and ultimately, degrade the overall performance of the ensemble. Therefore, ensembles will be assessed for the color datasets, where the OoD methods to be fused perform most closely to each other.

Figure 3.9 illustrates how the fusion of two OoD detection methods A and B works: for a given query instance X_{test} , the predictions issued by both fused methods are combined together under an AND criterion, so that the outcome of the ensemble for the query instance is OoD (out of distribution) only if both methods (A and B) predict the instance as such. Otherwise, the sample is predicted to belong to the ID dataset. By building ensembles with different methods, we aim to elucidate whether the two methods perform diversely yet coherently enough for their combination to yield better detection scores w.r.t the performance of the two methods in isolation.

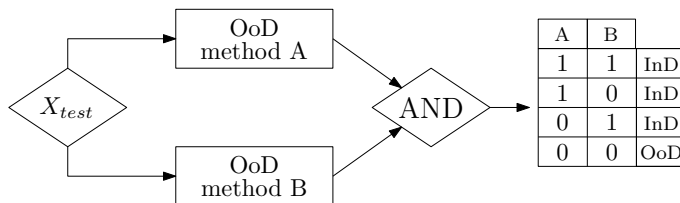


FIGURE 3.9: Fusion of OoD detection methods and voting criterion for their predictions.

When addressing RQ2, several ensembles are created, namely, *E-ODIN-Energy*, *E-ODIN-SCP* and *E-Energy-SCP*, and evaluated over color datasets, where differences reported in response to RQ1 are narrower. If the fusion of models produce similar decisions for the test samples in the different experiments, the ensemble will not improve the scores achieved by its fused methods. On the other hand, if decisions by the combined methods differ from each other, the fusion may yield differences in detection performance. Table A.1 included in Appendix A comprehensively collects the detection performance figures for these ensembles. For the sake of clarity, we herein discuss on the CD diagram (Figure 3.10) and the barycentric plot resulting from the Bayesian analysis of performance differences (Figure 3.11).

We begin by inspecting the critical distance diagram, which clearly depicts that both Energy and ODIN (logits-based approaches) perform outstandingly better when fused with the SCP-based detector. Contrarily,

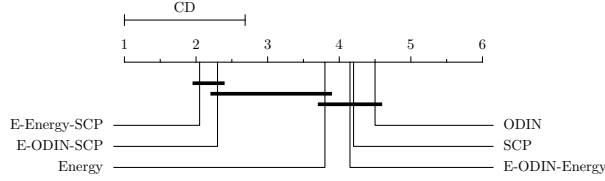


FIGURE 3.10: Critical distance diagram of ensembles, ODIN and Energy evaluated over color datasets.

when Energy and ODIN are fused together to yield the *E-ODIN-Energy* ensemble, the fusion of predictions seem not to lead to significant performance gains. Actually, all three methods' and the *ODIN-Energy* ensemble's average ranks are separated by less than the critical distances, meaning that their average ranks are not statistically different from each other. As in RQ1, this lack of statistical significance calls for a deeper analysis towards solidly answering RQ2.

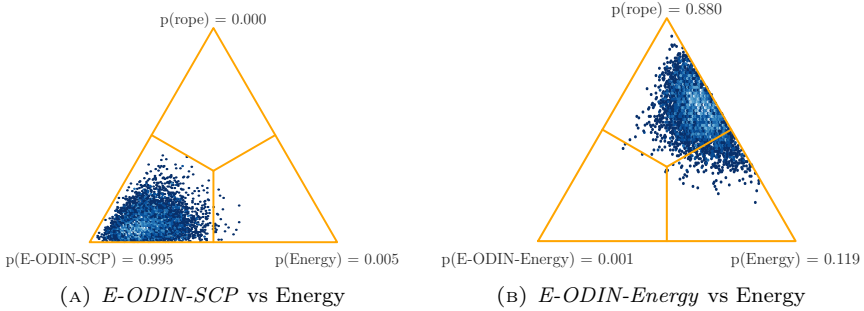


FIGURE 3.11: Posterior probability resulting from the Bayesian signed tests computed over the paired AUROC scores attained by the *E-ODIN-SCP* ensemble, the *E-ODIN-Energy* ensemble and the Energy method.

When applying a Bayesian analysis over the obtained results, we take into account not only the difference in ranks, but also the difference in raw performance, using AUROC as the metric like in the previous cases. In order to make a fair comparison, we compare the *E-ODIN-SCP* and *E-ODIN-Energy* ensembles to the Energy method, to verify whether the fusion of SCP with ODIN is more beneficial than combining ODIN with Energy. Such comparisons can be observed in Figures 3.11.a and 3.11.b, which clearly show how the fusion with our SCP-based technique outperforms Energy almost all the times. Specifically, Figure 3.11.a demonstrates that the *E-ODIN-SCP* ensemble, despite using ODIN (a lower performing method when compared to Energy), turns out to achieve better results with very high likelihood (99.6%). Conversely, the *E-ODIN-Energy* ensemble is not even capable of outperforming Energy, one of the techniques that composes it. Together with the detailed report of scores in Table A.1, it can be concluded that the predictions issued by our SCP-based method, when fused with those made by other OoD detectors, allow detecting OoD instances more effectively than any other approach in the benchmark, either

individual methods or any other ensemble not considering the SCP-based detector at its core.

3.4.3 RQ3: Does the local relevance attribution method yield informative explanations about the features that make a given sample be detected as OoD? Which are the limitations of this explainability technique?

At this point we recall that the local explanation technique for the SCP-based detector aims to generate a relevance attribution vector $\mathbf{h}(\mathbf{x})$ for a query sample $\mathbf{x}$ (i.e. a heatmap when dealing with image classification). The value of its components $\{h_d(\mathbf{x})\}_{d=1}^D$ indicates the relative relevance of regions of the input space that have influenced most the detector when detecting that the query sample is OoD. The heatmap must also deliver information about an *OoD score* of the sample, such that the intensity of the heatmap should be higher as the OoD score of the sample increases, and vice-versa.

Based on the above intuition, the qualitative evaluation of the attribution method made to reply RQ3 proceeds as follows: first, attribution generated for several test images with known artifacts are shown and analyzed to verify that the produced heatmaps spot those artifacts. Next, limitations of the attribution method noted during the experimentation are exposed, using examples and results that clearly illustrate these constraints.

We concentrate the discussion on FC-SNN and CNN-SNN architectures trained to classify MNIST images (ID dataset), using query instances from MNIST-Square and MNIST-C as OoD datasets. The detection performance of the SCP-based OoD detector in this setup is reported in Table 3.7, indicating values of the three detection scores used so far. We note that in this case the value of FPR80 is reported, i.e., the FPR value obtained when the target TPR is set to 80%. In other words, the detection threshold is set such that 80% of the positive (ID) samples are correctly detected as such. It is important to note that all the other three logit-based detectors considered in the benchmark obtain worst detection scores in this setup.

ID dataset	OoD dataset	Model	AUROC $\uparrow$	AUPR $\uparrow$	FPR80 $\downarrow$
MNIST	MNIST-Square	FC-SNN	96.04	96.70	02.92
		CNN-SNN	69.73	69.06	55.14
	MNIST-C	FC-SNN	83.73	83.03	27.05
		CNN-SNN	75.06	73.64	45.11

TABLE 3.7: Scores for the SCP-based detector in the experiments devised to address RQ3.

After training both SNN architectures over the MNIST dataset, several ID and OoD samples are tested and processed through the SCP-based detector and the local relevance attribution technique. Figure 3.12 depicts several of such query images. The text at the bottom of each image represents the predicted class for the image, whereas the number above the image corresponds to the difference between the OoD score for the given sample – given by the value returned by $f_{OoD}(\mathbf{x}, \hat{y})$ as per Expression (3.4) – and the threshold $\lambda^{\hat{y}}$ for a TPR of 80%. Hence, a positive value above every image means that the image is detected as OoD. Conversely, the more negative this value is, the more ID the sample can be thought to be, as it is *farther* from being detected as OoD based on the criterion in Expression (3.5).

The discussion starts with the images shown for the FC-SNN architecture (Figure 3.12.a). The first example, a digit corresponding to class 5, the rightmost tip of the digit is highlighted as relevant even if clearly belonging to the ID. Indeed, despite its largely negative score (therefore being confidently detected as ID), usually the 5 digits in the MNIST dataset do not reach the edges of the image. In this case, the 5 sample depicted in the figure has an untypically long tip. In the square and zigzag case this observation is further buttressed: not only the same digit with square and zigzag artifacts preserve a highlighted right tip, but relevance is also attributed by our approach to the parts of the image affected by the artifacts. The 5 digit with the zigzag pattern is not detected as OoD, but its OoD score is close to 0, hence the image is more likely to be an OoD than the normal case. Consequently, the produced heatmap is less intense. The 8 digit comprising square and zigzag artifacts would be detected as OoD, as both artifacts are partly remarked in the heatmaps. Some parts of the clean 8 digit image are also marked in the heatmap: this can be explained because the closest cluster for the samples with the corruption changes w.r.t. the clean digit from where they are produced, which may be composed by digits with different compositional characteristics. In the last example, the 4 digit, the zigzag case is predicted to be a digit 9 rather than a digit 4, and thus highlights several regions where the shape of an archetypical 9 could be.

Digits shown for the CNN-SNN architecture exhibit in general noisier and smoother heatmaps, as relevance attribution vectors at the input of the architecture are scaled up via interpolation to match the dimensions of the original input image (in this case, from 11×11 to 28×28 pixels). In all examples included in Figure 3.12.b, the artifacts induced in the clean 1, 9 and 7 digits are highlighted in the heatmap. In the case of the digit 7, the noisier heatmaps are due to the fact that the predicted classes are 9 (square) and 2 (zigzag) rather than a 7. Hence, the parts of the image spotted by the heatmaps include both the artifacts and the parts of the original digit that make the spike count patterns deviate from those of the closest centroid of its wrongly predicted class.

In summary, we conclude that the devised relevance attribution technique can generate heatmaps that provide meaningful information about the features of the input that make the SCP detector identify it as an OoD

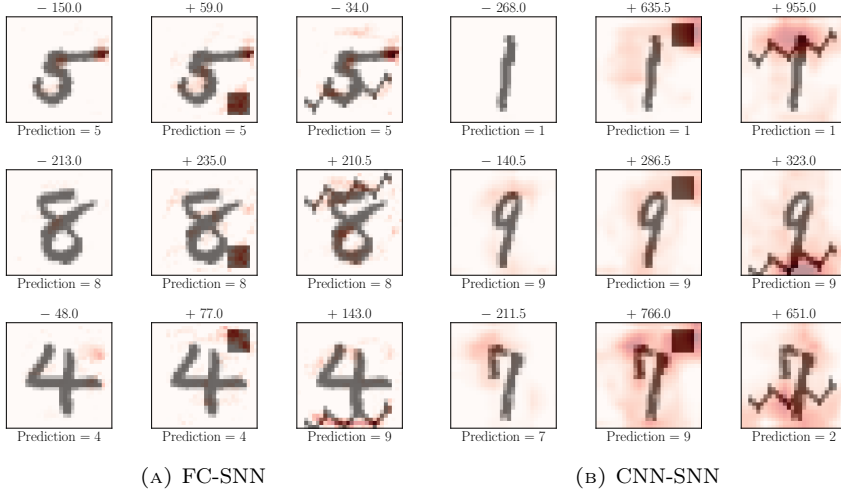


FIGURE 3.12: Heatmaps showing the relevance attributed to different digits (clean and corrupted versions with square and zigzag artifacts) for (a) a FC-SNN architecture; (b) a CNN-SNN architecture. Each heatmap is superimposed on its input, and for every image its predicted class is reported, together with the difference between the its OoD score and the class-conditional threshold $\lambda^{\hat{y}}$.

sample. However, during our experiments we noted several limitations of the approach that spur future technical extensions to the algorithm that are later described in the concluding part of the chapter. Such limitations are:

- *The depth of the network degrades the quality of attribution heatmaps:* one effect observed when applying the relevance attribution technique is that heatmaps become noisier as the depth of the fully connected part of the SNN increases, i.e., the number of fully connected layers L . This effect can be observed in Figure 3.13, where the same inputs as in Figure 3.12 are used over FC-SNN and CNN-SNN architectures with $L = 2$ fully-connected layers, instead of the single-layered architecture considered in previous experiments.

As noted in these plots, in general the attribution technique elicits less sharply defined heatmaps, evincing its lower capability to emphasize important features in the input image towards the OoD detection. Nonetheless, for some samples their predicted class has changed. Specifically, the digit 8 with the square artifact in the FC-SNN model, and the digit 7 in the CNN-SNN architecture. Moreover, the lower definition and quality of the produced attributions is not reflected on the detection scores, as can be verified in Tables A.2 and A.3 and in Figure A.1 of A.2. While in the case of the CNN-SNN architecture a small performance loss is noted, when it comes to the FC-SNN architecture the developed SCP-based detector yields even better detection scores when compared to ODIN.

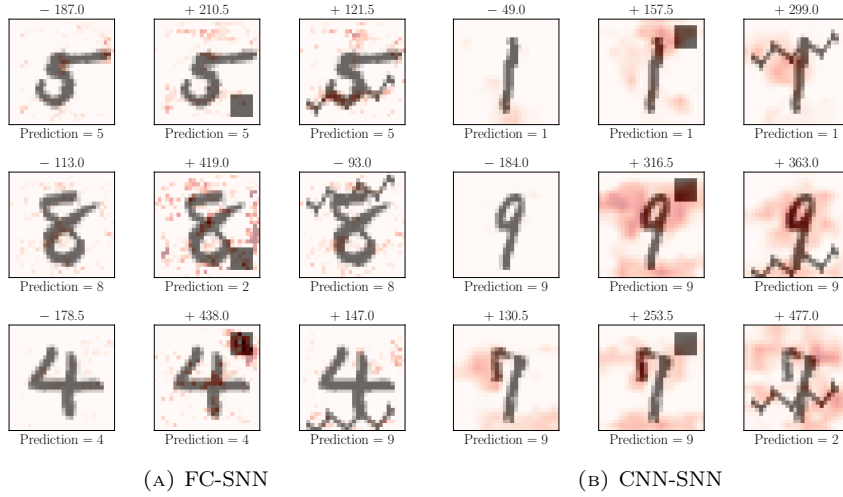


FIGURE 3.13: Heatmaps showing the relevance attributed to different digits (clean and corrupted versions with square and zigzag artifacts) for (a) a FC-SNN architecture; (b) a CNN-SNN architecture. In both cases the number of fully connected SNN layers is $L = 2$. The layout and interpretation of the plots are identical to those in Figure 3.12.

- *The attribution depends on features extracted from the input by the network:* Our detector stringently depends on the features extracted from the input image and that are input to the fully-connected part of the SNN. That is, when it is preceded by a feature extractor like a convolutional layer, our detection capabilities depend on the ability of this layer to learn feature maps that capture characteristics that push the SCP-detector to declare that the image is OoD. Unfortunately, satisfying this necessary condition is not sufficient to guarantee that OoD samples will be detected as such. A false negative could lead to a heatmap that does not highlight characteristics that expose the OoD nature of the misdetected input sample.

On the one hand, as feature maps go through a fully connected network before reaching the layer where the SCP-based detection is performed (the layer prior to softmax), the effect on the latter could be compensated for by the learned weights, as neurons are densely connected and every neuron affects all the neurons of the subsequent layers. On the other hand, due to the way the OoD score is computed, a high intra-cluster variance could eventually mask an OoD sample. As described in Algorithm 1, our method assigns an OoD score computed as the L_1 distance to the closest centroid among the spike count clusters of the predicted class. Hence, if the spike count pattern of the input changes, the closest cluster to the input may vary accordingly, specially if the intra-cluster variance is high. This is not a bad effect overall, as it allows ID samples to be more accurately represented by the centroids. But what could also occur is that the reconstruction of one of the centroids (mentioned in Subsection 3.2.3) of the predicted class coincidentally matches

the reconstruction of the input being tested, masking the OoD characteristics that should be highlighted as relevant.

Notwithstanding the above limitations, results reported in this section support that i) the SCP-based OoD detector performs very competitively with respect to other model-agnostic OoD detection approaches from the literature, with significance as per a Bayesian analysis of the performance differences; and ii) that the relevance attribution technique effectively highlights artificially induced modifications to ID data that a priori should contribute to their identification as OoD by the SCP-based detector.

3.5 Conclusions

This chapter has elaborated on the OoD detection problem when dealing with SNN architectures. Focusing on networks with rate-based encoding schemes and LIF neurons, the main idea of the designed OoD detector is to leverage the spike count patterns produced at the last intermediate layer of the neural model to create archetypes or representatives of those patterns for every class in the dataset at hand. By comparing these archetypes with spike count patterns stimulated by new query samples, the developed detector can effectively discriminate whether they are in- or out-distribution samples. Comparisons are made by using a distance metric, whereas archetypes are represented by the centroids of a cluster space computed over the spike count patterns of training samples by a clustering approach. We further inform the output of the OoD detector with explanations issued by a local relevance attribution method, providing a measure of the features of any given input sample that drive most the output of the detector towards declaring the input as OoD.

Extensive experiments done for image classification tasks over fully-connected and convolutional-based SNN architectures, for gray-scale, color and event-based datasets, have verified that the designed detector achieves a competitive detection performance over other model-agnostic post-hoc OoD detection techniques, which have been tailored to work with spiking-based neural models. Results are specially impressive in the neuromorphic datasets, where our technique consistently outperforms competitors. In addition, experiments with ensembles of OoD detection techniques have showcased that the fusion of the formulated SCP-based detector with a logits-based method attains unrivaled levels of performance when compared to other combinations of techniques. Furthermore, explanations issued by the devised local relevance attribution method excel at identifying artificially inserted perturbations on ID samples, which are highlighted as the parts of the image that are influential for their detection as OoD. Limitations of this technique have been exposed and discussed, showing i) the degradation of the quality of explanations with the increasing depth of the fully connected part of the network; ii) the dependence of the attributed relevance on the input features captured by the input of the SNN model; and iii) the intra-cluster variance of clusters computed from the spike counts of the input data.

Several technical extensions are envisioned for the near future arising from the promising results presented in this chapter. To begin with, efforts will be invested towards mitigating the impact of the aforementioned limitations on the produced heatmaps of the relevance attribution technique. In addition, it will be interesting to take a step beyond the fusion of OoD methods to combine them with the output of uncertainty estimation techniques, which can also give a hint about the unknown nature of a given query to the model. Furthermore, we plan to implement fusion of feature based and logits based techniques in other fields within the OoD detection framework. Finally, integrating this type of models to the OpenOOD repository presented in Section 2.1 would benefit the OoD detection community by eliciting more research in SNNs.

Chapter 4

Out-of-Distribution Detection in Object Detection Tasks

In the previous chapter, we designed an ad-hoc technique for a specific type of networks, SNNs, by leveraging the activation space of these networks, aiming to model the training distribution based on the spike count following the feature extraction part of the model. In this chapter, we will use classical and well-established neural networks used for object detection task, a less explored area by the OoD detection community when compared to image classification.

As discussed in the introduction of this Thesis, ensuring the safety of models involves developing robust systems capable of handling unknown semantics, effectively distinguishing between familiar and unfamiliar objects. In essence, these models must be able to navigate open-world environments by discerning between the background (irrelevant information), known objects (relevant information within the training distribution), and unknown objects (relevant information outside the training distribution).

In this context, substantial efforts have been directed towards the field of Open World Object Detection (OWOD) [100, 101, 102, 103]. The goal here is to develop models capable of detecting unknown objects and incrementally learning new categories over time. Predominantly, research in this area has concentrated on methodologies that require the retraining of a two-stage object detection models, such as Faster R-CNN. However, our focus is on leveraging pre-trained models to create an OoD detector capable of detecting the relevant information outside the training distribution, i.e., the unknown objects.

In this chapter, we first propose a simple method based on activations or feature maps for detecting OoD, similar to the one presented in the previous chapter. We will explore the optimal configurations for this method and compare its effectiveness with other approaches. Additionally, we introduce an unsupervised method that utilizes feature maps to increase the recall of unknown objects.

Subsequently, we test the performance of post-hoc methods [8, 12, 15] adapted to a single-stage object detector. We then assess the results from

a potential fusion or ensemble of the tested methods to validate the conclusions drawn in Chapter 3 concerning the fusion of activation-based and output-based OoD detection techniques. Finally, we compare the presented methods for single-stage object detector with the state-of-the-art of OWOD framework.

The remainder of this chapter is organized as follows: Section 4.1 introduces the OoD techniques tailored for object detection, along with a brief summary of the most influential works in OWOD. Section 4.2 presents our approach for detecting unknown objects¹ in images, beginning with an overview and followed by a detailed description of the method. This section also introduces two techniques aimed at enhancing the overall performance of our approach. The experimental setup is outlined in Section 4.3, and the results are analyzed in Section 4.4. Finally, Section 4.5 offers concluding remarks and discusses potential technical extensions for future research.

4.1 Related Work and Contribution

This section presents the context of existing OoD techniques specifically adapted for the object detection task (Subsection 4.1.1), introduces the key works within the OWOD framework (Subsection 4.1.2) and poses the contributions of this chapter to the OoD literature (Subsection 4.1.3).

4.1.1 Out-of-Distribution Detection

As presented throughout this Thesis and specially in Chapter 2, the OoD detection task aims to identify unknown samples in classification tasks, i.e., samples which semantics are not coming from the training distribution. Many techniques has been proposed in this area, as the ones introduced in Section 2.1 or the ones more deeply explained in Subsection 2.1.1.

While these techniques have been proposed for improving robustness in image classification tasks, methods such as the ones proposed in [104] and [105] are oriented towards object detection. Both are focused on regularizing a two-stage object detection model via the usage of unknowns. In the former, VOS technique is proposed that synthesizes virtual outliers in the feature space without relying on external data to allow the model to better learn the boundaries between known and unknown classes. In the latter, the same objective is pursued but, in contrast, the unknowns are obtained or distilled from videos leveraging contiguous frames and a dissimilarity metric based on the L_2 distance between features of the object proposals.

4.1.2 Open-World Object Detection

The OWOD framework was introduced by [100] to address the limitations of traditional object detection models that assume all object classes are known during training. In the open-world setting, a model is expected to

¹It is important to clarify that in this context and throughout this chapter, the terms “unknown object” and “OoD object” are used interchangeably.

not only detect known objects but also identify unknown objects that do not belong to the training classes. The evaluation protocol introduced was based on the concepts and metrics introduced in [106]. It consists of grouping the classes into sequential tasks, where each task introduces new classes incrementally and the not-yet-included classes are treated as the unknowns to be detected. This protocol is based on the PASCAL VOC [107] and COCO [108] datasets, which are the two most commonly used benchmarks in object detection. All PASCAL classes and data are considered to be the first task. The remaining 60 classes of COCO are grouped into three successive tasks with semantic drifts that are incrementally introduced to model the unknowns arriving at the model’s input. Since then, this evaluation protocol and the datasets involved therein have been adopted by most of the scarce contributions dealing with OWOD.

Additionally, it is important to note that the methods proposed in the SOTA comprise in the majority of the cases adding the “unknown” class to a two-stage object detection model, class that is trained using the pseudo-labels obtained by several heuristics or techniques. Thereupon, each paper includes different upgrades to the training algorithm to improve the performance of the model on the unknown class while preserving the detection capabilities on the known classes.

The solution proposed in [100] called ORE uses contrastive clustering, an unknown-aware proposal network and an energy-based unknown identification to address the challenges of open-world detection. This way ORE is able to predict unknown objects as “unknown” and incrementally learn new classes as their labels become available, without forgetting previously learned classes.

Building on this work, [101] proposed a method called ORDER for addressing the challenges of detecting both known and unknown objects in road scenes. The authors introduce Feature-Mix, which improves unknown object detection by mixing features of known and unknown objects at the feature level. Additionally, they propose a Focal Regression Loss to address intra-class scale variation and improve small object detection. The model is further improved through curriculum learning and is evaluated on the BDD and IDD road scene datasets.

Contrary to previous approaches, the work done in [109] introduces a novel end-to-end transformer-based framework for open-world object detection called OW-DETR. It comprises three dedicated components to address the challenges of OWOD, specifically attention-driven pseudo-labeling, novelty classification, and objectness scoring. The attention-driven pseudo-labeling scheme selects object query boxes with high attention scores as potential unknown objects, which are then used along with ground-truth known objects to train a novelty classifier. The objectness branch aims to effectively separate foreground objects (known and pseudo-unknown) from the background by enabling knowledge transfer from known to unknown classes. Finally, the novelty classification branch is in charge of distinguishing the pseudo unknowns from each of the known classes.

Authors in [102] point out that while the prior OWOD work introduced

the problem definition, the experimental settings were unreasonable, the metrics were incomplete and the methodology was inappropriate. To address these issues, the paper proposes five fundamental benchmark principles to guide the construction of OWOD benchmarks. It also introduces two new metrics (Unknown Detection Recall and Unknown Detection Precision) that focus on assessing the model’s performance from the perspective of unknown classes, rather than just known classes. Furthermore, the authors present a novel OWOD method that includes an auxiliary Proposal Advisor module based on Selective Search algorithm [110] to assist the RPN in identifying accurate unknown proposals by the addition of unsupervised information, and a Class-specific Expelling Classifier to calibrate overconfident predictions and filter out confusing outputs.

The paper [111] goes one step further than previous OWOD approaches an apart from aiming to detect unknown instances, it tries classify them into different unknown classes. First, unknown label-aware proposal and unknown-discriminative classification head are used to detect known and unknown objects. Then, similarity-based unknown classification and unknown clustering refinement modules are constructed to distinguish multiple unknown classes.

A common problem to above introduced approaches is that the module responsible of generating the proposals for the unknown objects, the RPN, is highly biased towards detecting known objects as it is the only ground truth available. To alleviate this effect, [103] substitutes the RPN with random region proposals that prevents the training from being confounded by the limited known object classes. Additionally, the authors introduce a matching score that does not penalize random proposals whose predictions do not match the known classes, encouraging more exploration of unknown objects.

Finally, the authors in [112] address only the initial stage of the OWOD framework, focusing solely on the identification of unknown objects. They evaluate their method using the benchmark they introduced, the Unknown Object Detection (UOD) benchmark, which is derived from two small subsets of the COCO dataset, augmented with annotations for unknown objects. These new annotations are generated by them and their team. The evaluation protocol involves assessing the models exclusively on one task within these two subsets, without any incremental addition of new classes to the model. The authors also propose the Unknown Sniffer model, which introduces the Generalized Object Confidence (GOC) score to avoid improper suppression of unknowns. Similarly, a negative energy suppression loss is used to further suppress non-object samples in the background. To address the challenge of obtaining the best bounding box for unknown objects during inference, the paper presents a graph-based determination scheme to replace hand-designed NMS post-processing.

4.1.3 Contribution

Upon reviewing the current SOTA, it becomes evident that there is a clear trend towards using two-stage models that undergo retraining to acquire

the capability to detect unknown objects. Consequently, it is imperative to analyze the performance of OoD techniques for one-stage models that do not require retraining of the object detection model.

In our research, we align our approach with the work of [112], so we focus exclusively on the first task of OWOD. As Figure 1.1 describes in the introduction of this Thesis, it only involves evaluating the ability to detect unknown objects, without subsequently retraining the model with these newly identified classes. Our contributions in this chapter can be summarized as follows:

- *Exploring the effectiveness of classic post-hoc OoD techniques on one-stage object detectors.* We investigate the performance traditional post-hoc OoD methods perform when applied to one-stage object detection models, which are generally more efficient but less studied in this context.
- *Proposing a simple OoD technique based on feature maps that does not require retraining.* We aim to introduce a straightforward and efficient method for OoD detection that leverages feature maps. This technique is designed to operate without the need for retraining the original object detection model, thus maintaining the efficiency of the one-stage detector.
- *Proposing techniques for enhancing detection of potential unknown objects using feature maps.* We experiment with various methods to improve the detection of unknown objects by utilizing feature maps. This includes refining feature extraction and leveraging advanced techniques to better identify and classify OoD objects.
- *Examining the potential of combining methods to robustly identify unknown objects.* We explore the benefits of combining or fusing different OoD detection methods. We want to test which of the methods combined in the form of a detection ensemble provide complementary information, so the fusion of detection issued by such detectors can yield more robust OoD detection capabilities for the overall model.

This research contributes to the understanding and development of more effective OoD detection techniques for one-stage object detectors, potentially leading to more robust and adaptable models in open-world scenarios.

4.2 Open-World Object Detection by Feature Map Characterization

In this section, we introduce our algorithm, based on the usage of the feature maps and hence termed the FMap detector. Subsection 4.2.1 presents the general workflow of the method, while Subsection 4.2.2 offers a detailed, in-depth description. Figure 4.1 illustrates the algorithm’s operation. In

addition, Subsection 4.2.3 elaborates on the benefits of employing Supervised Dimensionality Reduction (SDR) with our detector. Finally, Subsection 4.2.4 outlines an approach to enhance the capabilities for identifying unknown objects without penalizing the performance in detecting known objects.

4.2.1 High-level description

Here we present the comprehensive overview of our algorithm. Similar to the previous chapter, we aim to characterize the activation space at the output of a single-stage object detection model, i.e., we characterize the feature maps.

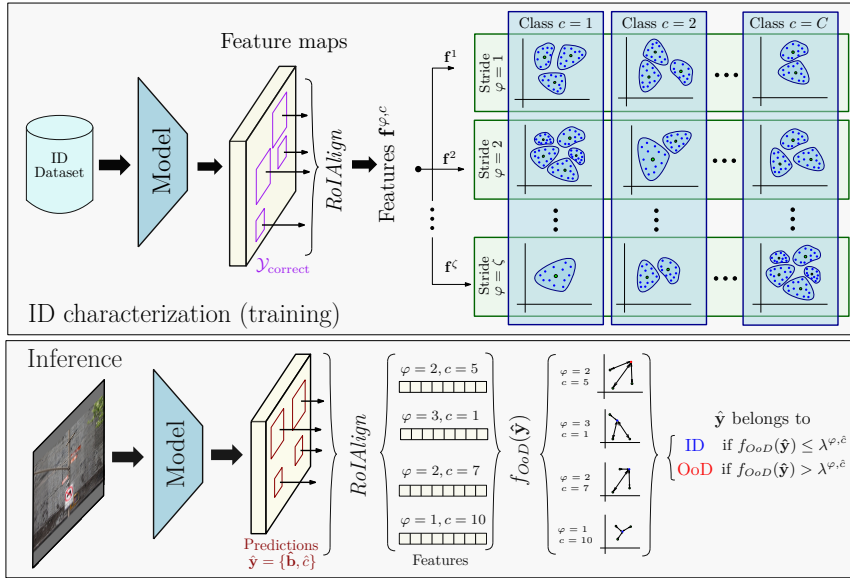


FIGURE 4.1: Description of the operation of the FMap detector. In the inference part, an example for the stride (φ) and class (c) values is displayed.

Modern object detection models employ a FPN to enhance their ability to detect objects at various scales. Therefore, the characterization of activations must account for this feature. In the case of two-stage detectors, as discussed in Chapter 2, Section 2.3, feature maps of different strides are merged and standardized to have a uniform number of output channels or dimensions after the FPN. In contrast, mainstream one-stage detectors do not perform this standardization, resulting in each stride having unique dimensions and producing predictions independently.

Consequently, we characterize each stride separately within its own space, as each prediction emanates solely from one of the strides. Therefore, the features from different strides are processed by distinct layers and may not share the same number of dimensions or semantic meaning. The workflow of our method is defined as follows.

First, we iterate over the ID data—typically the training dataset—to collect all the extracted feature maps of the correctly predicted samples, noting the stride from which each prediction originates. Subsequently, for each class within each stride independently, we cluster the embedding space and aggregate the features of each cluster into a centroid, e.g., by computing the mean of every feature within a cluster. This yields multiple centroids for each class in each stride, serving as the representatives of the concepts in the embedding space for that class.

The score for each query sample is defined as the minimum distance of the instance’s features to the centroids of the predicted class in the respective stride (since each prediction originates from only one stride). Based on this definition, to determine whether a query sample is an OoD instance, we must first establish a threshold for the score.

To this end, we iterate again over the ID data to obtain the scores for the correctly predicted instances. Each class in each stride will have its own ID score distribution and corresponding threshold.

Finally, we define the threshold such that we achieve a desired TPR for each class in each stride independently, ensuring that a specified number of ID samples are correctly classified as ID.

4.2.2 Low-level description

This section provides a more detailed description of the FMap detector, along with a schematic summary in Algorithm 4. As previously mentioned, the first step is to characterize the ID by creating the representative features or centroids of each class in each stride by making use of clustering techniques.

Hence, we start extracting the features from the ID samples by iterating over the training dataset $\mathcal{D}_{tr}$ and collecting the predictions. The dataset consists of images $\mathbf{x} \in \mathbb{R}^D$ with objects represented by targets $\mathbf{y} = \{\mathbf{b}, c\}$, where $\mathbf{b} \in \mathbb{R}^4$ denotes the bounding box coordinates and $c \in \{1, 2, \dots, C\}$ the class or category of the object. C represents the number of classes in $\mathcal{D}_{tr}$. Each image can contain one or multiple targets (labeled objects). Among all the predictions $\hat{\mathbf{y}} = \{\hat{\mathbf{b}}, \hat{c}\}$ generated by the model, only the correctly predicted ones are considered for feature extraction, i.e., $\mathcal{Y}_{\text{correct}} = \{\hat{\mathbf{y}} \mid \hat{c} = c \text{ and sufficient } \text{IoU}(\hat{\mathbf{b}}, \mathbf{b})\}$, where IoU denotes Intersection over Union. The features $\mathbf{f} \in \mathbb{R}^{D_\varphi}$ are extracted from the feature maps $\mathbf{v}$, where D_φ represents the dimensionality of the stride indexed by $\varphi \in \{1, 2, \dots, \zeta\}$ being ζ the total number of strides (line 5 of Algorithm 4).

To perform the extraction of the features corresponding to each prediction, we utilize the *RoIAlign* operation as described in [113]. The inputs are a bounding box $\mathbf{b}$, the desired output size (height and width) for the features $\mathbf{f}$ of the box, the complete feature maps $\mathbf{v}$, and the ratio between the spatial resolution of the bounding boxes (height and width of the original image) and the spatial resolution of the feature maps (height and width of them). The operation outputs the desired features $\mathbf{f}$ specific to the bounding box $\mathbf{b}$ out of the whole feature maps $\mathbf{v}$. It is crucial to emphasize that

each prediction originates from a specific stride, thereby implying that its features possess the dimensionality of that specific stride. In particular, the number of channels are maintained after *RoIAlign* operation.

Following the described process, we obtain a collection of features $\mathbf{f}^{c,\varphi}$ for each stride φ and each class c . These features are independently clustered using a clustering algorithm to form an arbitrary number $M^{\varphi,c}$ of clusters $\{\mathbf{Q}^{m,\varphi,c}\}_{m=1}^{M^{\varphi,c}}$ (line 10 of Algorithm 4). Each class in each stride is clustered separately. The number of clusters per class can be set as a hyperparameter or determined by optimizing a clustering performance metric. Next, using an aggregation function $f_{agg}(\cdot)$, we aggregate the features of all samples in each cluster and obtain the centroids $\mathbf{f}_{\odot}^{m,\varphi,c}$, which can be regarded as the representative features of each class c and each stride φ .

Algorithm 4: FMap Detector.

Data: Training dataset $\mathcal{D}_{tr}$, Validation dataset $\mathcal{D}_{val}$, Number of classes C , Number of strides ζ

Result: Detector ready to identify unknown objects

```

1 //ID definition stage
2 Collect the features from correct predictions in  $\mathcal{D}_{tr}$ :
3 for  $\mathbf{x} \in \mathcal{D}_{tr}$  do
4   | Collect correct predictions  $\hat{\mathbf{y}} \in \mathcal{Y}_{\text{correct}}$ 
5   | Extract their features  $\mathbf{f}^{\varphi,c} \in \mathbb{R}^{D_\varphi}$  using RoIAlign
6 end
7 Cluster features for each class  $c$  and each stride  $\varphi$ :
8 for  $c \in \{1, 2, \dots, C\}$  do
9   | for  $\varphi \in \{1, 2, \dots, \zeta\}$  do
10    | Obtain clusters  $\{\mathbf{Q}^{m,\varphi,c}\}_{m=1}^{M^{\varphi,c}}$  with a clustering algorithm
11    | for  $m \in \{1, \dots, M^{\varphi,c}\}$  do
12    |   | Compute centroids  $\mathbf{f}_{\odot}^{m,\varphi,c}$  using aggregation function
13    |   |  $f_{agg}$ 
14    | end
15 end
16 //Thresholds definition stage
17 Compute score distribution for  $\mathcal{D}_{val}$  using Eq. (4.1):
18 for  $\mathbf{x} \in \mathcal{D}_{val}$  do
19   | Compute scores  $f_{OoD}(\hat{\mathbf{y}})$  for all correct predictions
20 end
21 Determine thresholds  $\lambda^{\varphi,c}$  to achieve a certain TPR with the
   | obtained score distribution
22 //Inference
23 Classify predictions  $\hat{\mathbf{y}}$  from query samples  $\mathbf{x}_{query}$  into ID or OoD
   | using Eq. (4.2)
```

Now we can define the score function $f_{OoD}(\cdot)$ of our method as shown in Eq. (4.1), which assigns an score to every prediction $\hat{\mathbf{y}}$ based on the

features $\mathbf{f}^{\varphi, \hat{c}}$, calculated as the minimum distance $d(\cdot)$ between the features of the prediction and the centroids of the clusters of the predicted class and corresponding stride:

$$f_{OoD}(\hat{\mathbf{y}}) = \min\{d(\mathbf{f}^{\varphi, \hat{c}}, \mathbf{f}_{\odot}^{m, \varphi, c})\}_{m=1}^{M^{\varphi, \hat{c}}}. \quad (4.1)$$

Finally, we compute the score distribution of our ID and impose a threshold λ to determine whether a query sample is considered as ID or an OoD sample. To this end, we iterate the ID (this time typically the validation split $\mathcal{D}_{val}$), obtaining the scores of all the correct predictions $\hat{\mathbf{y}}$ using the above defined score function $f_{OoD}(\hat{\mathbf{y}})$ (line 19 of Algorithm 4). This process allows us to set the threshold $\lambda^{\varphi, c}$ for each class c and each stride φ to achieve the desired TPR, i.e., the percentage of ID samples that will be correctly classified as ID.

At inference time, each query sample $\mathbf{x}$ will be processed, obtaining some predictions $\hat{\mathbf{y}}$, that will be either classified as ID or OoD by the following expression:

$$\hat{\mathbf{y}} \text{ belongs to } \begin{cases} \text{ID} & \text{if } f_{OoD}(\hat{\mathbf{y}}) \leq \lambda^{\varphi, \hat{c}}, \\ \text{OoD} & \text{if } f_{OoD}(\hat{\mathbf{y}}) > \lambda^{\varphi, \hat{c}}. \end{cases} \quad (4.2)$$

4.2.3 Supervised Dimensionality Reduction

Given that the algorithm operates on feature maps that come from the feature extraction part of an object detection model, it typically attains working with high-dimensional features. what may hinder the potential benefits of our method due to the curse of dimensionality.

To alleviate this effect, we propose employing existing SDR techniques on the extracted features $\mathbf{f}$, so that our algorithm is able to distinguish between ID and OoD features more accurately. SDR techniques combine unsupervised dimensionality reduction algorithms with supervised loss functions to produce low-dimensional representations of the data while preserving clear class boundaries.

The SDR method can be represented as a model or trainable function g , dependent on parameters θ and that maps the features from dimensions $\mathbb{R}^D$ to $\mathbb{R}^{D'}$. In the case of single-stage object detectors, as the dimensionality of the features varies depending on the stride φ , we need to train a distinct function for each stride in the object detection model. Hence, the function for the SDR method can be expressed as follows:

$$g_{\theta}^{\varphi}(\mathbf{f}^{\varphi}) : \mathbb{R}^{D_{\varphi}} \rightarrow \mathbb{R}^{D'_{\varphi}}. \quad (4.3)$$

To train parameters θ , a labeled dataset of features must be provided for each stride φ , where the labels correspond to the classes to which the features belong. In fact, this labeled dataset is obtained during the feature collection from the ID data (lines 6 and 7 of Algorithm 4). Hence, at that point, right before creating the clusters, we can train each function g_{θ}^{φ} with those extract features. Subsequently, we use them to transform the features to the new dimensionality, specifically $\mathbf{f}^{\varphi} \rightarrow \mathbf{f}'^{\varphi} : \mathbf{f}'^{\varphi} \in \mathbb{R}^{D'_{\varphi}}$.

From this point onward, the FMap detector operates as previously described, with the key difference that each time the features of a prediction are extracted, they are transformed to the new dimensionality using the corresponding function $g_{\theta}^{\varphi}(\mathbf{f}^{\varphi})$, what happens in lines 19 and 23 of Algorithm 4.

4.2.4 Enhanced Unknown Object Localization

One key aspect of the usage of OoD techniques for unknown object detection is their reliance on the predictions generated by the single-stage model, which in turn depend on the inference confidence threshold, a user-selected hyperparameter. The inference confidence threshold represents the minimum confidence value a prediction must achieve to be considered valid and subsequently be outputted by the model.

Specifically, these OoD methods can only identify an unknown object if it has been incorrectly predicted as one of the known classes, at which point the algorithm must detect this prediction as an OoD instance or unknown object.

To address and mitigate this dependency, we propose an unsupervised algorithm to generate proposals for potential unknown objects, i.e., the idea is to improve the unknown recall capabilities. This algorithm, which we have termed Enhanced Unknown object Localization (EUL), is based on the hypothesis that high-resolution feature maps may contain information about unknown objects that the model does not ultimately output as predictions. The overall workflow is as follows:

1. *High resolution feature map extraction.* To accurately locate possible unknown objects in the image, we extract the feature maps from the stride with higher resolution.
2. *Feature map information aggregation.* Next we must condense the information from all channels of the feature maps into a saliency map, i.e., an image with the same resolution but only one channel which highlights areas of the image that, in this case, likely contain objects.
3. *Saliency map binarization.* Then we convert the saliency map $\mathbf{h}$, into a binary format. This involves setting a threshold value; pixels with saliency values above this threshold are set to one (indicating potential object presence), while those below are set to zero (indicating background). Multiple thresholds can be selected in order to obtain multiple binarized images to for search for objects of different sizes. The selection of the thresholds can be done manually as a hyperparameter or automatically using methods like Otsu thresholding [114].
4. *Region and Bounding box calculation.* After applying the threshold, use connected-component labeling to identify contiguous regions, each representing a potential object in the original image. For each of these regions, compute the bounding box that encompasses it by finding the minimum and maximum coordinates within each connected component.

Since the bounding boxes are calculated at the spatial resolution of a feature map, they need to be rescaled to the original image size.

5. *Unknown object proposals filtering.* The previous operations generate a great quantity of proposals, so they must be filtered to obtain meaningful ones. This is done using a method for ranking the proposals and selecting the top-k better ones.

In the process of ranking proposals, features are first extracted through the *RoIAlign* operation. Subsequently, the distances between these features and all clusters corresponding to each class within the utilized stride (the one with the highest resolution, as previously discussed) are computed. These distances are then consolidated into a single metric using their entropy. Proposals exhibiting the lowest entropy are preferentially retained, based on the hypothesis that lower entropy correlates with a higher probability of identifying a genuine object.

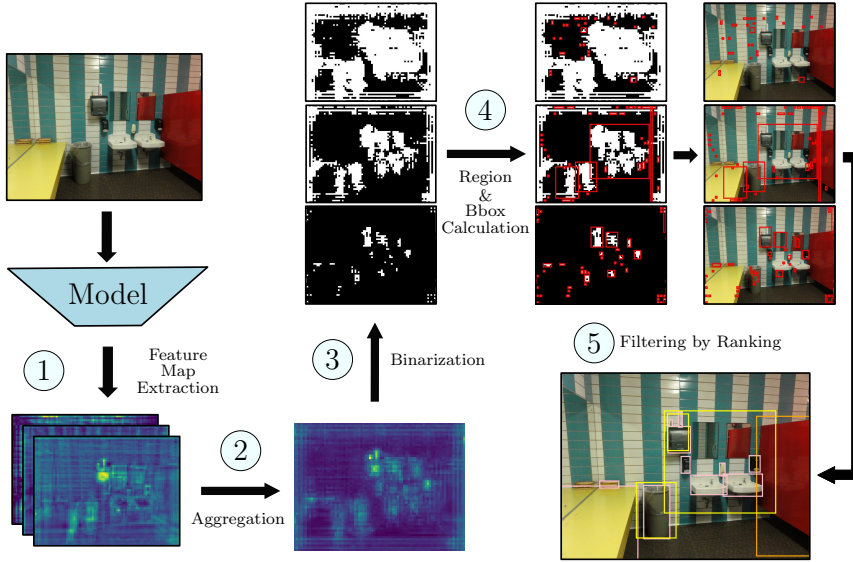


FIGURE 4.2: Enhanced unknown object localization graphically depicted with an example. The red boxes in the 4th step are bounding box proposals. In the final image after 5th step, the orange box is an unknown detected by the basic FMap algorithm, in yellow are the boxes selected by EUL, and finally the pink boxes are the unknown object annotations.

4.3 Experimental Setup

We guide the experimentation towards answering four research questions, namely:

- RQ1: Which is the optimal configuration for our detection method?

- RQ2: How does our method perform when compared to output-based post-hoc OoD detection methods in single-stage object detectors?
- RQ3: Does a fusion of feature-based methods with logits-based methods outperform other potential ensemble configurations?
- RQ4: How does the performance of unknown object detection algorithms implemented on single-stage models compare to that of the SOTA?

4.3.1 Evaluation Benchmark

We base our evaluation on the Unknown Object Detection benchmark presented by [112]. It follows previous works' convention of using PASCAL VOC [107] as training dataset, that contains annotations of 20 object categories. For testing, it uses two subsets of COCO [108], namely COCO-OoD and COCO-Mix. The former contains only objects that are not known and the latter mixes both known and unknown instances. In both cases, unknowns are instances of the other 60 classes COCO contains plus new unknown instances labeled by [112], with images been handpicked. More details of this benchmark can be found in their paper. The number of known and unknown objects can be found in the Table 4.1. This benchmark is used in the experiments of all research questions to derive the answers.

Dataset	Images	Known Objects	Unknown Objects
COCO-OoD	504	-	1655
COCO-Mix	897	2533	2658

TABLE 4.1: Number of images and annotated known and unknown objects per subset in the UOD benchmark.

4.3.2 Evaluation Metrics

To evaluate the models in known object detection, the most common practice in the literature is to employ the mean Average Precision (mAP). It provides a single value that measures the average precision across all categories, facilitating the comparison of different models. Essentially, the mAP is calculated by computing the AP for each object category separately and then taking the mean. The AP is a combination of the precision-recall and the Intersection over Union (IoU) metrics.

Precision quantifies the accuracy of the model's positive predictions, whereas recall assesses the percentage of actual positives that the model successfully detects. Typically, a trade-off exists between precision and recall; increasing object detection (enhancing recall) may lead to a rise in false positives (reducing precision). The AP metric addresses this balance by integrating the precision-recall curve, which graphs precision against recall at varying confidence thresholds.

The IoU is defined as the ratio of the intersection or overlap area to the union area of the predicted bounding box and the ground truth bounding

box, as shown in Figure 4.3. Hence, it is bounded to the range $[0, 1]$. It is used to assess the performance of the model on the localization of the objects, that is, measures the quality of the bounding boxes.

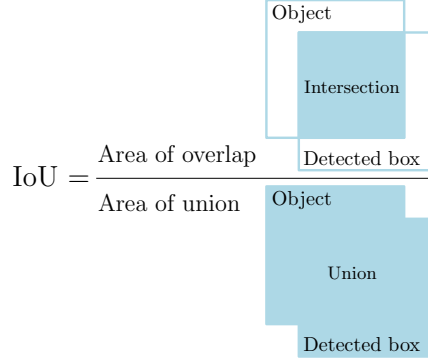


FIGURE 4.3: Intersection over Union (IoU) explained.

There are two main ways to compute the AP. The one used in **PASCAL VOC** [107] challenge is to compute the average of the precision at 11 different recall levels (from 0 to 1 with increments of 0.1), being the correct predictions the ones with an IoU of 0.5 or higher. For the **COCO** [108] challenge, the number of recall interpolation points was 101—the increments are of 0.01—and to better assess the performance of the model at different localization qualities, different IoU thresholds are used. In this Thesis, as in the literature related to OWOD, the **PASCAL VOC** method is used.

Regarding the metrics for evaluating the performance in OoD or unknown object detection, we follow the conventions of [112] and other previous works. TP_u is for the true positive proposals of unknown objects (i.e., detected OoDs), FN_u for false negative proposals, and FP_u for false positive proposals. The metrics are:

- The Unknown Average Precision (U-AP) is the average precision computed only for the unknown class.
- The Recall Rate (U-REC) and Precision Rate of Unknown objects (U-PRE) are defined as:

$$\text{U-REC} = \frac{\text{TP}_u}{\text{TP}_u + \text{FN}_u}, \quad (4.4)$$

$$\text{U-PRE} = \frac{\text{TP}_u}{\text{TP}_u + \text{FP}_u}. \quad (4.5)$$

- The Unknown F1-Score defined as the harmonic mean of U-PRE and U-REC, i.e:

$$\text{U-F1} = \frac{2 \cdot \text{U-PRE} \cdot \text{U-REC}}{\text{U-PRE} + \text{U-REC}}. \quad (4.6)$$

- The Absolute Open-Set Error (A-OSE), by [115], is employed to report the count of unknown objects that are wrongly classified as any of the known classes.
- Wilderness Impact (WI), proposed in [106], is defined as:

$$\text{WI} = \frac{\text{Precision in closed-set}}{\text{Precision in open-set}} - 1, \quad (4.7)$$

and is used also to characterize the case that unknown objects are confused with the known.

4.3.3 Implementation Details

We use Ultralytics [116] library and employ YOLOv8 as the One-Stage object detector. The number of strides ζ is 3. We trained the model from scratch for 300 epochs using PASCAL VOC as training dataset. More details about the training can be found in the GitHub repository with the code (https://github.com/aitor-martinez-seras/OoD_in_Object_Detection.git).

RoIAlign and aggregation operations. We use average as the operation to perform the aggregation of the features ($f_{agg}(\cdot)$) in vanilla FMap method. For *RoIAlign*, it is necessary to choose the height and width of the resultant features. We opt for a one-pixel (1×1) output size, hence every prediction's features will be of size equal to the number of channels in the corresponding stride.

Determining the thresholds for the OoD methods. As our method needs to determine a threshold for each class in each stride, we need sufficient samples in each case for the thresholds to be representative of the ID. Therefore, instead of only using the validation split from the training dataset, PASCAL VOC in this case, we use the samples of both the train and validation splits to compute the thresholds. Furthermore, the TPR used to determine the thresholds $\lambda^{\varphi, c}$ is set at 95%, a common choice in the literature.

Distance metrics. The scoring function of FMap method, i.e., the $f_{OoD}(\cdot)$ function, uses a distance computation to assess the dissimilarity between features. We test L_1 , L_2 and *Cosine* distances for this purpose.

Clustering algorithms. Our method can be combined with any clustering technique, so we have tested a few in this research, namely *KMeans* and *HDBSCAN*. To optimize them, we make a grid search of the different hyperparameters and select the configuration that attains the best silhouette score, individually for each class in each stride. For the case of *KMeans*, as the optimization algorithm tends to create 2 to 3 clusters in each case, we have also tested to force the algorithm to create 10 clusters. In addition, we use the *One* method, basically assuming we have only one big cluster.

Supervised dimensionality reduction. To implement it, we have chosen the ivis framework presented in [117]. Essentially, the technique

works by utilizing a siamese network architecture trained by a variant of the standard triplet loss. After a grid search, the hyperparameters selected are an output embedding dimension of 32 ($D'_\varphi = 32$) and a k value of 15, that controls the number of nearest neighbors to retrieve for each point. The same output dimension is selected for the three strides. For more details of the workings of the algorithm, refer to the paper [117].

Enhanced Unknown object Localization. For the feature map aggregation, we compute the mean absolute deviation. For the binarization, a fast variation of the otsu thresholding [114] method that works in a recursive fashion is used.

4.3.4 Fusion Strategies

When integrating two techniques within a fusion framework, it is crucial to establish a decision criterion to resolve discrepancies between the methods. In the previous chapter, we employed the AND criterion, wherein a sample is deemed ID if at least one method classifies it as such. This approach typically results in increased precision (U-PRE) but decreased recall (U-REC) for unknown objects.

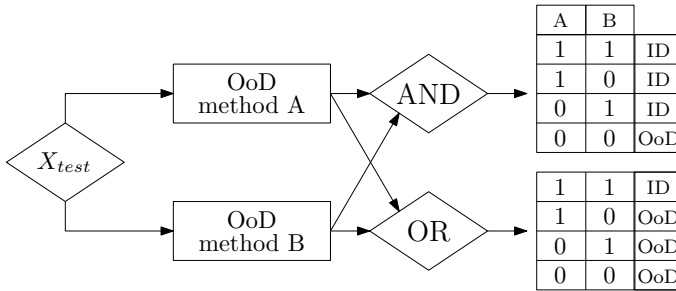


FIGURE 4.4: Illustration of the AND and OR fusion strategies. A decision of ‘1’ indicates that the method identifies the prediction as ID, whereas ‘0’ indicates the contrary.

This chapter explores alternative strategies, specifically the OR and the SCORE strategies. The OR criterion is straightforward and operates contrary to the AND criterion: a sample is classified as OoD if either method considers it OoD, thereby potentially increasing the recall at the expense of precision for both known and unknown objects. Figure 4.4 delineates the operational mechanics of both the AND and OR strategies.

Conversely, the SCORE strategy is designed to balance these objectives, necessitating preliminary calculations during the definition stage of the ID criteria for the techniques involved. The essence of the SCORE strategy lies in each method computing a fusion score via a designated fusion function, $f_{fusion}(\cdot)$. This score quantifies the extent to which a sample is considered ID, based on the deviation of the OoD score for a given OoD technique from the corresponding threshold. The fusion score ranges from -1 to 1, where 1 signifies a strong likelihood of the sample being ID, and -1 indicates a strong likelihood of the sample being OoD. These scores are

aggregated to produce a final fusion score that determines the sample’s classification:

$$f_{fusion}(\hat{\mathbf{y}}) = f_{OoD}^A(\hat{\mathbf{y}}) + f_{OoD}^B(\hat{\mathbf{y}}), \quad (4.8)$$

$$\hat{\mathbf{y}} \text{ is classified as } \begin{cases} \text{ID} & \text{if } f_{fusion}(\hat{\mathbf{y}}) > 0, \\ \text{OoD} & \text{if } f_{fusion}(\hat{\mathbf{y}}) \leq 0, \end{cases} \quad (4.9)$$

where the superscripts A and B represent the scores from the respective fused methods.

Figure 4.5 illustrates the calculation of the fusion score for feature maps and logits-based methods. The process involves collating the minimum and maximum ID values, which subsequently define the boundaries of the piece-wise function $f_{fusion}(\cdot)$. Depending on whether the OoD score for a particular method falls below or above the threshold, the corresponding segment of the piece-wise function is utilized to compute the fusion score.

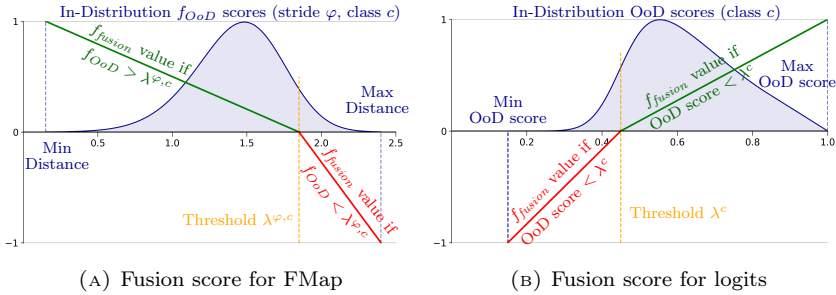


FIGURE 4.5: Illustration of the SCORE strategy. Each method computes a fusion score for $f_{fusion}(\cdot)$ using the piece-wise functions depicted in green and red. Values outside the defined range are clipped to the minimum and maximum values, respectively.

4.3.5 Post-hoc Methods

The post-hoc methods that are implemented in this chapter are the ones meeting the criteria of being directly applicable to any type of classification branch of a model, without any ambiguity of how they should be implemented.

In this sense, algorithms based on extracting the output of the penultimate layer of the network and post-processing are not implemented. This is in virtue of the fact that, modern one-stage detectors, specially those in the YOLO family, extract multidimensional features from the FPN therefore rendering these methods not directly applicable to these architectures. Instead, post-hoc techniques like MSP [8], Energy [15] and ODIN [12], that only rely on the output of a classification branch, meet the criteria of being directly applicable to the one-stage object detector. In case of ODIN, the input perturbation is not applied as it is not clear how it should be implemented when multiple predictions are issued per image. For more details about the configuration of these methods, refer to Section 2.1.1.

Given our choice to utilize YOLOv8, it is important to note that this model does not produce a vector of class probabilities for each bounding

box prediction. Instead, it outputs a vector of class “confidences”, obtained by applying the sigmoid activation function in the final layer. For the purposes of this study, we have used the raw output of this final layer, prior to the application of the sigmoid activation function, as the foundation for the implemented post-hoc techniques.

4.4 Results and Analysis

This section presents the results of the experiments conducted to address the research questions posed. First, we aim to determine which configuration of our detection technique performs best. Then, we compare it to existing post-hoc methods. Subsequently, we combine both distance-based and post-hoc methods to ascertain if the ensemble of feature-based and logits-based methods leads to better results than when used separately. Finally, we compare our findings against state-of-the-art methods.

4.4.1 Preamble

Given that the goal of this chapter is to develop a single-stage object detection model with open-world detection capabilities, we focus on finding the optimal configuration by balancing the performance in detecting both known and unknown objects. Hence, we try to balance the mAP as the representative value for the performance in the detection of known instances and the U-F1 as the indicator of the method’s effectiveness in recognizing unknown objects. Specifically, we use the sum of unknown F1 (hereafter referred to as U-F1 in the text and $U-F1_{SUM}$ in the plots) obtained on both COCO-0oD and COCO-Mix, while mAP is calculated only from COCO-Mix since COCO-0oD includes only annotations for unknowns objects.

Building on these methodological choices, it is important to highlight the insight presented in Subsection 4.2.4, where we emphasize the significance of the inference confidence threshold. Delving deeper, this threshold can be regarded as a variable that calibrates the balance between the model’s known and unknown object detection capabilities.

Thus, during the performance assessment of the single-stage model (RQ1, RQ2 and RQ3), we aim to identify the Pareto fronts that arising from this trade-off, which is regulated by the inference confidence threshold. They are going to be represented in plots, with the different configurations of cluster methods and distance metrics distinguished by unique markers and colors. **For each configuration, several inference confidence thresholds are tested and plotted.** The optimal configurations, i.e., the non-dominated ones, are highlighted with red circles and connected by a red dashed line. Additionally, their A-OSE values are displayed alongside each red circle. Finally, the results on all the metrics of the best configurations for every research question are available in Appendix B.

4.4.2 RQ1: Which is the optimal configuration for our detection method?

To address this inquiry, we first identify the optimal or non-dominated configurations for the FMap method when applied independently, and in conjunction with SDR and EUL techniques. Subsequently, we compare these optimal configurations to delineate the final Pareto front for FMap.

Vanilla FMap. The initial analysis involves the baseline (vanilla) version of the FMap method, examined in conjunction with various cluster methods and distance metrics. The comparative results are depicted in Figure 4.6.

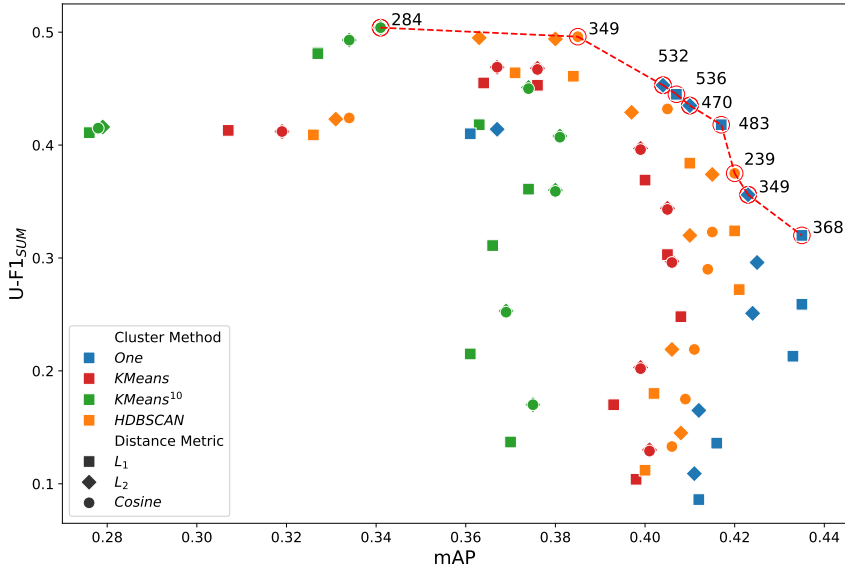


FIGURE 4.6: mAP and U-F1_{SUM} pareto front for vanilla FMap of the tested distance metrics (represented by marker types) and cluster methods (indicated by color). For each configuration, various inference confidence thresholds are represented. Refer to Subsection 4.4.1 for further explanations.

At first glance, it is evident that the choice of cluster method (indicated by color) significantly impacts the performance with respect to known objects (mAP). Generally, aggregating all features into a single large cluster per class and stride (denoted as the *One* cluster method, shown in blue) results in higher mAP and lower U-F1 scores. Conversely, configurations that generate multiple clusters (thereby increasing the number of centroids for comparison) typically exhibit reduced mAP scores but enhanced performance in detecting unknown objects. Notably, increasing the cluster count from 2 to 3, as commonly achieved with both *KMeans* (in red) and *HDBSCAN* (in orange) — as detailed in Subsection 4.3.3 — to deliberately forcing 10 clusters in *KMeans*¹⁰ (in green), usually leads to a further decrease in mAP while marginally improving U-F1.

The aforementioned conclusions are corroborated by examining the non-dominated configurations. Configurations yielding the highest mAP predominantly utilize the *One* cluster method, while those achieving larger U-F1 scores generally employ a greater number of clusters. Notably, the optimal U-F1 score is achieved using a method based on *KMeans*¹⁰. Furthermore, with respect to the A-OSE metric (which quantifies the number of unknown instances erroneously classified as known objects) the data indicates a clear trend: an increase in the number of clusters correlates with a reduction in classification errors for unknown instances.

Additionally, an analysis of the impact of distance metrics reveals significant findings. The L_1 and L_2 distance metrics typically enhance the detection of known objects. Conversely, the *Cosine* distance metric demonstrates superior performance in terms of robustness and the recognition of unknown objects, as evidenced by improved U-F1 and A-OSE scores.

FMap with Supervised Dimensionality Reduction. Next we apply a SDR algorithm to lower the dimensions of the features in order to reduce the impact of the curse of dimensionality.

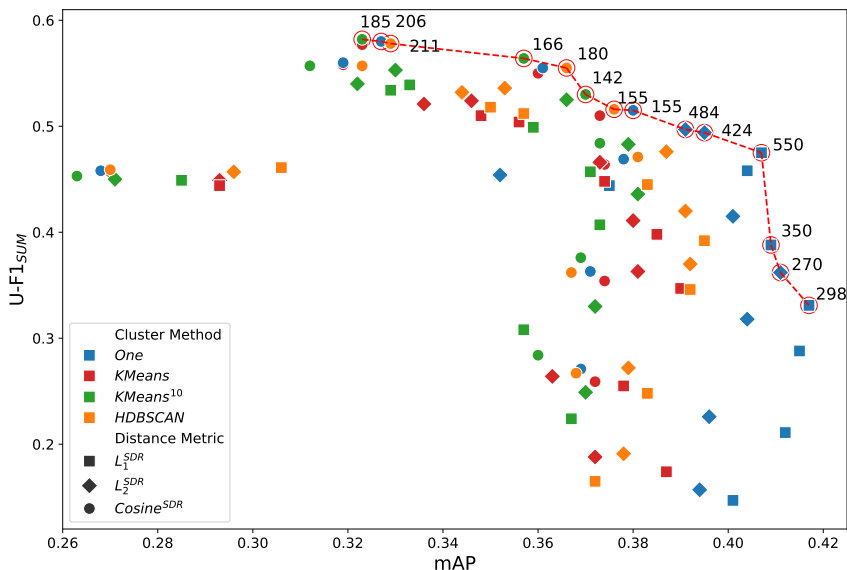


FIGURE 4.7: mAP and U-F1_{SUM} pareto front for FMap with SDR of the tested distance metrics (represented by marker types) and cluster methods (indicated by color) for the FMap with SDR. For each configuration, various inference confidence thresholds are represented.

Similar insights concerning the impact of distance metrics and clustering methods on performance are elucidated in Figure 4.7. Generally, employing the *Cosine* distance and increasing the number of clusters correlates with larger U-F1 scores. In contrast, using L_1 and L_2 metrics and only one cluster leads to increased known object performance and slightly reduced U-F1 scores. Another key finding is that the robustness *Cosine* is providing in reducing the unknown objects wrongly classified as known

classes becomes even more apparent when utilizing SDR methods. This is showcased by the lower A-OSE values.

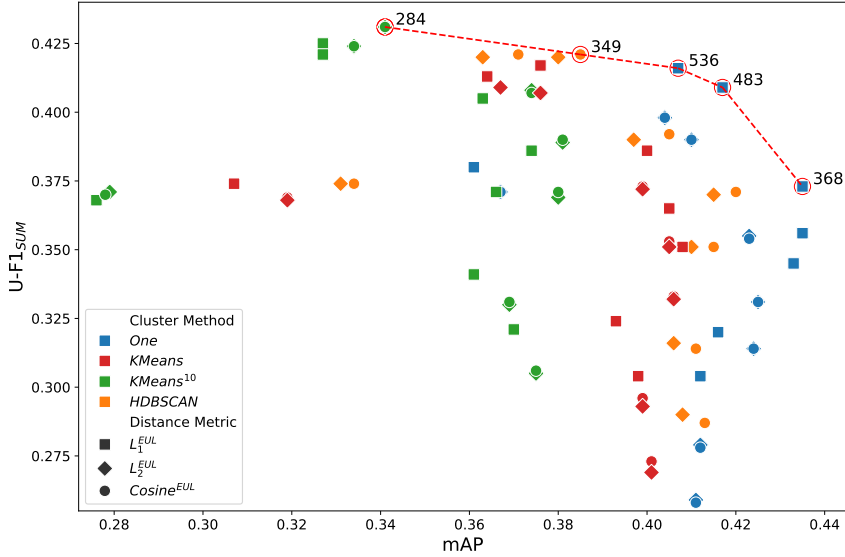


FIGURE 4.8: mAP and $U-F1_{SUM}$ pareto front for FMap method with EUL of the tested distance metrics (represented by marker types) and cluster methods (indicated by color). For each configuration, various inference confidence thresholds are represented.

FMap with Enhanced Unknown localization. Finally, we apply the algorithm presented in Subsection 4.2.4 to boost the unknown localization capabilities of the one-stage object detector.

Consistent with earlier results, Figure 4.8 illustrates that the *Cosine* distance, when paired with a clustering technique that generates more than one cluster, consistently achieves the best U-F1 scores. Meanwhile, the highest mAP is obtained using the L_2 metric with a single cluster. Nevertheless, in this scenario, the variance in unknown detection performance between the poorest and best methods is significantly smaller, narrowing from 18% with vanilla implementation and 25% when SDR is applied, to merely 5% under the EUL configuration. In contrast, the performance disparity in detecting known objects remains consistent across all three scenarios.

Comprehensive comparison. Lastly, we evaluate the optimal combinations of cluster method, distance metric, and inference confidence threshold across all three versions of the algorithm (the non-dominated configurations from each). This comparative analysis is depicted in the Figure 4.9. The results clearly demonstrate a general trend: the FMap method enhanced with SDR techniques consistently excels in detecting unknown objects, also achieving excellent A-OSE values compared to other configurations. Conversely, the basic or vanilla version of the FMap detector more effectively maintains the original known object detection capabilities while delivering reasonable U-F1 scores. Finally, the addition of

EUL technique is only beneficial in when the specific combination of cluster method, distance metric and confidence threshold yield high mAP but low U-F1. In such scenarios, EUL is capable of enhancing the unknown detection capabilities without sacrificing the mAP performance.

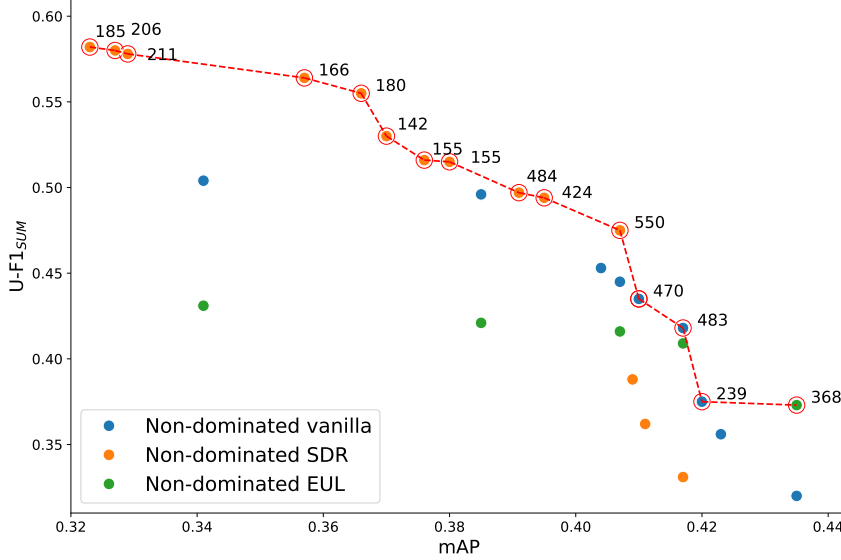


FIGURE 4.9: mAP and U-F1_{SUM} pareto front for non-dominated FMap parameter combinations of the three versions of the algorithm (vanilla, SDR and EUL).

Nonetheless, it is crucial to highlight that EUL technique focuses on improving the recall of unknown objects, what in turn causes a decrease in precision finally leading to lower U-F1 scores. Hence, it is also interesting to inspect the precision-recall trade-off in unknown object detection for these non-dominated solutions. This is presented in Figure 4.10, clearly evincing how EUL sacrifices the precision to gain recall in unknown objects.

In conclusion, each version of the algorithm presented exhibits distinct strengths and weaknesses. On the one hand, using a SDR method provides superior unknown detection capabilities and robustness while maintaining an acceptable known detection performance. On the other hand, applying EUL algorithm to FMap significantly boosts the recall of unknown objects at the cost of considerably reducing the precision in most of the cases, hence obtaining lower U-F1 scores. Nevertheless, it manages to enhance the U-F1 of the vanilla version to a degree that it becomes one of the non-dominated parameter configurations of our method. Finally, the basic version's configurations obtained an acceptable balance between mAP and U-F1 without incurring the computational overhead associated with the other two versions.

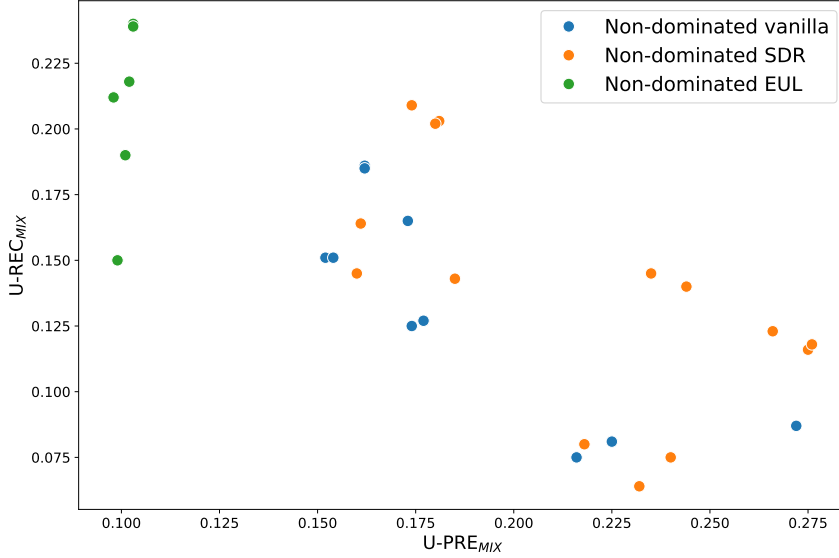


FIGURE 4.10: Precision vs Recall in COCO-Mix subset of non-dominated FMap parameter combinations for the three versions of the algorithm (vanilla, SDR and EUL).

4.4.3 RQ2. How does our method perform when compared to output-based post-hoc OoD detection methods in single-stage object detectors?

In this section, we evaluate the performance of the optimal configurations of our method in comparison to post-hoc or logits-based techniques. As previously mentioned, these methods also depend on the inference confidence threshold set for the single-stage object detector, prompting the testing of several confidence thresholds for each post-hoc algorithm.

The results are depicted in Figure 4.11. Among the post-hoc methods, the MSP method emerges as the top performer. When compared to the FMap method at comparable levels of unknown F1 scores, post-hoc methods generally maintain better known object detection capabilities. Notwithstanding the foregoing, the SDR version of our method manages to surpass the U-F1 score of the MSP method, though this comes with a reduction in mean average precision.

Summarizing, the post-hoc methods achieve similar U-F1 scores when compared to our technique while maintaining remarkable performance in the known classes.

4.4.4 RQ3. Does a fusion of feature-based methods with logits-based methods outperform other potential ensemble configurations?

The results obtained, in conjunction with the insights presented in Chapter 3, prompt a critical inquiry. Although the FMap method is marginally

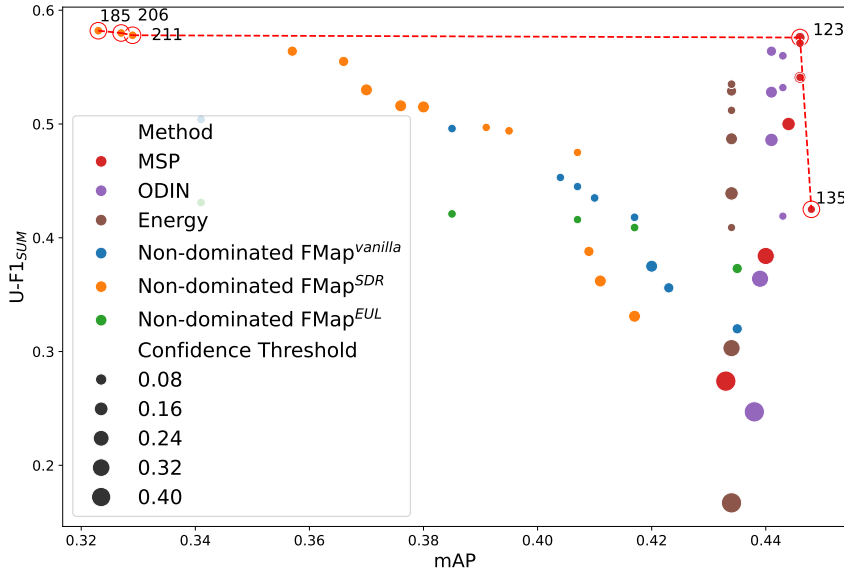


FIGURE 4.11: Pareto front for non-dominated FMap parameter combinations of the three versions of the algorithm (vanilla, SDR and EUL) against the post-hoc logits-based techniques.

outperformed by post-hoc detection techniques, could it potentially serve as a superior choice for method fusion, as opposed to relying solely on the more effective logits-based methods for ensemble strategies?

To respond to such question, we first have tested several possible ensembles together with the different fusion strategies defined in Subsection 4.3.4 to elucidate the optimal combinations. Figure 4.12 presents the results, where the distance metric used in the FMap method is used to name the fusion of methods. It can be clearly see how the OR criterion (in blue) provides the best U-F1 values in exchange for known object detection capabilities, whereas the AND strategy (in orange) does the opposite. As stated, the SCORE criterion (in green) creates a balance between both. Nevertheless, only one of the ensembles of this strategy is a non-dominated one. The other technique of the pareto fron are combinations of MSP with either another post-hoc method or a feature-based technique (FMap). The fusion of different well performing versions of FMap does not yield any non-dominated solution.

Subsequently, to enable better visualization of the final results, we have grouped together the different ensembles attending to the type (feature- or logits-based) of the fused methods. Additionally, we compare them to the techniques previously selected in research questions 1 and 2. This is graphically depicted in Figure 4.13. We can extract valuable insights from it.

On the one hand, FMap method (in blue), despite obtaining remarkable unknown object detection capabilities for some of the configurations, it alone can not outperform the rest the techniques when both mAP and U-F1

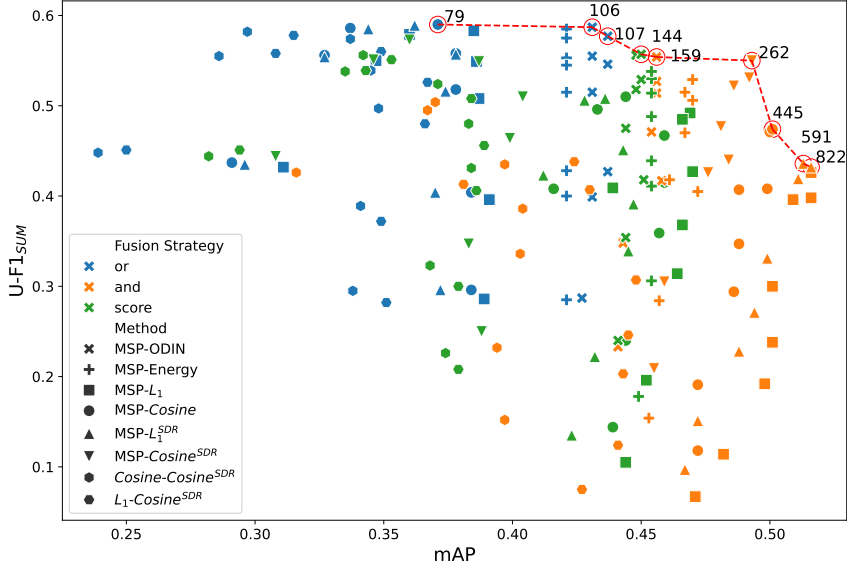


FIGURE 4.12: Pareto front for mAP and U-F1_{SUM} across various fusion possibilities of detection methods, distinguished by marker style, and for the three different fusion strategies, indicated by color. The naming convention is Method_A-Method_B, and the FMap method name is substituted by the distance metric used.

are the goal. On the other hand, the most successful post-hoc technique, MSP, is capable of becoming part of the final pareto front. In fact, it yields very similar performance the the combination of two post-hoc or logits-based methods. Finally, the combination of FMap with the best logits-based approaches (MSP), achieves the best U-F1 score for one on the possible configurations while obtaining also outstanding known object detection performance in other configurations.

In conclusion, the results presented in this subsection clearly evince that the fusion of post-hoc logit-based methods attain very similar performance in both known and unknown object detection. Conversely, ensembling a feature-based method (FMap in this case) with post-hoc detection methods provides diverse decision, ultimately leading to the enhancement of the performance obtained by both methods independently. Therefore, we can confidently assure that the fusion of feature-based methods with logits-based methods outperform other fusion possibilities.

4.4.5 RQ4. How does the performance of unknown object detection algorithms implemented on single-stage models compare to that of the SOTA?

To conclude, we compare some of the best performing methods among the ones presented against the SOTA methods in the Unknown Object

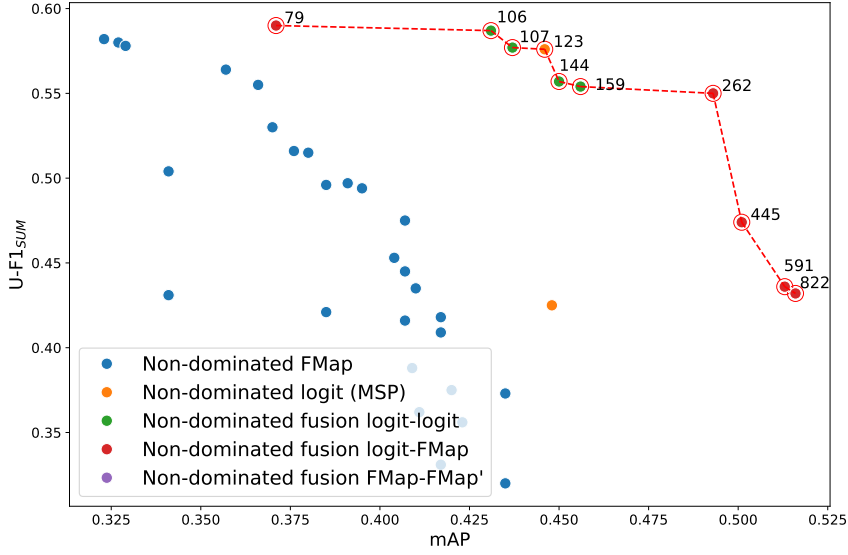


FIGURE 4.13: Pareto front for mAP and $U-F1_{SUM}$ across the best feature-based (FMap), logit-based (post-hoc) and fusion types. The fusion of different parameter combinations of FMap does not yield any non-dominated solution compared to the other ensembles, hence not appearing in the plot. For reference, VOS² reaches 0.36 mAP and 0.49 $U-F1_{SUM}$, ORE reaches 0.21 mAP and 0.43 $U-F1_{SUM}$ and UnSniffer reaches 0.36 mAP and 0.77 $U-F1_{SUM}$

Detection benchmark [112]. For the sake of clarity, we hand-picked some of them and we also include some results from the vanilla, SDR and EUL versions of the FMap method. Results are presented in Table 4.2.

Upon examining the detection outcomes for known objects, it is evident that the mAP values for the single-stage YOLOv8 model consistently surpass those achieved by Faster R-CNN and DETR. Additionally, it is noteworthy that YOLOv8 methods exhibit generally higher precision for unknown objects. However, these methods often achieve lower recall rates, which contributes to a more balanced U-F1. Subsequently, the highest U-F1 is achieved using the UnSniffer method. As for the rest, our methods demonstrate enhanced robustness, as evidenced by consistently lower scores in both A-OSE and WI metrics, indicative of a reduced propensity to misclassify unknown objects as known ones. This robustness is attributed to our approach as it avoids any retraining (contrary to SOTA methods, that retrain with pseudo-labels for unknown classes), significantly diminishing the likelihood of such misclassifications.

Method	Base Model	VOC-test	COCO-OoD				COCO-Mix						
		mAP	U-AP	U-F1	U-PRE	U-REC	mAP	U-AP	U-F1	U-PRE	U-REC	A-OSE	WI
-		0.483	-	-	-	-	-	-	-	-	-	-	-
MSP		0.470	0.213	0.314	0.279	0.359	0.364	0.055	0.169	0.190	0.153	588	0.135
Mahalanobis		0.447	0.129	0.271	0.309	0.241	0.351	0.051	0.149	0.207	0.116	604	0.165
Energy score	Faster	0.474	0.213	0.308	0.260	0.377	0.364	0.048	0.169	0.167	0.171	470	0.137
ORE	R-CNN	0.243	0.214	0.255	0.153	0.782	0.213	0.140	0.175	0.103	0.592	485	0.089
VOS ¹		0.485	0.135	0.196	0.342	0.137	0.377	0.040	0.101	0.262	0.062	640	0.152
VOS ²		0.469	0.205	0.317	0.291	0.348	0.364	0.051	0.172	0.184	0.163	409	0.124
UnSniffer		0.464	0.454	0.479	0.433	0.535	0.359	0.150	0.287	0.222	0.409	398	0.175
OW-DETR	DETR	0.420	0.033	0.056	0.030	0.380	0.414	0.007	0.025	0.014	0.161	569	0.086
-		0.750	-	-	-	-	-	-	-	-	-	-	-
<i>Cosine</i>		0.620	0.160	0.243	0.655	0.149	0.420	0.025	0.132	0.272	0.087	239	0.120
<i>Cosine</i> ^{SDR}		0.471	0.296	0.392	0.427	0.363	0.323	0.061	0.190	0.174	0.209	185	0.082
<i>Cosine</i> ^{SDR}		0.527	0.238	0.351	0.668	0.238	0.380	0.051	0.164	0.275	0.116	155	0.087
<i>L</i> ^{EUL}	YOLOv8	0.662	0.161	0.254	0.239	0.270	0.435	0.031	0.119	0.099	0.150	368	0.141
MSP		0.719	0.237	0.390	0.565	0.298	0.446	0.083	0.186	0.237	0.153	123	0.073
MSP- <i>Cosine</i>		0.567	0.289	0.397	0.563	0.307	0.371	0.048	0.193	0.222	0.171	079	0.057
MSP- <i>Cosine</i> ^{SDR}		0.765	0.275	0.362	0.417	0.320	0.493	0.107	0.188	0.197	0.179	262	0.092
MSP-ODIN		0.727	0.271	0.376	0.397	0.357	0.450	0.115	0.181	0.159	0.211	144	0.079

TABLE 4.2: Comparison between the SOTA methods and several methods presented in this chapter. The best methods for each metric in each dataset have the cells shaded in dark gray. In all the metrics, greater value is best except for A-OSE and WI, where lower is better. The hyperparameters are displayed in Table 4.3.

Method	Confidence Threshold	Cluster Method	Fusion Strategy
<i>Cosine</i>	0.100	<i>HDBSCAN</i>	-
<i>Cosine</i> ^{SDR}	0.010	<i>KMeans</i> ¹⁰	-
<i>Cosine</i> ^{SDR}	0.100	<i>One</i>	-
<i>L</i> ^{EUL}	0.050	<i>One</i>	-
MSP	0.050	-	-
MSP- <i>Cosine</i>	0.050	<i>HDBSCAN</i>	or
MSP- <i>Cosine</i> ^{SDR}	0.010	<i>HDBSCAN</i>	and
MSP-ODIN	0.010	-	score

TABLE 4.3: Hyperparameters used in our methods.

4.5 Conclusions

In this chapter, we have explored the UOD paradigm, closely related to OWOOD, for the single-stage object detector models. A new OoD or unknown object detection method, FMap, has been proposed. It is based on the information extracted from the feature maps and adapts to and leverages the particularities from current single-stage object detectors.

In addition, we elucidate two ways of improving the performance of this technique. The first works by lowering the dimensionality of extracted features by capitalizing on existing SDR methods. The second, named EUL, tries to alleviate the reduced inherent capabilities of the model to localize unknown object by exploiting the information present on the feature maps.

Subsequently, as the objective of the UOD and OWOOD paradigms is to demonstrate unknown object detection capabilities while preserving the performance on known classes, we explored the different configurations of FMap that lead to better outcome in both. Specifically, we tested different combinations of cluster methods, distance metrics and inference confidence threshold for the vanilla, SDR and EUL versions of our algorithm. We use the UOD benchmark, by [112], for this and the subsequent experiments, obtaining in each case the pareto fronts emerging from the balance of the objectives. We concluded that each version of the algorithm presented has its own unique advantages and disadvantages. The

SDR method, for instance, excels in detecting unknown objects (measured by the U-F1) and in providing robustness (measured by the A-OSE), while preserving a satisfactory performance for known objects (measured by the mAP). Conversely, integrating the EUL algorithm with FMap markedly enhances the detection of unknown objects, although this often results in a notable decrease in precision across many instances. The vanilla version obtained an acceptable balance between both objectives—known and unknown detection—without the additional complexity associated with the other two.

Thereupon, we compared the best configurations of FMap against several post-hoc methods. The results showed that post-hoc methods generally lead to better known object detection capabilities, for the same unknown object detection performance.

With these results at hand, we explored different possible combinations or fusions between the tested detection methods. For the fusion, we developed three different possible strategies to elucidate the final outcome when participating methods disagree. Experiments evince how the ensemble of two post-hoc or logits-based techniques barely improve the results obtained by each of the methods independently. In contrast, when the fusion is between FMap, a feature-based approach, with a post-hoc technique, remarkable performance is achieved, outperforming the ensemble’s component methods. This conclusion goes in line with one of the insights gained in Chapter 3 that demonstrate how logits- and feature-based techniques produce diverse decisions, leading to outstanding results.

Finally, we compare a few of the best configurations found in our experiments against the SOTA methods for UOD and OWOD. Results show how our methods, that use a single-stage object detector, consistently surpasses those achieved by the SOTA methods in known classes. Regarding the detection of unknowns, they demonstrate superior recall (ORE [100]) and F1 (UnSniffer [112]) scores, whereas one of FMap configurations obtained the best precision. Furthermore, our methods show enhanced robustness, as evidenced by lower scores in both A-OSE and WI metrics, which suggest a reduced likelihood of misclassifying unknown objects as known. This robustness advantage stems from our strategy of avoiding retraining—a common practice in SOTA methods that use pseudo-labels for unknown classes—thus significantly reducing the chances of misclassifications.

To conclude, in this chapter we demonstrate how single-stage object detectors are inherently robust and how they can be easily endowed with decent unknown object detection capabilities compared to the SOTA. Moreover, we have designed an OoD or unknown object detection method for single-stage object detectors that obtains acceptable results and is very suitable for the fusion with the existing post-hoc methods.

In the future, we plan to further develop the presented FMap method, as the presented approach was a simple algorithm to leverage the features extracted by the model. We want to test techniques to fuse the features of each stride into one common space, as well as implement other SDR algorithms.

Chapter 5

Out-of-Distribution Detection in Reinforcement Learning Scenarios with Sparse Rewards

The previous chapters have expanded the OoD strategies to new type of models, the SNNs, and to less explored learning problems, specifically, object detection tasks. In this chapter we tackle AI safety in the RL paradigm by increasing the robustness of RL agents against OoD observations, as well as by exploring the best strategies to adapt to new RL scenarios.

As previously explained, although OoD detection and adaptation has been addressed mostly for classification/regression tasks, the challenges posed by OoD data in RL scenarios acquire a unique dimension. RL, where agents learn to make decisions by interacting with their environment, is particularly susceptible to safety risks due to its inherently exploratory nature. RL agents are trained to generalize from their experiences, but they often face difficulties when exposed to situations (*states*) that are not represented in their training data. This limitation can lead to suboptimal or even catastrophic decisions, making the integration of OoD detection in RL systems a critical design concern for AI safety and a research priority over the last years [118, 119, 120].

From the technical perspective, the incorporation of OoD detection and adaptation functionalities in RL agents is not straightforward. To begin with, RL environments are dynamic and often complex, making the identification of OoD states from the interaction dynamics of the agent a non-trivial task [121]. Furthermore, most literature dealing with OoD adaptation strategies assume environments with dense reward signals [122, 123, 124]. Consequently, it is unclear how the agent should detect OoD and adapt to new environments in RL scenarios where the agent receives infrequent or occasional feedback from the environment (*sparse rewards*). The need for properly balancing between exploration and exploitation in sparse settings opens new possibilities for OoD detection (e.g. unsupervised intrinsic motivation, self-supervised approaches). However, one cannot ascertain whether off-the-shelf OoD detection techniques developed for

supervised learning scenarios should be utilized instead. Furthermore, the sparsity of the environment questions whether the knowledge learned by the agent before the occurrence of OoD data should be retained (*transferred*) or discarded.

This chapter addresses this gap by advancing over the detection and adaptation to OoD data in RL environments with sparse rewards. Specifically, we introduce and describe several OoD detectors and adaptation strategies that can be adopted for a trained RL agent to realize a change in their observed environment, and adjust its knowledge accordingly. On one hand, the OoD detection techniques include approaches that leverage the existence of IM modules in the agent’s architecture, as well as techniques that rely on the internal activations of the agent’s models. On the other hand, adaptation strategies under consideration vary in terms of the amount of knowledge retained by the agent once OoD data has been captured in its observation. Preliminary experiments over Procedural Content Generation (PCG) scenarios with sparse rewards are run for different OoD artifacts, showing that the choice of the best detection and adaptation strategy depends on manifold factors related to the impact of the OoD on the learned behavioral policy and the similarity of the OoD object to known elements in the environment. Our findings expose the difficulty posed by sparsity in the design of effective OoD detection and adaptation strategies for safe and trustworthy RL agents.

The rest of the chapter is structured as follows. First, Section 5.1 briefly revisits sparse RL, continual RL and generalization, OoD detection and adaptation in RL, ending up with a statement about the novelty and contribution of this chapter. Next, Section 5.2 poses the mathematical problem targeted, and describes each of the detection and adaptation baselines considered in our experiments. The experimental setup is detailed in Section 5.3, including the formulation of the research questions that experiments aim to answer. Results for each research question are presented and discussed in Section 5.4. Finally, Section 5.5 concludes the technical contribution with a summary and prospects towards future research lines.

5.1 Related Work and Contribution

Before delving into the problem statement and the description of the OoD detection techniques and adaptation strategies, we briefly overview key concepts and advances around sparse RL (Subsection 5.1.1), continual RL and generalization (Subsection 5.1.2) and OoD detection in RL environments (Subsection 5.1.3). Finally, we outline the novel aspects of the technical contribution of this chapter based on the reviewed literature (Subsection 5.1.4).

5.1.1 Sparse Reinforcement Learning

In sparse reward settings, where dense feedback signals are lacking, balancing properly the trade-off between exploration and exploitation in the agent’s interaction with the environment becomes crucial [125]. To this

end, various strategies can be devised, based on the data and available resources. When an (sub)expert is available, transfer learning based on pretrained models offers a practical solution to leverage existing expertise that is assumed to be helpful for solving a new (target) RL scenarios [126]. In scenarios where only (sub)optimal demonstrations are accessible, imitation learning techniques can be adopted instead [127, 128, 129, 130]. In scenarios with no prior knowledge, self-supervised learning (SSL) can significantly enhance the agent’s sample-efficiency by focusing on the representation learning capability of the agent [131, 132, 133, 134]. Paramount among these strategies are the so-called Intrinsic Motivation (IM) techniques. By motivating the agent to discover novel states [135], such approaches address the exploration problem by generating additional reward signals that encourage the agent to be more curious about its current knowledge [136, 137, 138, 139, 140].

5.1.2 Continual Reinforcement Learning and Generalization

Open-world learning has been studied in RL to a lesser extent than in other learning scenarios, mainly due to the implementation issues inherently posed by RL in real-world settings (e.g. sample efficiency). Even so, some areas related to open-world learning have drawn the attention of the RL community. Two of them deserve a specific mention in the context of this work. On one hand, the goal of continual RL is to obtain an autonomous agent that learns incrementally and online from a continuous stream of tasks, acquires and builds upon new skills in a task-agnostic manner, minimizes forgetting of previous knowledge, and efficiently adapt and recover from changes over time [141]. Generalization in RL takes a more approachable path towards the deployment of RL-based systems that are able to extrapolate its learned skills to distributions outside their training data, either by learning policies that are robust to environment variations, or by modeling policies that resiliently adapt to such variations [142].

5.1.3 Out-of-Distribution Detection in Reinforcement Learning

The aforementioned paradigms pursue an agent capable of incrementally learning new tasks or, alternatively, behavioral policies that are robust enough to solve never-seen-before tasks [143]. An effective approximation to these goals would be an active adaptation, where the focus is on relearning as quickly as possible once a change is detected [144]. In this context, OoD detection in RL frameworks has not been extensively explored, despite gaining increasing momentum in recent years.

The first OoD detection technique in RL was reported in [145], which proposed to use the entropy of the predicted actions of an agent as a score for detecting novel states. Nevertheless, their solution required Q-value based algorithms and only two types of disturbances were evaluated over

two simple environments, *GridWorld* and *LunarLander*. The next work related to OoD detection is [124], which evaluated the previous technique over a novel benchmark. Environments were still relatively simple (*Pong* and discrete *Cartpole*), but methods were tested with more semantically meaningful elements, e.g., changing the dynamics of the environments by applying external forces or changing the mass of the cart. In [123], the authors introduced a method based on recurrent implicit quantile networks for detecting OoD dynamics, which essentially refers to detecting OoD states. They compared their method against non-probabilistic networks and random forest in the previously mentioned environments, again being characterized by dense rewards. More recently, [121] takes a step forward in OoD detection for RL by providing more clear definitions of what an OoD means in this field, distinguishing between covariate and semantic shifts following the standard in supervised machine learning [10]. They also propose a method that relies on predicting the forward dynamics of the environment via probabilistic models to enable detection by monitoring the prediction error on the next state predictions. *CartPole*, *HalfCheetah*, *Reacher* and *Pusher* environments are considered to assess the performance of their detector.

5.1.4 Contribution

In view of the existing literature, we identify several gaps. First, although the generalization of RL agents has been examined in the past, the criticality of OoD detection, which is necessary for triggering the need for adaptation, has been widely overlooked by the literature. Secondly, the environments used to evaluate OoD detection and adaptation algorithms in RL have been predominantly based on the assumption of dense reward signals and full observability. Our research aims to explore OoD detection in environments characterized by sparse rewards and partial observations, where the complete state is not made available to the agent.

5.2 OoD Detection Techniques and Adaptation Strategies

After outlining the contributions of this chapter, we begin by presenting a brief rationale for the use of IM in sparse environments in Subsection 5.2.1. We then detail the OoD detection formulation tailored for sparse RL environments in Subsection 5.2.2, building on the concepts introduced in Section 2.1. Subsequently, Subsection 5.2.3 details the different OoD detection methods utilized. Finally, we enumerate the different strategies considered for the adaptation of the agents in Subsection 5.2.4.

5.2.1 Exploration with Intrinsic Motivation

In complex RL scenarios, the agent’s learning process can be hindered by the absence of dense reward signals from the environment. As previously

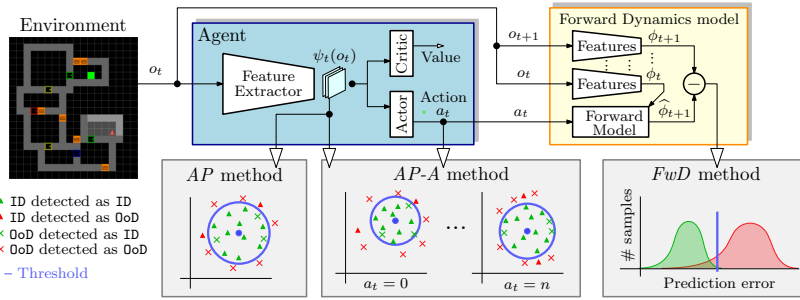


FIGURE 5.1: Schematic diagram showing the three methods (*FwD*, *AP* and *AP-A*) used to detect OoD observations in RL.

mentioned in Section 5.1, IM can be used to foster exploration by generating an additional intrinsic reward r_t^i that aims on modelling the agent’s curiosity. As a consequence, the reward at each time step t is updated as $r_t = r_t^e + \beta \cdot r_t^i$, where β is a coefficient that balances between the extrinsic and intrinsic rewards.

In this context, various approaches in the literature aim to generate exploration bonuses. For the purposes of this chapter, we focus on the prediction-error methods, as one of the OoD techniques incorporates an element of these approaches. Prediction-error methods calculate the intrinsic reward based on the error when predicting the outcome of the agent’s action in the environment, essentially assessing how predictable the environmental changes are. The underlying idea is straightforward: the more accurate the prediction, the more frequently that situation has likely been encountered, and the smaller the novelty bonus should be.

Intrinsic Curiosity Module (ICM) [136] is one of the most prominent research work in this area. It is focused on creating a small feature space to compute the expected changes that affect the prediction, and achieves it by using two strategies. On the one hand, uses an inverse dynamics model in a self-supervised manner to predict the agent’s action $\hat{a}_t$ given the current state feature representation or embedding ($\phi(s_t)$ or ϕ_t) and next state embedding ($\phi(s_{t+1})$ or ϕ_{t+1}). On the other hand, it trains a forward dynamics model that uses the embedding (ϕ_t) of the actual state s_t and the actual action a_t to predict the next state embedding ϕ_{t+1} . Both predictions are then compared to the actual action and state embedding obtained to train the modules, that are modeled as NNs.

5.2.2 Out-of-Distribution Detection Formulation in Reinforcement Learning

Bearing the previous RL notation in mind, we follow a similar rationale to that recently posed in [121], but with two main differences: 1) we assume that the RL environment produces sparse rewards signals; and 2) we deal with a POMDP. The latter implies that at time t , the agent will only have access to the observation at that specific instant of time (o_t). Therefore, we define an OoD within the observation (or equivalently, an

OoD observation) if a semantically different element with respect to those encountered by the agent during the training phase appears within the observation. Likewise, we consider that an element deployed on the environment is OoD if it corresponds to a new concept and it affects the behavioral policy learned by the agent when no OoD objects are present. In mathematical terms, we can assume an OoD-free distribution of environments $\mathcal{D}$ characterized by a state transition probability $P_{\mathcal{D}}$. This distribution can be sampled to yield levels which the agent explores, so that a learning algorithm can learn a policy $\pi_{\mathcal{D}}(a_t|s_t)$ from the samples collected during its interaction.

Once the agent is trained over $\mathcal{D}$, we assume the appearance of OoD elements that modify the state transition probability $P_{\mathcal{D}}$ to $P_{\mathcal{D}}^{ood}$. Consequently, we can think of a new distribution of levels $\mathcal{D}^{ood}$ which is similar to $\mathcal{D}$ in regard to their goal and available actions, but wherein the insertion of OoD elements affects the state transition probability and hence, the agent must adapt its learned policy to the new transition probability. Actively adapting to this distributional change requires first detecting the presence of an OoD element based on the information available in the agent at time t (namely, its observation o_t) and its knowledge, represented by its internal learnable parameters. To this end, an OoD detection technique can be defined as a binary mapping $f_{ood} : \Omega_{\mathcal{D}}^{ood} \rightarrow \{\text{ID}, \text{OoD}\}$, which processes every observation $o_t \sim \Omega_{\mathcal{D}}^{ood}$ (with $\Omega_{\mathcal{D}}^{ood}$ denoting the expanded space of observations as per the insertion of OoD elements) and assigns label ID if the observation is declared to be free of OoD objects. Otherwise, label OoD represents the case when the OoD detection technique declares that the observation contains an OoD element.

Once an OoD has been detected in a given observation, the agent must choose an strategy to accommodate the change in the distribution of the levels to solve. At this point several strategies can be defined depending on the degree of knowledge retained from the previous, OoD-free distribution $\mathcal{D}$. In all cases we will assume that upon the detection of an OoD observation, the agent continuously collects samples from its interaction with the levels over time, updating its trainable parameters accordingly. The goal of the different adaptation strategies that can be devised to this end is to resiliently improve the speed and effectiveness of the agent’s performance recovery in the new distribution of OoD levels. It is important to remark that the assumption of sparse environments and the different impact of the OoD elements in the learned behavior of the agent exacerbate even further the decision of how much knowledge should be transferred between distributions, even if the distribution of OoD levels $\mathcal{D}^{ood}$ is similar to $\mathcal{D}$.

5.2.3 Out-of-Distribution Detection Techniques

When it comes to the detection of OoD objects in the agent’s observation, we propose straightforward methods based on leveraging the agent’s internal structure and further learning elements that are often employed to learn effectively from sparse environments, summarized graphically in

Figure 5.1. In all cases, OoD detection rely on characterizing the behavior of the agent while learning to solve levels belonging to the OoD-free distribution $\mathcal{D}$ (as per the activations of its compounding modules), and on assessing whether activations unchained by observations o_t at inference time deviate from such characterized activation patterns:

- The first approach (*Forward Dynamics*, hereafter labeled as *FwD*) assumes the existence of an IM module to compare the embedding vector of the next predicted state $\hat{\phi}_{t+1}(o_t, a_t)$ with the embedding vector of the true state obtained ϕ_{t+1} . Note that the state in a POMDP is perceived by the agent as an observation o_t . If a measure of distance between the embeddings $d(\phi_{t+1}, \hat{\phi}_{t+1}(o_t, a_t))$ is greater than a predefined threshold τ (which can be tailored based on the behavior of the distance over the OoD-free distribution $\mathcal{D}$), the agent declares that observation o_t is OoD. Similar methods that also hinge on the prediction of the next state embedding have been proposed recently [121].
- Based on the agent’s model mapping observations to actions $g_\theta(o_t)$, we can devise a novel detection approach that generates a space of embeddings based on the features extracted by the agent from the environment. This detection approach, in what follows referred to as *AP* (*Agent’s Perception*), extracts the activation patterns $\psi_t(o_t)$ generated within $g_\theta(o_t)$ by every observation $o_t \sim \Omega_{\mathcal{D}}$ seen during the agent’s training process over $\mathcal{D}$. In this space of embeddings, we compute a measure of distance between the activation pattern of a query observation to the centroid (the average) of the activation patterns collected during training. A threshold imposed over such a distribution establishes that if the distance of its activation patterns surpasses this threshold, it is declared to be an OoD observation.
- A variation of this latter method, labeled as *AP-A*, associates one subspace of embeddings to each action, so that instead of one space we end up with as many subspaces as actions, each one filled with the activations $\psi_t(o_t)$ that led to the action a_t for observation o_t . Each subspace of thresholds is configured with its own threshold. Then, during inference, we proceed as in the original *AP* approach, but each observation’s activations are now compared only with the subspace corresponding with the action a_t taken by the policy observing o_t . Figure 5.1 depicts schematically the similarities and differences between *FwD*, *AP* and *AP-A*.
- Post-hoc OoD detection methods proposed in the context of supervised learning scenarios can be also utilized over $g_\theta(o_t)$. While proposals in this realm abound in the literature, we concentrate on two of them for their algorithmic simplicity and widespread adoption as baselines for preliminary benchmarks. Specifically, two presented in the Subsection 2.1.1 in the Chapter 2, the MSP and the Energy methods.

5.2.4 Out-of-Distribution Adaptation Strategies

Once OoD states have been detected, the agent must decide how to adapt to the new distribution of levels $\mathcal{D}^{ood}$. To this end, several adaptation

strategies can be implemented while the agent collects samples and learns online over the new distribution of levels:

- The simplest strategy is to start the agent’s training from scratch by discarding any prior knowledge modeled over $\mathcal{D}$, a method coined as *Train_scratch*. This simple procedure could be appropriate if the semantic shift between $\mathcal{D}$ and $\mathcal{D}^{ood}$ is large enough for the knowledge of the agent to be of no help in the new level regime. Our experiments will reveal that this strategy is detrimental when OoD elements emerge in the environment incrementally, without imprinting major changes in its state transition.
- Part of the knowledge acquired by the agent over $\mathcal{D}$ can be retained and used to expedite the learning process over the new distribution of levels $\mathcal{D}^{ood}$. Several choices can be made in this regard:
 - To transfer all parameters learned by the agent over $\mathcal{D}$, including those defining the AC architecture (*Transfer AC*). This approach implicitly assumes that the trained agent already possesses an enough explorative behavior to solve the new distribution of OoD levels.
 - To transfer not only the parameters defining the AC, but also the IM module. Two possibilities hold:
 - *Transfer AC + IM_scratch*: here we still use an IM module, but we do not inherit the weights learned from the previous distribution $\mathcal{D}$. The agent will be forced to explore the levels of the new environment distribution as if everything is actually novel.
 - *Transfer AC + IM*: here we inherit the learned weights of both the AC and IM modules. We expect that the latter will be able to capture the novelty present in OoD observations belonging to the new distribution $\mathcal{D}^{ood}$.
 - To transfer not only the parameters of the AC, but also an additional self-supervised module that extends the feature extraction capability of the agent [133]. Three different options apply in this case:
 - *Transfer AC + SSL_only*: the agent will be updated solely with the self-supervised auxiliary loss defined in [133], i.e., there is no RL loss. With this variant we aim to update only the learned representations, and verify if it suffices for adapting the agent to $\mathcal{D}^{ood}$.
 - *Transfer AC + SSL*: the agent will be updated with both the RL and the self-supervised auxiliary losses. In this context, the agent can update its learned knowledge based on the two losses, and also adapt its behavior with the corresponding RL loss.
 - *Transfer AC + SSL + IM_scratch*: the agent will be updated with both the RL and the self-supervised auxiliary losses, being the former augmented with more exploratory capabilities with a newly initialized IM module. As a consequence, the agent should be able not only to adapt its representation, but to also adapt even faster with the curiosity driven rewards.

By establishing the above adaptation strategies we aim to observe several aspects. First, we would like to gauge whether transfer learning is beneficial in the context of OoD. Secondly, by comparing transfer learning with and without IM we can assess if the exploratory capacity learned by the agent over $\mathcal{D}$ is sufficient to adapt to new OoD levels. Thirdly, we seek to determine whether it is more effective to force the agent to explore everything as novel (using IM, but without transferring its weight values) than to make novelty primarily associated with OoD observations (transferring the learned IM weights). Last but not least, we analyze whether using SSL can be useful in order to adapt the representation of the agent when facing the new OoD observations under different scenarios.

5.3 Experimental Setup

In order to assess the performance of the detection techniques and adaptation strategies presented previously, several experiments were carried out towards answering with evidence two research questions:

- *RQ1*: How do the OoD detection method perform when compared to each other and to other post-hoc detectors? Is their performance subject to the distance metric under choice? Is it convenient to link states to actions in our proposed method?
- *RQ2*: Which adaptation strategy performs best, transfer learning or learning from scratch? How should knowledge transfer in the presence of OoD objects be done?

To answer the above two questions we use procedurally generated RL scenarios from the so-called Minimalistic Gridworld Environment [146], a collection of discrete grid-world sparse reward environments with flexible design options. In all environments the agent has a discrete action space and has to navigate a two-dimensional map featuring different obstacles. We have selected the environments labeled as **MultiRoom** $\mathbf{MNXY}$, where the agent must open multiple doors (without requiring a key) between sequentially placed rooms in order to reach the goal, which is placed inside the last room. These environments are denoted as $\mathbf{MNXY}$, with $\mathbf{X}$ denoting the number of rooms and $\mathbf{Y}$ their maximum size (with its minimum equal to 4). Observations are provided as a $7 \times 7 \times 3$ matrix representing a 7×7 tile set in the direction agent is facing, with the 3 channels representing the type (or id), color and status, respectively, of the object in the corresponding tile cell. The available actions are *turn left*, *turn right*, *go forward*, *toggle* (to open doors), *pickup*, *drop* and *done*¹.

In our experiments, the agent is trained on **MN5S8**, i.e., an environment with 5 rooms with varying sized between 4 and 8. Three new environments with added OoD elements (Figure 5.2) based on this environment are presented to the trained agent, which are labeled depending on the OoD

¹The last three actions might be useless in most of our experimental setup. However, experiments assume that the whole set of actions defined in MiniGrid's is available for the agent.

element itself: **MN5S8-lava**, **MN5S8-ball** and **MN5S8-sand**. All these new environments are generated procedurally; the OoD elements are placed following certain probabilistic rules. The *lava* block, which terminates the level if stepped on by the agent, replaces *wall* blocks with a 10% probability. On the other hand, the *ball* block, which cannot be passed through, appears in a random location in each room once and without blocking the doors. Finally, unlike *lava*, the *sand* block does not terminate the level if stepped on; it appears in a random location of every room with a probability equal to 50%. The objective of these environments remains the agent’s original goal (i.e., to navigate the whole maze until reaching the green square), but with the added complexity that now new OoD elements that are unknown to the agent (in both representation meaning and their dynamics) have been introduced.

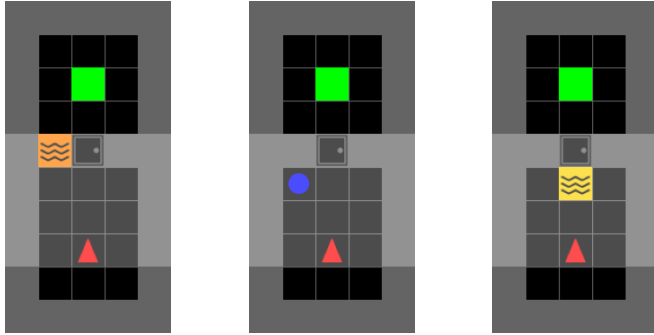


FIGURE 5.2: Three distinct levels illustrating environments that the lava, balls and sand OoD elements. The goal is still to traverse the door and reach the green square. Note this are just examples, not the actual new OoD environments.

Regarding the experiments to answer *RQ1*, we compare the detection methods presented in Section 5.2.3, where logits for post-hoc detection methods are taken from the actor’s head. To evaluate the detection performance, we considered up to 1,000 different levels of MN5S8 (i.e. without OoD elements) and another 1,000 different configurations in each of the other new environments (i.e., **MN5S8-lava**, **MN5S8-ball**, **MN5S8-sand**). A state is assumed to be OoD if the agent has an OoD in its observed tile. To avoid overoptimistic performance figures, they are computed over unique states, i.e. we eliminate repeated states from the computation. Indistribution states are taken from any of the possible 1,000 levels of the regular MN5S8 environment over which the agent is trained. We report the AUROC (explained in Subsection 3.3.4) of the different approaches under consideration. The Energy method temperature parameter is set to 1 as paper suggests [15].

When it comes to *RQ2*, we assume that the agent, already trained over 1,000 different configurations of the *regular* or OoD-free MN5S8 environment, is queried with levels belonging to any of the new tasks (i.e., **MN5S8-lava**, **MN5S8-ball** and **MN5S8-sand**) that contain OoD elements. Again, 1,000 different level configurations are considered for each one. The agent must adapt via any of the strategies in Section 5.2.4 and continue

solving the maze by learning how these OoD elements affect in achieving the goal. We evaluate each adaptation strategy in terms of the speed of convergence, as well as the success rate achieved by the agent over the levels of the environment with OoD elements.

In what refers to the algorithmic implementations, we use PPO [60], explained in Subsection 2.4.2, with rollout size equal to 2048 and a clipping factor (ϵ) of 0.2, alongside GAE for computing advantages [59], with $\lambda = 0.95$. When it comes to IM, we resort to Rewarding Impact Driven Exploration (RIDE) [138] with episodic counts. This method is based on ICM, that is briefly introduced in 5.2.1. After a grid search, we select an intrinsic coefficient β equal to 0.05, and an entropy coefficient $\xi = 5 \cdot 10^{-4}$ for all the environments under study. However, for the specific environment MN5S8-*lava*, these values are modified to $\beta = 0.005$ and $\xi = 5 \cdot 10^{-5}$, respectively. In accordance with other contributions [138, 140], the RL agents are build upon an AC neural network architecture with 3 convolutional layers acting as feature extractors (parameterized with θ_e), followed by fully-connected layers that precede their policy and value heads. Regarding SSL, we employ the inverse dynamic module as in [133], where the self-supervised parameters (parameterized with θ_s) are directly connected to the feature extractor (θ_e) with 2 fully-connected layers of 64 neurons each.

All source codes of this chapter have been released in a GitHub repository available at <https://github.com/aitor-martinez-seras/ood-s-parse-rl>.

5.4 Results and Discussion

We now present and discuss on the experimental results corresponding to RQ1 (Section 5.4.1) and RQ2 (Section 5.4.2).

5.4.1 RQ1: Out-of-Distribution Detection Performance

To begin with, it is important to examine the impact of the different OoD elements on the behavior of the trained agent. To this end we first stop at Table 5.1, which shows the mean return and the success rate achieved by the agent trained with IM over MN5S8 (i.e. a more complex environment), and for a random agent (i.e. an agent without any training whatsoever).

We note that the performance decreases notably upon the insertion of the OoD elements; as expected, the agent almost unable of completing the maze in the case of *lava* and *ball*, as the agent does not finish the level upon stepping on these objects. This conforms to intuition given the behavior of every OoD element described previously. When trained over MN5S8, the agent’s learned behavior is biased to getting closer and toggling any object different than a *wall* (i.e. a door). Therefore, when encountering an OoD object, the agents tends to toggle and traverse it, unchaining different effects in its behavior: *lava* and *ball* terminate the level (either immediately or after the simulation horizon is met), whereas the *sand* object is configured to distract the agent and delay its trajectory towards

TABLE 5.1: Test performance in terms of average return and success rate of an agent trained on MN5S8 and a **Random** agent over different PCG environments (evaluated over 1,000 randomly selected levels). It can be seen that both metrics dramatically decrease when evaluating the agent in OoD scenarios. Note that using a random policy is insufficient to complete successfully any task.

		Trained on MN5S8 (with IM)		Random agent	
		Mean Return	Success Rate [%]	Mean Return	Success Rate [%]
Regular	MN5S8	0.65	98.21	0.00	0.00
	MN5S20	0.36	71.00	0.00	0.00
OoD	MN5S8-lava	0.01	1.78	0.00	0.00
	MN5S8-ball	0.01	1.22	0.00	0.00
	MN5S8-sand	0.32	53.05	0.00	0.00

the goal. In the three cases, if the observation comprises both a door and an OoD element, the agent will select one or another depending on their arrangement inside the observation. All in all, the performance decreases of the OoD objects reported in Table 5.1 follow logically from their behavioral impact on the agent. Overall, these preliminary results evince that the performance of an agent cannot be regarded as a reliable indicator of the presence of OoD objects, as it can be related to the complexity of the environment itself, and not necessarily to the presence of an unfamiliar object. In fact, when introducing MN5S20, where no OoD elements are present, performance decays.

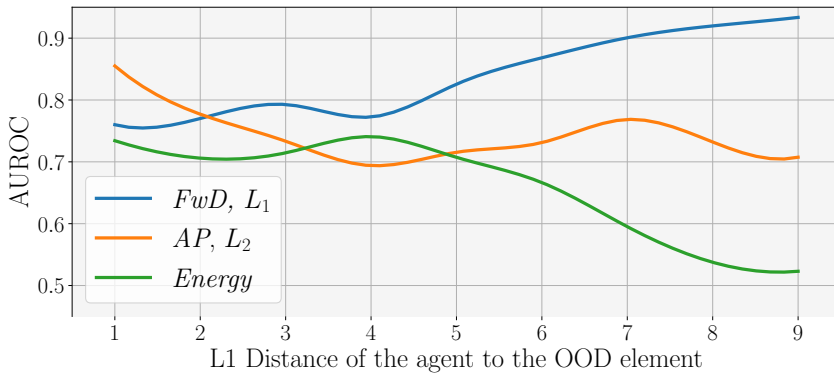
We proceed by examining the performance of the different OoD detection techniques under consideration in our benchmark and that of the technique proposed in this chapter. To this end, we focus on the AUROC scores shown in Table 5.2, which are given separately for each OoD environment. Several interesting conclusions can be drawn about the methods. First, the use of either an L_1 or an L_2 distance metric yields different results for the different detectors. For *FwD*, the L_1 distance attains achieves slightly better detection scores, whereas for *AP* or *AP-A* L_2 renders a superior performance. In any case, gaps between the results of both distances are less than 2%. A comparative analysis between *AP* and *AP-A* consistently demonstrates the superior performance of the former, as evidenced by its consistently higher AUROC scores across all OoD environments.

A closer inspection at Table 5.2 reveals that low detection results are obtained in the MN5S8-ball environment by all the approaches in the benchmark. This can be explained by the structure of the observations, which are represented as 7×7 tiles. Each tile entry denotes the Identification Number or IDN of the object at the corresponding position within the grid in the first of the channels. Specifically, IDNs 9 and 6 correspond to *lava/sand* and *ball*, respectively. During training in the MN5S8 environment, only *walls* (IDN 1) and *doors* (IDN 4) are present. Consequently, the ball object, with IDN 6, is numerically closer to a *door* block than a

TABLE 5.2: AUROC performance over each OoD environment for different detection techniques. Best results for each environment are highlighted in bold.

Technique	Distance	MN5S8 lava	MN5S8 ball	MN5S8 sand
FwD	L_1	88.39	70.10	77.48
	L_2	87.80	69.59	75.64
AP	L_1	84.30	59.70	84.46
	L_2	84.82	60.30	84.91
$AP-A$	L_1	81.13	56.41	80.32
	L_2	82.41	57.30	81.53
MSP	–	48.22	58.04	63.36
Energy	–	75.89	59.93	77.72

lava block (IDN 9). This numerical proximity results in neural networks, which process information as continuous values, categorizing a *ball* as more akin to *doors* than to *lava*. Consequently, *lava* objects are perceived as more distant from the training distribution, rendering them more easily identifiable as OoD objects within the RL model.

FIGURE 5.3: Average AUROC of best performing FwD , AP and Energy versus the L_1 distance between the OoD object and the agent.

Another interesting view on the results obtained for RQ1 is the relationship between the effectiveness of each OoD detection method in detecting unfamiliar objects and the distance of the OoD object to the agent in the observation space. Figure 5.3 illustrates the detection performance of three different detectors (FwD and AP with L_2 distance and Energy) versus the L_1 distance of the OoD element to the agent. For each distance value, we consider only the states where the OoD object was located at that specific distance value from the agent. This plot exposes that the closer the OoD object is to the agent, the better performance AP exhibits with respect to the other two approaches. The opposite is observed for the FwD method, which excels when the OoD object is far from the agent. The reason for

this lies in the fact that *AP* relies on the features extracted by the agent from the environment, which grants more importance to parts of the observation closer to the agent. In contrast, the forward dynamics model lacks such bias since its objective is not to take the next step, but rather to predict the next latent state. As for Energy method, it hinges on how confident the agent is in taking an action, it aligns with the expectation that the more distant an OoD element is from the agent, the less it influences the probabilities of making each decision. Consequently, a distant OoD object becomes more challenging to detect by this method.

5.4.2 RQ2: Adaptation Strategies

We now turn the focus of the discussion on Figure 5.4, which depicts the average return of the agent when adapting to different environments with OoD objects: MN5S8-*lava* (Figure 5.4.a), MN5S8-*ball* (Figure 5.4.b) and MN5S8-*sand* (Figure 5.4.c). We compare the different strategies described in Section 5.2.4 with an agent that learns in the considered environments with OoD elements by randomly initializing the weights of both the AC and IM modules (i.e., *Train_scratch*).

We begin our analysis by examining the performance of the strategy of not using IM during adaptation; this is, *Transfer AC* (light blue). Regarding this strategy, both in the MN5S8-*lava* and MN5S8-*ball* environments the average return does not reach the expected performance and reports a significant standard deviation. This is attributed to simulation seeds where the agent was not able to learn, which may occur for various reasons. Given that we are dealing with environments with sparse rewards and that the agent lacks an IM module, it only receives a reward when it successfully reaches the end. As a consequence, regardless of the introduced OoD elements, if the agent manages to successfully solve the task after observing an OoD element, it would be able to learn the new OoD objects' representation and their impact in the environment (i.e. dynamic) and therefore to adapt itself. In contrast, if the initial performance drop is too severe rendering the agent unable to solve any level (of the 1,000 different ones available), or the solved ones are because no OoD observation was present (by chance), no learning will be done about the OoD elements' representation and the agent will fail to adapt. For instance, when introducing *sand*, which is a distractor element that does not hinder the agent too much in the achievement of the goal, we can see that there is little to none performance decay and the agent can learn the OoD environment in all the runs consistently. Nevertheless, in *lava* and *ball* environments, which are elements that kill the agent or obstruct its navigation and where the return initially decreases to 0.01 (check Table 5.1), the adaptation is slower and in some runs the agent is unable to learn.

With respect to the strategy of using the IM module without learned weights, it successfully resolves the majority of cases (*Transfer AC + IM_scratch*, colored in yellow), but also poses some challenges. Since the IM module does not import the previously learned weights, this strategy heavily prioritizes exploration. This proves to be beneficial in the

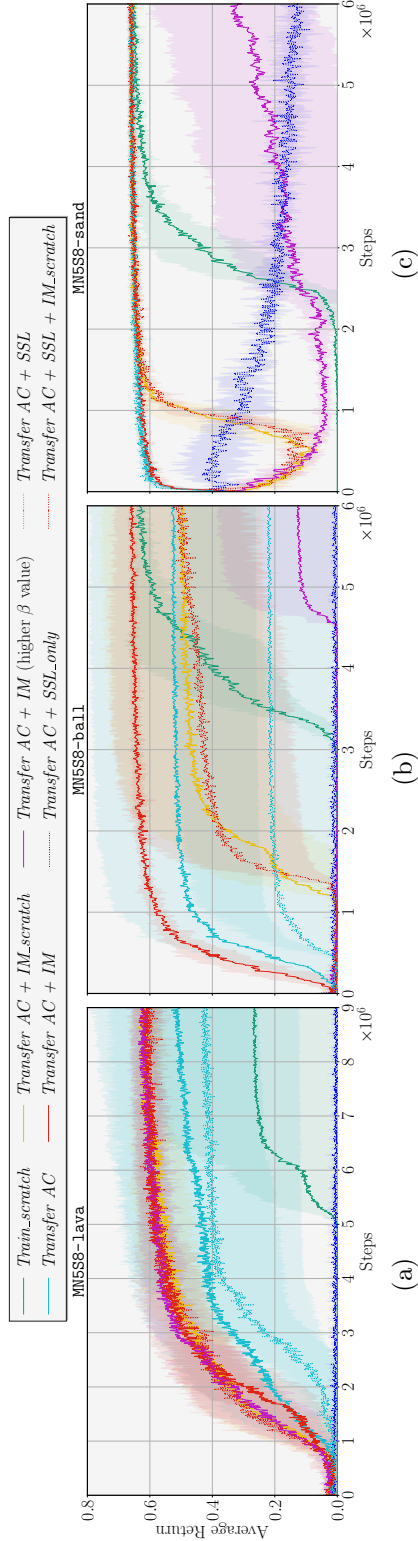


FIGURE 5.4: Average performance returns of various adaptation strategies employed by an agent initially trained on MN5S8 when faced with distinct OoD environments: (a) MN5S8-lava; (b) MN5S8-ball; and (c) MN5S8-sand. The curves illustrate the average and standard deviation over 5 runs.

case of **MN5S8-lava**, where exploration is necessary to navigate around the OoD elements and eventually reach the goal. However, in **MN5S8-sand** the impact of the OoD object (*sand*) on the agent’s behavior is less dramatic. Consequently, the agent does not require an intense exploration to adapt its learned policy, so the reinitialization of the IM module leads to a slower adaptation to the OoD environment. In **MN5S8-ball**, where a compromise between exploration and exploitation is needed, this approach is completely detrimental for the training process in some seeds while moderately reducing the training speed in the other seeds. These conclusions are buttressed by the fact that transferring the IM module and their weights performs best in the experiments (*Transfer AC + IM*, colored in red) corresponding to RQ2. Indeed, not only it solves effectively all OoD environments, but it adapts the knowledge of the agent faster than the rest of strategies.

Considering that inheriting the pretrained IM module could lead to insufficient novelty in the rewards, which might ultimately hinder the exploration, we conducted an additional experiment in which we increased the amplitude of the reward signal by increasing the β coefficient by an order of magnitude. This adaptation strategy is shown to be effective only in the case of **MN5S8-lava**, as it inhibits the agent’s learning in the other two OoD environments (catastrophically in the case of **MN5S8-ball**). Consequently, the better strategy when transferring the IM module along its weights is to maintain its β coefficient.

Regarding the use of an additional self-supervised loss, overall it can be concluded that it does not contribute to the adaptation of the agent. Transferring AC and using only the SSL loss (*Transfer AC + SSL_Only*, dotted blue curve) fails to train in any of the three environments; the addition of the RL loss (*Transfer AC + SSL*, dotted light blue curve) manages to learn although it is outperformed by using solely the RL loss (*Transfer AC*); lastly, the incorporation of the IM module (*Transfer AC + SSL + IM_scratch*, dotted red curve) attains almost the same results as using the IM from scratch (*Transfer AC + IM_scratch*). These results suggest that the addition of SSL does not yield any benefits to update the representation of the agent when facing new OoD elements under the considered environments. In fact, it hampers the performance with respect to not using it at all.

5.5 Conclusions

Safety is essential for deploying AI-based models in real-world applications. Among the research areas aiming at AI safety, OoD detection and adaptation has received little attention in RL. In this context, this study has explored the performance of several OoD detection techniques and adaptation strategies for RL agents operating on sparse RL environments, where we introduce elements that exhibit dynamics never seen by the agent previously. Our analyzed approaches leverage the architectural elements inside the agent that are necessary for learning to solve environments with sparse rewards:

- *OoD detection*: the proposed detectors include an approach based on the features extracted from the environment by the agent, and another method that relies on the forward dynamics of intrinsic motivation techniques, used by the agent to induce exploration in the absence of dense rewards.
- *OoD adaptation*: several strategies have been tested to adapt the behavior of the agent to the presence of OoD elements in the environment, varying the amount of knowledge that can be reused. Our baselines consider the transfer of the architectures and/or the parameters of the AC and/or the intrinsic motivation modules of the agent, as well as the inclusion of a self-supervised loss to foster learning new representations in the presence of OoD.

In the conducted experiments over procedurally generated environments provided by MiniGrid, an agent trained over a certain distribution of levels has been queried by a separate distribution of levels containing OoD elements, each impacting differently on the agent’s modeled behavior. In OoD detection, the proposed techniques report acceptable results when compared to off-the-shelf post-hoc detection techniques used for classification tasks. On the other hand, it has been proven that following the addition of OoD elements, the transfer of both the AC and the learned intrinsic motivation module is a good combination that maintains the agent’s policy exploitation capability, while also promoting exploration in the areas where OoD elements appear. Moreover, although SSL approaches have obtained good results in other environments to accelerate and build more robust representations, they seem to be insufficient if the agent actually has access to disentangled representations (a numerical identifier for each element in this case), and the effect of the OoD elements on the agent is a priori unknown. Furthermore, it has been observed that the adaptation is highly dependent on the considered new OoD elements.

Our preliminary findings open up exciting research directions for the near future. Firstly, we plan to improve the performance of the *AP* and *AP-A* detection techniques with a fine-grained characterization of the embedding space via clustering to assess observations that potentially belong to different distributions of the state. Moreover, following the insights obtained in our analysis in Figure 5.3, we will study the potential of combining the two proposed detection techniques (*FwD* and *AP*) to improve the detection across various distances between OoD objects and the agent. Finally, we aim to expand the coverage of our study towards high-dimensional state spaces, where SSL methods can be actually beneficial to disentangle representations [132, 147]. In this regard we plan to study not only which kind of adaptation strategy (or combinations of them) performs best, but also to explore their effectiveness under diverse dynamic changes in the environment, posing non-trivial challenges beyond the ones identified in this chapter.

Chapter 6

Concluding Remarks

This Thesis is driven by the observed gaps in research concerning the safety of AI models. Specifically, our objective has been to enhance model robustness, a crucial attribute for systems operating in open-world scenarios. The OoD detection framework reinforces model robustness by enabling the differentiation between known and unknown examples—distinguishing inputs that align with the training data distribution from those that are semantically divergent or originate from alternative distributions.

Within the realm of OoD detection, a significant oversight has been the limited focus on models, tasks, or learning paradigms beyond neural networks for supervised image classification, which constitutes the bulk of existing research. Consequently, our aim has been to investigate and expand upon less explored or entirely new frameworks within OoD detection. This Thesis contributions and conclusions are below remarked along with future possible works.

- **Chapter 3.** *Out-of-Distribution Detection in Image Classification for Spiking Neural Networks.* The so called third generation of neural networks, the SNNs, have not been yet studied under the OoD detection paradigm. For this purpose, we proposed a technique based on the operation of the SNNs by characterizing the internal activations of the hidden layers of the network in the form of spike count patterns, the SCP-based detector. Furthermore, a local explanation method has been devised to produce attribution maps revealing which parts of the input instance push most towards the detection of an example as an OoD sample. Our experiments also assessed whether the fusion of our proposed approach with other baseline OoD detection schemes can complement and boost the overall OoD detection capabilities. The conclusion drawn from this chapter can be summarized as follows:

- Our technique delivers competitive detection performance relative to post-hoc detection methods and yields particularly impressive results on neuromorphic datasets.
- Experiments with ensembles of OoD detection methods have demonstrated that integrating the proposed SCP-based detector with a logits-based approach achieves unparalleled performance relative to other technique pairings.

- The proposed local relevance attribution method effectively identifies artificially inserted perturbations in in-distribution samples, highlighting them as the image segments crucial for detecting them as OoD. Nonetheless, the limitations of this technique have been revealed and examined, including the deterioration in explanation quality as the depth of the network’s fully connected layers increases; the reliance of attributed relevance on the input features captured by the SNN model; and finally the variability within clusters formed from the spike counts of the input data.

As future technical extensions, we aim to broaden the application of the SCP detector and its foundational principles to different encoding methods suitable for spiking neural architectures, particularly those influencing the spike count strategy central to the SCP-based detector, e.g., the TTFS encoding [148]. Additionally, more effort needs to be invested on overcoming the limitations impacting the heatmaps generated by the relevance attribution technique.

• **Chapter 4.** *Out-of-Distribution Detection in Object Detection.* The current SOTA of unknown object detection is greatly dominated by the two-stage object detectors retrained with a pseudo-labeling strategy for unknowns. With the objective of studying this paradigm for the one-stage models, we developed an unknown or OoD detection technique—FMap—based on the internal activations of this kind of models, similarly to the previous chapter but attending to the particularities of these models (independent strides), and improved it by employing SDR techniques. Additionally, we designed an unsupervised method based on the feature maps extracted from the input that enhances the localization of unknown objects, coined as EUL. Furthermore, we studied the performance of the logits-based post-hoc detection methods and the fusion of all examined methods (as in the previous chapter), all for different possible configurations of the hyperparameters, finally comparing the best performing techniques against the SOTA. Our research found that:

- Each version of the FMap algorithm has its strengths and weaknesses. While the basic version maintains a balance between known and unknown detection, SDR achieves results with superior unknown detection and robustness. Conversely, EUL allows configurations of the vanilla method with low unknown detection performance to enhance this aspect without sacrificing the known detection capabilities.
- Post-hoc methods achieve remarkable performance, demonstrating the inherent ability of these one-stage object detection models to be robust. When testing various combinations of these methods among themselves and with FMap, it is clear that fusing similar types of methods does not yield significant improvements. However, merging logits-based methods with feature-based methods, our method in this case, leads to superior performance.
- Compared to the SOTA two-stage models, one-stage models enhance the detection capabilities for known objects, achieving more than acceptable

performance in detecting unknowns. Additionally, the techniques presented in this chapter do not require retraining.

For the future, the FMap method could be tested with other SDR techniques, as well as be improved by unifying the features of the different strides. Furthermore, we are very interested in exploring different possibilities of leveraging the feature maps extracted from every input. The EUL algorithm was one of those possibilities, so we will continue developing new techniques for exploiting the feature maps for the enhancement of the robustness.

• **Chapter 5.** *Out-of-Distribution Detection in Reinforcement Learning with Sparse Rewards.* In the realm of AI safety, OoD detection and adaptation have been relatively underexplored in RL. This chapter delves into several OoD detection techniques and adaptation strategies for RL agents operating in sparse RL environments, which involve introducing dynamics previously unseen by the agent. The OoD detection strategies we explored include one method that utilizes features extracted from the environment by the agent (*AP*), and another that capitalizes on the forward dynamics of intrinsic motivation techniques (*FwD*), which the agent employs to encourage exploration in the absence of dense rewards. For OoD adaptation, we evaluated various strategies to adjust the agent’s behavior to the presence of OoD elements, exploring different levels of knowledge transfer. These strategies ranged from transferring architectures and parameters of the actor-critic and intrinsic motivation modules, to implementing a self-supervised loss that aids in learning new representations when encountering OoD elements. The experiments were conducted in procedurally generated environments using MiniGrid, where an agent was trained on a specific distribution of levels and subsequently tested with a distinct distribution that included OoD elements. The main insights obtained from this part can be abbreviated as follows:

- In OoD detection, the proposed methods (*FwD* and *AP*) demonstrate satisfactory performance compared to standard post-hoc detection techniques traditionally used for classification tasks.
- As for the adaptation to the OoD elements, the combined transfer of the actor-critic and intrinsic motivation modules effectively preserves the agent’s policy exploitation abilities while enhancing exploration in areas with OoD elements. Additionally, while SSL approaches have shown promise in other settings for developing robust representations, they appear less effective when the agent possesses disentangled representations, especially when the impact of OoD elements on the agent’s behavior is initially unknown.

Our findings point to promising future technical extensions. Firstly, we plan to improve the performance of the *AP* and *AP-A* detection techniques with a fine-grained characterization of the embedding space via clustering to assess observations that potentially belong to different distributions of the state. Moreover, following the insights obtained in our analysis in Figure 5.3, we will study the potential of combining the two

proposed detection techniques (*FwD* and *AP*) to improve the detection across various distances between OoD objects and the agent. Finally, we aim to expand the coverage of our study towards high-dimensional state spaces, where SSL methods can be actually beneficial to disentangle representations [132, 147].

As subsequent extensions for this research, first we aim to enhance the performance of the *AP* and *AP-A* detection techniques by implementing a more detailed characterization of the embedding space using clustering. Additionally, it will be interesting to explore the combined potential of the two proposed detection methods (*FwD* and *AP*) to improve detection performance across varying distances between OoD objects and the agent. Lastly, new environments including high-dimensional state spaces must be addressed, where SSL techniques may play a crucial role in disentangling representations [132, 147].

6.1 Research Outcomes

As a result of the research conducted during the development of this PhD Thesis, several contributions have been achieved in the form of publications in conferences and journals, and public software repositories to stimulate follow-up studies in the future. Details are provided in what follows:

- **Journal publications:**

- **Aitor Martinez-Seras**, Javier Del Ser, Jesus L. Lobo, Pablo Garcia-Bringas, Nikola Kasabov, “A novel Out-of-Distribution detection approach for Spiking Neural Networks: Design, fusion, performance evaluation and explainability”. en. In: Information Fusion 100 (Dec. 2023), p. 101943. issn: 15662535. doi: [10.1016/j.inffus.2023.101943](https://doi.org/10.1016/j.inffus.2023.101943) (IF: 14.7, Q1, 4/197 ARTIFICIAL INTELLIGENCE).
- **Aitor Martinez-Seras**, Javier Del Ser, Alain Andres, Pablo Garcia-Bringas, “On the Inherent Robustness of One-Stage Object Detection Models against Open-World Out-of-Distribution Data”, Neural Networks, under review, 2024.

- **Conference contributions:**

- **Aitor Martinez-Seras**, Javier Del Ser, and Pablo Garcia-Bringas. “Can Post-hoc Explanations Effectively Detect Out-of-Distribution Samples?” In: 2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). ISSN: 1558-4739. July 2022, pp. 1–9. doi: [10.1109/FUZZ-IEEE55066.2022.9882726](https://doi.org/10.1109/FUZZ-IEEE55066.2022.9882726).
- **Aitor Martinez Seras**, Javier Del Ser, and Pablo Garcia-Bringas. “Efficient Object Detection in Autonomous Driving using Spiking Neural Networks: Performance, Energy Consumption Analysis, and Insights into Open-set Object Discovery”. In: 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC). pp. 5756–5763. doi: <https://doi.org/10.1109/ITSC57777.2023.10422139>.

- **Aitor Martinez Seras**, Alain Andres and Javier Del Ser, “Advancing towards Safe Reinforcement Learning over Sparse Environments with Out-of-Distribution Observations: Detection and Adaptation Strategies”, 2024 International Joint Conference on Neural Networks (IJCNN), Yokohama, Japan, 2024, pp. 1-10, doi: [10.1109/IJCNN60899.2024.10650670](https://doi.org/10.1109/IJCNN60899.2024.10650670).
- **Software repositories:**

A public software repository has been released for every technical contribution of this Thesis to foster follow-up studies in OoD research:

 - **Chapter 3.** *Out-of-Distribution Detection in Image Classification for Spiking Neural Networks*. URL: https://github.com/aitor-martinez-seras/OoD_on_SNNs. Accessed on 09/24/2024.
 - **Chapter 4.** *Out-of-Distribution Detection in Object Detection*. URL: https://github.com/aitor-martinez-seras/OoD_in_Object_Detection.git. Accessed on 09/24/2024.
 - **Chapter 5.** *Out-of-Distribution Detection in Reinforcement learning with Sparse Rewards*. URL: <https://github.com/aitor-martinez-seras/ood-sparse-rl>. Accessed on 09/24/2024.

6.1.1 Other Publications

Collaborations with other researchers have given rise to other publications that relate partially to OoD, are not considered as an original contribution of this Thesis:

- Javier Poyatos, Daniel Molina, **Aitor Martinez-Seras**, Javier Del Ser, and Francisco Herrera, (2023). “Multiobjective evolutionary pruning of Deep Neural Networks with Transfer Learning for improving their performance and robustness”. In: Applied Soft Computing, 147, 110757. ISSN 1568-4946. doi: [10.1016/j.asoc.2023.110757](https://doi.org/10.1016/j.asoc.2023.110757)
- **Aitor Martinez-Seras**, Javier Del Ser, and Pablo Garcia-Bringas. “Can Post-hoc Explanations Effectively Detect Out-of-Distribution Samples?” In: 2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). ISSN: 1558-4739. July 2022, pp. 1–9. doi: [10.1109/FUZZ-IEEE55066.2022.9882726](https://doi.org/10.1109/FUZZ-IEEE55066.2022.9882726).
- Javier Del Ser, **Aitor Martinez-Seras**, Miren Nekane Bilbao, Jesus L. Lobo, Ibai Laña and Francisco Herrera, “Balancing Performance, Efficiency and Robustness in Open-World Machine Learning via Evolutionary Multi-objective Model Compression”, 2024 International Joint Conference on Neural Networks (IJCNN), Yokohama, Japan, 2024, pp. 1-8, doi: [10.1109/IJCNN60899.2024.10651338](https://doi.org/10.1109/IJCNN60899.2024.10651338). (Best Paper Award, IJCNN 2024).
- Erik B. Terres-Escudero, Javier Del Ser, **Aitor Martinez-Seras**, Pablo Garcia-Bringas. (2024). “On the Robustness of Fully-Spiking Neural

Networks in Open-World Scenarios using Forward-Only Learning Algorithms”. IEEE Transactions on Neural Networks and Learning Systems, under review, 2024

- Javier Del Ser, **Aitor Martinez-Seras**, Ibai Laña, Miren Nekane Bilbao, Panagiotis Fafoutellis, Eleni I. Vlahogianni, “On the Connection between Neural Activations and Uncertainty in Object Detection Transformers Using Topological Data Analysis”, 2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC), Edmonton, Canada, September 24- 27, 2024.

6.2 Future Research Lines

This thesis concludes by outlining future research lines that have been identified as interesting directions during the PhD Thesis:

SNNs are emerging as a promising model for the future, primarily due to their low energy consumption when deployed on neuromorphic hardware [26, 27, 28] and their intrinsic ability to process temporal inputs. Consequently, advancing algorithms that enhance their robustness is crucial for ensuring their effective implementation. A particularly promising research direction involves the development of OoD detection algorithms that utilize more efficient spike-timing-based encoding methods [30]. Additionally, advancing OoD detection techniques for SNNs by directly testing them on neuromorphic hardware, facilitated by platforms such as Lava [149] which supports the Loihi chip [28], represents a significant stride towards their practical application. Finally, OoD detection with SNNs in temporal or spatio-temporal streaming data, such as brain EEG and fMRI data [150], seismic data [31] or financial data [151] represents an innovative area for further exploration.

Another exciting avenue opens up from our investigation of neural activations. Throughout our technical contributions, we have extensively utilized the activations from the feature extraction components of the model, namely the feature maps. These feature maps have proven invaluable for OoD detection, being leveraged in diverse ways. Consequently, recognizing their substantial potential to enhance model robustness, we plan on developing research focused on conducting more exploratory and comprehensive analyses in this area. In line with this, we have a forthcoming publication titled “*On the Connection between Neural Activations and Uncertainty in Object Detection Transformers using Topological Data Analysis*”, listed within the collaborations conducted during this Thesis in Subsection 6.1.1. This exploratory study examines the intricate relationship between the cohesiveness of neural activation patterns across known object categories, the model’s predictive confidence for such categories, and the inherent semantics of the objects.

A further compelling area we have identified involves exploring the RL paradigm for testing OoD detection algorithms, leveraging the variety of available environments to design targeted experiments. According to the

survey by [10], one of the foremost challenges in this domain is to establish robust evaluation and benchmarking frameworks that more closely mirror real-world conditions. We consider that RL environments and simulators, which offer highly customizable settings and the ability to generate distribution shifts on demand (often through procedural generation), could significantly advance these goals. That is, environments like CARLA [152], environments that closely emulate real-world scenarios, could be employed to not only generate realistic perturbations as demonstrated in [153], but also enhance the practical relevance of the research outcomes.

In conclusion, this Thesis underscores that a broadening of the OoD detection paradigm to encompass a wider range of tasks and model types is needed, as also the survey [10] highlights. A prospective avenue for research involves assessing the efficacy of post-hoc techniques within these emergent paradigms and developing methods specifically tailored to enhance their performance. Although substantial progress has been made in the field of OoD detection, it is imperative to push the boundaries of this discipline towards uncharted territories to foster continued innovation and improvement.

Appendix A

Results of Ensembles in Spiking Neural Networks

In this appendix, we show the tables with the results obtained by all the tested ensembles.

A.1 Results of ensembles in color datasets

TABLE A.1: OoD detection scores obtained by the individual methods and the ensembles over the color datasets using a convolutional SNN architecture.

ID dataset	OoD dataset	AUROC $\uparrow$						AUPR $\uparrow$						FPR95 $\downarrow$					
		SCP (ours)	ODIN	Energy	Ensemble ODIN-SCP	Ensemble Energy	Ensemble Energy-SCP	SCP (ours)	ODIN	Energy	Ensemble ODIN-SCP	Ensemble Energy	Ensemble Energy-SCP	SCP (ours)	ODIN	Energy	Ensemble ODIN-SCP	Ensemble Energy	Ensemble Energy-SCP
CIFAR10 (65% acc)	Food101	71.65	68.10	67.45	74.85	67.75	74.56	70.30	70.12	69.80	74.74	69.96	74.64	81.86	80.99	92.31	98.13	95.24	98.55
	LSUN (crop)	87.63	74.09	74.67	86.34	74.96	86.70	86.97	77.23	78.40	87.30	78.64	87.52	49.06	90.15	92.42	95.11	96.20	96.06
	LSUN (resize)	69.54	72.80	74.46	76.43	73.58	76.86	68.95	74.21	74.97	76.17	74.58	76.23	84.18	86.81	84.59	97.90	91.86	97.56
	Flower102	75.77	68.71	67.75	77.36	68.04	76.82	70.93	70.32	70.32	77.73	70.58	77.42	75.69	90.54	91.84	97.76	95.72	98.26
	Caltech101	67.02	65.88	66.47	70.40	65.96	70.81	64.47	67.90	68.58	69.66	68.11	70.02	83.81	91.53	91.79	98.61	95.62	98.64
	GTSRB	78.76	71.51	69.42	82.58	69.50	80.07	75.42	72.91	72.91	82.31	74.25	81.49	90.49	91.38	91.71	97.20	95.87	97.46
	SVHN	78.93	70.81	67.58	87.28	74.50	85.40	78.40	73.42	74.50	85.41	77.86	81.68	92.43	93.41	93.41	97.40	95.87	97.46
	ThyridImageNet	65.88	70.00	69.56	72.50	69.76	72.26	64.91	71.43	70.70	72.20	71.05	71.73	85.69	88.96	89.60	98.36	91.06	98.52
	EuroSAT	61.43	81.76	81.01	76.80	81.36	76.44	63.91	85.12	83.92	77.94	84.56	77.70	90.60	83.95	80.71	98.38	91.06	98.08
	DTD	61.73	63.97	65.64	65.38	64.46	66.73	59.24	64.94	67.58	64.68	65.81	66.30	84.89	92.71	92.55	98.62	96.60	98.67
SVHN (75% acc)	Food101	84.74	78.35	80.20	88.08	77.35	89.64	85.82	81.40	83.76	89.79	82.10	90.78	50.59	94.20	80.90	97.11	94.69	90.72
	LSUN (crop)	73.20	81.73	82.80	85.96	81.63	86.23	76.87	85.49	86.29	88.03	85.32	88.24	73.95	86.75	77.49	96.53	89.26	94.12
	LSUN (resize)	80.47	77.92	81.32	87.31	78.79	88.43	81.66	82.68	84.66	88.65	83.28	89.29	60.79	93.20	78.76	97.36	94.00	91.68
	Flower102	80.94	77.24	78.39	87.01	77.62	87.29	82.75	81.86	82.42	89.01	82.12	89.12	60.35	91.97	84.31	96.55	93.48	93.67
	Caltech101	81.82	77.02	79.88	86.77	77.82	87.77	82.23	81.85	83.45	87.90	82.38	88.46	56.24	93.60	82.10	97.25	94.48	91.99
	GTSRB	83.11	78.02	78.90	87.47	78.39	87.05	83.80	81.10	81.79	88.04	81.55	88.43	55.59	88.64	81.76	94.80	90.63	91.93
	CIFAR10	80.23	77.80	80.90	86.98	78.67	88.01	81.23	82.71	84.46	88.38	83.31	88.92	60.81	93.08	81.21	97.25	93.63	92.46
	ThyridImageNet	81.16	77.69	81.01	86.92	78.55	88.10	81.61	82.39	84.42	88.69	83.01	88.79	56.75	92.85	78.89	97.06	93.68	96.69
	EuroSAT	72.77	80.51	83.85	70.47	79.91	70.18	71.66	91.89	90.96	80.29	81.28	80.04	95.18	99.55	63.11	97.92	98.59	98.00
	DTD	68.09	83.81	84.08	72.74	84.01	72.65	67.29	86.87	87.15	69.36	87.07	69.82	83.03	78.72	73.67	99.80	82.60	94.39

A.2 OoD detection results for SNN with increased depth

TABLE A.2: OoD detection scores corresponding to the FC-SNN architecture with $L = 2$ fully connected layers.

ID dataset	OoD dataset	AUROC $\uparrow$				AUPR $\uparrow$				FPR95 $\downarrow$			
		SCP (ours)	MSP	ODIN	Energy	SCP (ours)	MSP	ODIN	Energy	SCP (ours)	MSP	ODIN	Energy
MNIST (97% acc)	FMNIST	85.76	85.57	87.16	86.40	86.79	83.79	86.59	87.27	60.73	58.38	59.08	67.25
	KMNIST	91.43	91.41	92.10	91.19	91.74	91.79	92.17	91.33	39.63	48.81	42.97	47.24
	Letters	87.96	87.26	87.99	86.89	87.80	86.20	86.97	86.37	49.71	58.87	55.34	62.66
	notMNIST	97.41	82.30	82.30	77.11	97.76	80.49	80.49	74.44	10.71	68.61	68.61	79.02
	omniglot	81.83	96.01	97.97	98.34	82.24	96.68	98.23	98.58	72.01	27.77	11.60	08.72
CIFAR10-BW		92.26	85.46	86.20	84.69	93.21	86.86	87.34	86.13	39.57	70.46	69.40	75.46
FMNIST (83% acc)	MNIST	94.60	83.26	93.70	95.69	93.45	85.73	94.26	95.85	20.57	66.17	33.27	21.27
	KMNIST	89.81	77.10	85.24	86.45	89.41	79.26	85.58	86.66	45.26	77.71	64.04	61.74
	Letters	94.06	81.00	90.97	92.66	94.14	83.75	91.86	93.23	28.31	71.37	46.87	38.29
	notMNIST	94.29	67.69	77.01	80.85	95.16	71.63	78.21	81.14	34.62	85.43	78.45	74.44
	omniglot	81.58	88.06	95.67	97.28	77.71	88.51	95.56	97.28	59.30	50.79	20.41	12.00
CIFAR10-BW		79.90	52.31	68.27	75.53	83.78	64.79	74.69	80.03	85.64	96.01	92.53	90.52
KMNIIST (89% acc)	MNIST	77.28	83.84	85.35	84.89	78.85	86.20	87.26	86.96	69.90	53.45	44.99	47.75
	FMNIST	78.36	84.41	84.41	77.08	80.04	87.68	87.68	81.86	68.49	55.48	55.48	71.94
	Letters	82.23	84.99	86.54	85.87	83.04	87.70	88.73	88.24	59.82	53.03	43.49	46.78
	notMNIST	95.68	79.61	79.61	75.00	96.21	82.14	82.14	78.10	12.30	63.42	63.42	75.35
	omniglot	72.45	92.49	95.45	96.29	74.39	94.01	96.43	97.16	78.65	25.38	09.77	06.99
CIFAR10-BW		91.66	89.67	90.74	87.95	92.77	91.92	92.70	91.14	30.70	36.63	27.44	41.20
Letters (89% acc)	MNIST	74.95	80.32	82.40	82.49	73.98	78.76	79.47	79.39	78.48	71.56	61.60	62.14
	FMNIST	87.85	81.80	81.80	66.07	88.17	83.07	83.07	70.52	47.97	76.42	76.42	88.31
	KMNIST	88.80	82.74	84.38	81.75	89.23	83.35	84.28	81.92	48.59	71.89	67.89	73.30
	notMNIST	98.12	74.57	74.57	66.70	98.14	72.27	72.27	64.35	07.90	80.19	80.19	82.95
	omniglot	84.65	93.75	97.66	97.84	86.91	94.23	97.88	98.05	73.25	33.94	12.99	11.47
CIFAR10-BW		98.21	71.32	71.32	50.54	98.41	75.89	75.89	60.61	07.46	91.11	91.11	98.78

TABLE A.3: OoD detection scores corresponding to the CNN-SNN architecture with $L = 2$ fully connected layers.

ID dataset	OoD dataset	AUROC $\uparrow$				AUPR $\uparrow$				FPR95 $\downarrow$			
		SCP (ours)	MSP	ODIN	Energy	SCP (ours)	MSP	ODIN	Energy	SCP (ours)	MSP	ODIN	Energy
MNIST (98% acc)	FMNIST	95.47	88.32	89.85	88.52	95.96	88.22	89.40	88.22	21.28	53.85	44.14	49.56
	KMNIST	95.55	90.83	91.70	91.28	96.02	90.62	91.13	90.65	20.42	46.69	39.71	40.91
	Letters	90.18	83.60	84.19	83.87	88.26	78.88	78.46	78.20	35.61	54.41	50.13	51.22
	notMNIST	96.27	85.94	86.80	86.79	96.70	83.07	84.11	84.11	16.44	53.70	50.28	51.14
	omniglot	95.59	91.67	93.18	93.36	96.10	91.81	93.00	93.16	20.81	42.94	33.48	32.51
CIFAR10-BW		97.09	94.66	96.03	94.96	97.72	95.20	96.20	95.19	13.20	30.87	19.62	26.71
FMNIST (82% acc)	MNIST	94.33	78.04	90.45	92.96	94.37	80.92	91.15	93.30	24.22	72.16	41.28	31.64
	KMNIST	92.76	78.67	88.27	90.85	93.29	81.05	88.55	90.68	34.72	70.88	47.62	37.46
	Letters	91.38	76.29	88.48	90.58	91.31	79.63	89.54	91.25	38.28	75.29	48.55	41.47
	notMNIST	93.68	77.25	86.14	86.14	93.44	77.25	83.64	83.31	26.95	74.95	52.48	52.68
	omniglot	94.68	79.66	89.85	92.47	94.96	82.78	90.65	92.90	24.31	73.41	45.13	35.59
CIFAR10-BW		85.68	82.16	88.50	89.58	85.63	84.33	89.08	90.02	58.43	66.38	48.84	44.91
KMNIIST (92% acc)	MNIST	89.19	88.92	90.57	90.49	90.80	90.51	91.80	91.69	28.61	31.88	23.16	23.79
	FMNIST	83.02	88.06	91.24	91.86	86.92	90.95	93.24	93.66	54.23	40.40	24.22	22.84
	Letters	92.07	88.72	90.81	90.95	93.61	90.61	92.18	92.28	18.26	34.76	20.09	22.90
	notMNIST	90.85	88.20	89.45	89.22	92.41	90.35	91.18	90.91	23.56	36.08	28.65	30.06
	omniglot	91.15	89.72	92.38	93.07	93.07	91.48	93.66	94.24	22.17	30.11	17.65	15.68
CIFAR10-BW		81.63	93.75	96.08	96.79	85.74	95.12	97.12	97.78	54.67	12.91	04.54	03.16
Letters (90% acc)	MNIST	76.04	81.15	84.48	84.38	73.44	81.31	83.94	84.05	74.44	73.56	62.23	64.79
	FMNIST	87.12	85.19	85.19	74.39	88.25	87.25	87.25	75.27	61.90	70.03	70.03	74.37
	KMNIST	91.26	85.31	87.47	84.11	91.90	85.88	87.09	83.60	44.83	66.51	55.99	64.45
	notMNIST	87.38	76.31	76.44	72.06	87.12	76.46	75.96	71.64	51.15	80.72	77.84	81.81
	omniglot	93.32	87.76	92.07	91.95	93.99	88.02	91.44	91.17	36.86	58.40	37.39	36.73
CIFAR10-BW		96.88	88.67	88.67	73.60	97.56	91.55	91.55	81.54	17.69	73.17	73.17	97.13

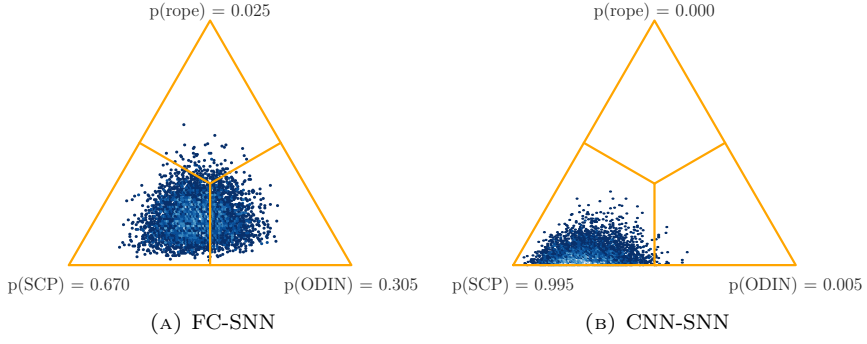


FIGURE A.1: Posterior probability resulting from the Bayesian signed tests computed over the paired AUROC scores attained by ODIN and the SCP-based detector with (a) the FC-SNN architecture; (b) the CNN-SNN architecture. In both cases $L = 2$.

Appendix B

Results on Unknown Object Detection Benchmark

This appendix contains the tables with all the numerical results of all the configurations tested in the Chapter [4](#).

Group	Method	Confidence Threshold	Distance Metric	Cluster Method	Fusion Strategy	mAP	U-AP	U-F1	U-PRE	mAP	U-AP	U-F1	U-PRE	AOSF	WI		
RQ1	FMap	0.050	L_1	One	-	0.662	0.152	0.269	0.578	0.127	0.435	0.022	0.111	0.216	368	0.141	
	RQ1	0.010	L_1	One	-	0.657	0.157	0.272	0.440	0.197	0.417	0.039	0.146	0.174	483	0.145	
	FMap	0.005	L_1	One	-	0.629	0.200	0.283	0.381	0.235	0.407	0.039	0.152	0.174	549	0.142	
	RQ1	0.050	L_2	One	-	0.668	0.158	0.237	0.592	0.148	0.423	0.023	0.119	0.225	0.081	319	0.139
	RQ1	0.010	L_2	One	-	0.650	0.204	0.287	0.451	0.211	0.410	0.040	0.148	0.177	0.127	470	0.141
	FMap	0.005	L_2	One	-	0.651	0.209	0.301	0.386	0.246	0.404	0.039	0.152	0.154	532	0.136	
	RQ1	0.050	L_2	KMeans ¹⁰	-	0.529	0.225	0.331	0.412	0.277	0.341	0.042	0.173	0.162	0.186	284	0.138
	RQ1	0.050	Costme	One	-	0.640	0.158	0.237	0.592	0.148	0.423	0.023	0.119	0.225	0.081	349	0.139
	FMap	0.010	Costme	One	-	0.650	0.204	0.287	0.451	0.211	0.410	0.040	0.148	0.177	0.127	470	0.141
	RQ1	0.010	Costme	One	-	0.651	0.209	0.301	0.386	0.246	0.404	0.039	0.152	0.154	532	0.136	
RQ2	FMap	0.050	Costme	KMeans ¹⁰	-	0.529	0.225	0.331	0.412	0.276	0.341	0.042	0.173	0.162	0.186	284	0.138
	RQ2	0.010	Costme	One	-	0.650	0.160	0.243	0.655	0.149	0.420	0.025	0.132	0.272	0.087	239	0.120
	FMap	0.010	Costme	HD BSCAN	-	0.577	0.219	0.327	0.431	0.264	0.385	0.045	0.169	0.173	0.165	349	0.135
	RQ2	0.050	Costme	One	-	0.656	0.165	0.231	0.732	0.137	0.417	0.021	0.100	0.232	0.064	298	0.135
	FMap	0.010	Costme	One	-	0.655	0.167	0.271	0.647	0.171	0.409	0.021	0.117	0.218	0.080	350	0.144
	RQ2	0.005	Costme	One	-	0.612	0.221	0.323	0.424	0.261	0.407	0.038	0.152	0.160	0.145	550	0.134
	FMap	0.010	Costme	One	-	0.507	0.164	0.248	0.713	0.150	0.411	0.022	0.114	0.240	0.075	270	0.127
	RQ2	0.010	Costme	One	-	0.581	0.223	0.333	0.480	0.255	0.395	0.040	0.161	0.185	0.143	424	0.133
	FMap	0.005	Costme	One	-	0.573	0.224	0.334	0.408	0.283	0.391	0.041	0.163	0.161	0.164	484	0.129
	RQ2	0.010	Costme	One	-	0.527	0.228	0.351	0.668	0.238	0.390	0.051	0.164	0.275	0.116	155	0.087
RQ3	FMap	0.010	Costme	One	-	0.477	0.292	0.389	0.453	0.352	0.327	0.062	0.191	0.181	0.203	206	0.086
	RQ3	0.010	Costme	KMeans ¹⁰	-	0.520	0.241	0.362	0.663	0.249	0.370	0.049	0.198	0.206	0.123	142	0.082
	FMap	0.050	Costme	One	-	0.367	0.241	0.384	0.381	0.287	0.357	0.047	0.180	0.235	0.145	166	0.085
	RQ3	0.010	Costme	KMeans ¹⁰	-	0.507	0.241	0.384	0.451	0.287	0.357	0.047	0.180	0.235	0.145	166	0.085
	FMap	0.050	Costme	HD BSCAN	-	0.471	0.296	0.392	0.427	0.363	0.323	0.061	0.190	0.174	0.209	185	0.089
	RQ3	0.010	Costme	HD BSCAN	-	0.533	0.241	0.351	0.667	0.238	0.376	0.051	0.165	0.276	0.118	155	0.089
	FMap	0.050	Costme	HD BSCAN	-	0.519	0.241	0.357	0.687	0.278	0.366	0.049	0.178	0.244	0.140	180	0.092
	RQ3	0.010	Costme	HD BSCAN	-	0.478	0.295	0.388	0.432	0.352	0.329	0.062	0.190	0.244	0.140	211	0.091
	FMap	0.050	Costme	One	-	0.682	0.161	0.254	0.239	0.270	0.435	0.031	0.119	0.099	0.180	368	0.141
	RQ3	0.010	Costme	One	-	0.637	0.203	0.277	0.637	0.203	0.277	0.031	0.132	0.101	0.190	483	0.145
RQ4	FMap	0.005	Costme	One	-	0.629	0.206	0.282	0.233	0.358	0.407	0.040	0.134	0.098	0.212	536	0.142
	RQ4	0.010	Costme	KMeans ¹⁰	-	0.529	0.225	0.287	0.234	0.371	0.341	0.051	0.144	0.103	0.240	284	0.118
	FMap	0.010	Costme	KMeans ¹⁰	-	0.529	0.225	0.287	0.234	0.371	0.341	0.051	0.144	0.103	0.240	284	0.118
	RQ4	0.010	Costme	KMeans ¹⁰	-	0.529	0.225	0.287	0.234	0.371	0.341	0.051	0.144	0.103	0.240	284	0.118
	FMap	0.050	Costme	HD BSCAN	-	0.577	0.217	0.282	0.234	0.356	0.385	0.046	0.139	0.102	0.218	249	0.125
	RQ4	0.050	Costme	HD BSCAN	-	0.719	0.237	0.280	0.465	0.298	0.446	0.083	0.156	0.237	0.133	123	0.073
	FMap	0.010	Costme	HD BSCAN	-	0.567	0.289	0.397	0.463	0.307	0.371	0.048	0.139	0.222	0.171	79	0.057
	RQ4	0.010	Costme	HD BSCAN	-	0.775	0.209	0.314	0.354	0.282	0.301	0.113	0.160	0.152	0.169	445	0.105
	FMap	0.005	Costme	HD BSCAN	-	0.758	0.202	0.287	0.416	0.280	0.313	0.112	0.149	0.178	0.188	591	0.126
	RQ4	0.001	Costme	HD BSCAN	-	0.756	0.207	0.282	0.276	0.288	0.316	0.112	0.150	0.125	0.188	822	0.133
RQ5	FMap	0.010	Costme	HD BSCAN	-	0.765	0.275	0.362	0.417	0.330	0.433	0.107	0.188	0.197	0.179	262	0.092
	RQ5	0.010	Costme	HD BSCAN	-	0.765	0.275	0.362	0.417	0.330	0.433	0.107	0.188	0.197	0.179	262	0.092
	FMap	0.050	Costme	One	-	0.702	0.289	0.389	0.567	0.308	0.431	0.077	0.188	0.228	0.160	106	0.090
	RQ5	0.050	Costme	One	-	0.702	0.289	0.389	0.567	0.308	0.431	0.077	0.188	0.228	0.160	106	0.090
	FMap	0.010	Costme	One	-	0.702	0.289	0.389	0.405	0.381	0.437	0.094	0.184	0.157	0.223	107	0.089
	RQ5	0.010	Costme	One	-	0.704	0.270	0.375	0.397	0.355	0.456	0.109	0.179	0.158	0.206	150	0.089
	FMap	0.010	Costme	One	-	0.704	0.270	0.375	0.397	0.355	0.456	0.109	0.179	0.158	0.206	150	0.089
	RQ5	0.010	Costme	One	-	0.704	0.270	0.375	0.397	0.355	0.456	0.109	0.179	0.158	0.206	150	0.089
	FMap	0.010	Costme	One	-	0.727	0.271	0.376	0.397	0.357	0.450	0.115	0.181	0.159	0.211	144	0.079
	RQ5	0.010	Costme	One	-	0.727	0.271	0.376	0.397	0.357	0.450	0.115	0.181	0.159	0.211	144	0.079

TABLE B.1: Results from best method of RQ1, RQ2 and RQ3

Bibliography

- [1] High-level expert group on artificial intelligence. “Ethics guidelines for trustworthy AI”. In: *European Commission* 6 (2019). Publisher: European Commission. Available at: <https://ec.europa.eu/digital-single>.
- [2] Natalia Díaz-Rodríguez et al. “Connecting the dots in trustworthy Artificial Intelligence: From AI principles, ethics, and key requirements to responsible AI systems and regulation”. In: *Information Fusion* 99 (2023). Publisher: Elsevier, p. 101896.
- [3] Yoshua Bengio et al. “Managing extreme AI risks amid rapid progress”. In: *Science* 384.6698 (2024). Publisher: American Association for the Advancement of Science, pp. 842–845.
- [4] Elham Tabassi. “Artificial intelligence risk management framework (AI RMF 1.0)”. In: (2023). Publisher: Elham Tabassi.
- [5] Gregory Ditzler et al. “Learning in nonstationary environments: A survey”. In: *IEEE Computational Intelligence Magazine* 10.4 (2015). Publisher: IEEE, pp. 12–25.
- [6] Da-Wei Zhou et al. “Class-incremental learning: A survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024). Publisher: IEEE.
- [7] Jesse Read and Indrė Žliobaitė. “Learning from data streams: An overview and update”. In: *arXiv preprint arXiv:2212.14720* (2022).
- [8] Dan Hendrycks and Kevin Gimpel. “A baseline for detecting misclassified and out-of-distribution examples in neural networks”. In: *arXiv preprint arXiv:1610.02136* (2016).
- [9] European Commission. “Proposal For A Regulation Of The European Parliament And Of The Council Laying Down Harmonised Rules On Artificial Intelligence (Artificial Intelligence Act) And Amending Certain Union Legislative Acts”. In: *2021/0106(COD)* (2021).
- [10] Jingkang Yang et al. “Generalized out-of-distribution detection: A survey”. In: *International Journal of Computer Vision* (2024). Publisher: Springer, pp. 1–28.
- [11] Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. “Recent advances in open set recognition: A survey”. In: *IEEE transactions on pattern analysis and machine intelligence* 43.10 (2020). Publisher: IEEE, pp. 3614–3631.

- [12] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. “Enhancing the reliability of out-of-distribution image detection in neural networks”. In: *arXiv preprint arXiv:1706.02690* (2017).
- [13] Kimin Lee et al. “A simple unified framework for detecting out-of-distribution samples and adversarial attacks”. In: *Advances in neural information processing systems* 31 (2018).
- [14] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. “Deep anomaly detection with outlier exposure”. In: *arXiv preprint arXiv:1812.04606* (2018).
- [15] Weitang Liu et al. “Energy-based out-of-distribution detection”. In: *Advances in neural information processing systems* 33 (2020), pp. 21464–21475.
- [16] Ziqian Lin, Sreya Dutta Roy, and Yixuan Li. “Mood: Multi-level out-of-distribution detection”. In: *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*. 2021, 15313–15323.
- [17] Haoran Wang et al. “Can multi-label classification networks know what they don’t know?” In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 29074–29087.
- [18] Rui Huang, Andrew Geng, and Yixuan Li. “On the importance of gradients for detecting distributional shifts in the wild”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 677–689.
- [19] Yiyu Sun, Chuan Guo, and Yixuan Li. “React: Out-of-distribution detection with rectified activations”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 144–157.
- [20] Yue Song, Nicu Sebe, and Wei Wang. “Rankfeat: Rank-1 feature removal for out-of-distribution detection”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 17885–17898.
- [21] Jinsong Zhang et al. “Out-of-distribution detection based on in-distribution data patterns memorization with modern hopfield energy”. In: *The Eleventh International Conference on Learning Representations*. 2022.
- [22] Andrija Djurisić et al. “Extremely simple activation shaping for out-of-distribution detection”. In: *arXiv preprint arXiv:2209.09858* (2022).
- [23] Kai Xu et al. “Scaling for training time and post-hoc out-of-distribution detection enhancement”. In: *arXiv preprint arXiv:2310.00227* (2023).
- [24] Jingkang Yang et al. “Openood: Benchmarking generalized out-of-distribution detection”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 32598–32611.
- [25] Jingyang Zhang et al. “Openood v1. 5: Enhanced benchmark for out-of-distribution detection”. In: *arXiv preprint arXiv:2306.09301* (2023).

- [26] Filipp Akopyan et al. “Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip”. In: *IEEE transactions on computer-aided design of integrated circuits and systems* 34.10 (2015). Publisher: IEEE, pp. 1537–1557.
- [27] Steve B Furber et al. “The spinnaker project”. In: *Proceedings of the IEEE* 102.5 (2014). Publisher: IEEE, pp. 652–665.
- [28] Mike Davies et al. “Loihi: A neuromorphic manycore processor with on-chip learning”. In: *Ieee Micro* 38.1 (2018). Publisher: IEEE, pp. 82–99.
- [29] Xiangwen Wang, Xianghong Lin, and Xiaochao Dang. “Supervised learning in spiking neural networks: A review of algorithms and evaluations”. In: *Neural Networks* 125 (2020). Publisher: Elsevier, pp. 258–280.
- [30] Daniel Auge et al. “A survey of encoding techniques for signal processing in spiking neural networks”. In: *Neural Processing Letters* 53.6 (2021). Publisher: Springer, pp. 4693–4710.
- [31] Nikola K Kasabov. *Time-space, spiking neural networks and brain-inspired artificial intelligence*. Springer, 2019.
- [32] Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.
- [33] Eugene M Izhikevich. “Simple model of spiking neurons”. In: *IEEE Transactions on neural networks* 14.6 (2003). Publisher: IEEE, 1569–1572.
- [34] Jesus L Lobo et al. “Spiking neural networks and online learning: An overview and perspectives”. In: *Neural Networks* 121 (2020). Publisher: Elsevier, pp. 88–100.
- [35] Natalia Caporale and Yang Dan. “Spike timing-dependent plasticity: a Hebbian learning rule”. In: *Annu. Rev. Neurosci.* 31.1 (2008). Publisher: Annual Reviews, pp. 25–46.
- [36] Sander M Bohte, Joost N Kok, and Han La Poutre. “Error back-propagation in temporally encoded networks of spiking neurons”. In: *Neurocomputing* 48.1-4 (2002). Publisher: Elsevier, pp. 17–37.
- [37] Olaf Booiij and Hieu tat Nguyen. “A gradient descent rule for spiking neurons emitting multiple spikes”. In: *Information Processing Letters* 95.6 (2005). Publisher: Elsevier, pp. 552–558.
- [38] Ammar Mohemmed et al. “Span: Spike pattern association neuron for learning spatio-temporal spike patterns”. In: *International journal of neural systems* 22.04 (2012). Publisher: World Scientific, p. 1250012.
- [39] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. “Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks”. In: *IEEE Signal Processing Magazine* 36.6 (2019). Publisher: IEEE, pp. 51–63.

- [40] Friedemann Zenke and Surya Ganguli. “Superspike: Supervised learning in multilayer spiking neural networks”. In: *Neural computation* 30.6 (2018). Publisher: MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info ..., pp. 1514–1541.
- [41] Zhengxia Zou et al. “Object detection in 20 years: A survey”. In: *Proceedings of the IEEE* 111.3 (2023). Publisher: IEEE, pp. 257–276.
- [42] Ross Girshick et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.
- [43] Kaiming He et al. “Spatial pyramid pooling in deep convolutional networks for visual recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 37.9 (2015). Publisher: IEEE, pp. 1904–1916.
- [44] R Girshick. “Fast r-cnn”. In: *arXiv preprint arXiv:1504.08083* (2015).
- [45] Shaoqing Ren et al. “Faster R-CNN: Towards real-time object detection with region proposal networks”. In: *IEEE transactions on pattern analysis and machine intelligence* 39.6 (2016). Publisher: IEEE, pp. 1137–1149.
- [46] Joseph Redmon et al. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.
- [47] Juan Terven, Diana-Margarita Córdova-Esparza, and Julio Alejandro Romero-González. “A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas”. In: *Machine Learning and Knowledge Extraction* 5.4 (2023). Publisher: MDPI, pp. 1680–1716.
- [48] Ao Wang et al. “Yolov10: Real-time end-to-end object detection”. In: *arXiv preprint arXiv:2405.14458* (2024).
- [49] Wei Liu et al. “Ssd: Single shot multibox detector”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I* 14. Springer, 2016, pp. 21–37.
- [50] T-YLPG Ross and GKHP Dollár. “Focal loss for dense object detection”. In: *proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2980–2988.
- [51] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. “Objects as points”. In: *arXiv preprint arXiv:1904.07850* (2019).
- [52] Nicolas Carion et al. “End-to-end object detection with transformers”. In: *European conference on computer vision*. Springer, 2020, pp. 213–229.
- [53] Tsung-Yi Lin et al. “Feature pyramid networks for object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.

- [54] Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*. Vol. 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.
- [55] John Schulman et al. “Trust Region Policy Optimization”. In: *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. 2017.
- [56] Christopher JCH Watkins and Peter Dayan. “Q-learning”. In: *Machine learning* 8 (1992). Publisher: Springer, pp. 279–292.
- [57] Volodymyr Mnih et al. “Human-level control through deep reinforcement learning”. In: *nature* 518.7540 (2015). Publisher: Nature Publishing Group UK London, pp. 529–533.
- [58] TP Lillicrap. “Continuous control with deep reinforcement learning”. In: *arXiv preprint arXiv:1509.02971* (2015).
- [59] John Schulman et al. “High-dimensional continuous control using generalized advantage estimation”. In: *arXiv preprint arXiv:1506.02438* (2015).
- [60] John Schulman et al. “Proximal policy optimization algorithms”. In: *arXiv preprint arXiv:1707.06347* (2017).
- [61] Volodymyr Mnih et al. “Asynchronous Methods for Deep Reinforcement Learning”. In: *Proceedings of the 33rd International Conference on Machine Learning (ICML)*. 2016.
- [62] John Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. In: *nature* 596.7873 (2021). Publisher: Nature Publishing Group, pp. 583–589.
- [63] Tom B Brown. “Language models are few-shot learners”. In: *arXiv preprint arXiv:2005.14165* (2020).
- [64] Alejandro Barredo Arrieta et al. “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI”. In: *Information fusion* 58 (2020). Publisher: Elsevier, pp. 82–115.
- [65] Amina Adadi and Mohammed Berrada. “Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)”. In: *IEEE access* 6 (2018). Publisher: IEEE, pp. 52138–52160.
- [66] Erico Tjoa and Cuntai Guan. “A survey on explainable artificial intelligence (xai): Toward medical xai”. In: *IEEE transactions on neural networks and learning systems* 32.11 (2020). Publisher: IEEE, pp. 4793–4813.
- [67] Samanwoy Ghosh-Dastidar and Hojjat Adeli. “Third generation neural networks: Spiking neural networks”. In: *Advances in computational intelligence*. Springer, 2009, pp. 167–178.
- [68] Lukas Paulun, Anne Wendt, and Nikola Kasabov. “A retinotopic spiking neural network system for accurate recognition of moving objects using neucube and dynamic vision sensors”. In: *Frontiers in Computational Neuroscience* 12 (2018). Publisher: Frontiers Media SA, p. 42.

- [69] Abeegithan Jeyasothy et al. “A novel method for extracting interpretable knowledge from a spiking neural classifier with time-varying synaptic weights”. In: *arXiv preprint arXiv:1904.11367* (2019).
- [70] Abeegithan Jeyasothy, Suresh Sundaram, and Narasimhan Sundararajan. “SEFRON: A new spiking neuron model with time-varying synaptic efficacy function for pattern classification”. In: *IEEE transactions on neural networks and learning systems* 30.4 (2018). Publisher: IEEE, pp. 1231–1240.
- [71] Snjezana Soltic and Nikola Kasabov. “Knowledge extraction from evolving spiking neural networks with rank order population coding”. In: *International Journal of Neural Systems* 20.06 (2010). Publisher: World Scientific, pp. 437–445.
- [72] Maryam Doborjeh et al. “Deep learning of explainable EEG patterns as dynamic spatiotemporal clusters and rules in a brain-inspired spiking neural network”. In: *Sensors* 21.14 (2021). Publisher: MDPI, p. 4900.
- [73] Kaushalya Kumarasinghe, Nikola Kasabov, and Denise Taylor. “Deep learning and deep knowledge representation in spiking neural networks for brain-computer interfaces”. In: *Neural Networks* 121 (2020). Publisher: Elsevier, pp. 169–185.
- [74] Nikola K Kasabov. “NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data”. In: *Neural Networks* 52 (2014). Publisher: Elsevier, pp. 62–76.
- [75] Youngeun Kim and Priyadarshini Panda. “Visual explanations from spiking neural networks using inter-spike intervals”. In: *Scientific reports* 11.1 (2021). Publisher: Nature Publishing Group UK London, p. 19037.
- [76] Elisa Nguyen. “Temporal spike attribution: A local feature-based explanation for temporally coded spiking neural networks”. MA thesis. University of Twente, 2021.
- [77] Yujie Wu et al. “Direct training for spiking neural networks: Faster, larger, better”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. Issue: 01. 2019, pp. 1311–1318.
- [78] Ali Samadzadeh et al. “Convolutional spiking neural networks for spatio-temporal feature extraction”. In: *Neural Processing Letters* 55.6 (2023). Publisher: Springer, pp. 6979–6995.
- [79] Shuang Lian et al. “Training deep convolutional spiking neural networks with spike probabilistic global pooling”. In: *Neural Computation* 34.5 (2022). Publisher: MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , pp. 1170–1188.
- [80] Ruthvik Vaila. *Deep convolutional spiking neural networks for image classification*. Boise State University, 2021.

- [81] David Heeger et al. “Poisson model of spike generation”. In: *Hand-out, University of Stanford* 5.1-13 (2000), p. 76.
- [82] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. “On the surprising behavior of distance metrics in high dimensional space”. In: *Database theory—ICDT 2001: 8th international conference London, UK, January 4–6, 2001 proceedings* 8. Springer, 2001, pp. 420–434.
- [83] Ángel M García-Vico and Francisco Herrera. “A preliminary analysis on software frameworks for the development of spiking neural networks”. In: *International Conference on Hybrid Artificial Intelligence Systems*. Springer, 2021, pp. 564–575.
- [84] Wulfram Gerstner et al. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.
- [85] Christian-Gernot Pehle and Jens Egholm Pedersen. “Norse-A deep learning library for spiking neural networks”. In: *Zenodo* (2021).
- [86] Li Deng. “The mnist database of handwritten digit images for machine learning research [best of the web]”. In: *IEEE signal processing magazine* 29.6 (2012). Publisher: IEEE, pp. 141–142.
- [87] Han Xiao, Kashif Rasul, and Roland Vollgraf. “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms”. In: *arXiv preprint arXiv:1708.07747* (2017).
- [88] Tarin Clanuwat et al. “Deep learning for classical japanese literature”. In: *arXiv preprint arXiv:1812.01718* (2018).
- [89] Gregory Cohen et al. “EMNIST: Extending MNIST to handwritten letters”. In: *2017 international joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 2921–2926.
- [90] Garrick Orchard et al. “Converting static image datasets to spiking neuromorphic datasets using saccades”. In: *Frontiers in neuroscience* 9 (2015). Publisher: Frontiers Media SA, p. 437.
- [91] Arnon Amir et al. “A low power, fully event-based gesture recognition system”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7243–7252.
- [92] Yaroslav Bulatov. “Notmnist dataset”. In: *Google (Books/OCR), Tech. Rep.* 2 (2011), p. 4.
- [93] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. “Human-level concept learning through probabilistic program induction”. In: *Science* 350.6266 (2015). Publisher: American Association for the Advancement of Science, pp. 1332–1338.
- [94] Alex Krizhevsky, Geoffrey Hinton, et al. “Learning multiple layers of features from tiny images”. In: (2009). Publisher: Toronto, ON, Canada.
- [95] Hongmin Li et al. “Cifar10-dvs: an event-stream dataset for object classification”. In: *Frontiers in neuroscience* 11 (2017). Publisher: Frontiers Media SA, p. 309.

- [96] Ganchao Tan et al. “Multi-grained spatio-temporal features perceived network for event-based lip-reading”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 20094–20103.
- [97] Norman Mu and Justin Gilmer. “Mnist-c: A robustness benchmark for computer vision”. In: *arXiv preprint arXiv:1906.02337* (2019).
- [98] Janez Demšar. “Statistical comparisons of classifiers over multiple data sets”. In: *The Journal of Machine learning research* 7 (2006). Publisher: JMLR. org, pp. 1–30.
- [99] Alessio Benavoli et al. “Time for a change: a tutorial for comparing multiple classifiers through Bayesian analysis”. In: *Journal of Machine Learning Research* 18.77 (2017), pp. 1–36.
- [100] KJ Joseph et al. “Towards open world object detection”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 5830–5840.
- [101] Deepak Kumar Singh et al. “Order: Open world object detection on road scenes”. In: *Proc. NeurIPS Workshops*. Vol. 1. Issue: 2. 2021, p. 3.
- [102] Xiaowei Zhao et al. “Revisiting open world object detection”. In: *IEEE Transactions on Circuits and Systems for Video Technology* (2023). Publisher: IEEE.
- [103] Yanghao Wang et al. “Random boxes are open-world object detectors”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 6233–6243.
- [104] Xuefeng Du et al. “Vos: Learning what you don’t know by virtual outlier synthesis”. In: *arXiv preprint arXiv:2202.01197* (2022).
- [105] Xuefeng Du et al. “Unknown-aware object detection: Learning what you don’t know from videos in the wild”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 13678–13688.
- [106] Akshay Dhamija et al. “The overlooked elephant of object detection: Open set”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pp. 1021–1030.
- [107] Mark Everingham et al. “The pascal visual object classes (voc) challenge”. In: *International journal of computer vision* 88 (2010). Publisher: Springer, pp. 303–338.
- [108] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.
- [109] Akshita Gupta et al. “Ow-detr: Open-world detection transformer”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 9235–9244.

- [110] Jasper RR Uijlings et al. “Selective search for object recognition”. In: *International journal of computer vision* 104 (2013). Publisher: Springer, pp. 154–171.
- [111] Zhiheng Wu et al. “UC-OWOD: Unknown-classified open world object detection”. In: *European Conference on Computer Vision*. Springer, 2022, pp. 193–210.
- [112] Wenteng Liang et al. “Unknown Sniffer for Object Detection: Don’t Turn a Blind Eye to Unknown Objects”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 3230–3239.
- [113] Kaiming He et al. “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.
- [114] Nobuyuki Otsu et al. “A threshold selection method from gray-level histograms”. In: *Automatica* 11.285-296 (1975), pp. 23–27.
- [115] Dimity Miller et al. “Dropout sampling for robust object detection in open-set conditions”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3243–3249.
- [116] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. *Ultralytics YOLO*. Jan. 2023.
- [117] Benjamin Szubert et al. “Structure-preserving visualisation of high dimensional single-cell datasets”. In: *Scientific reports* 9.1 (2019). Publisher: Nature Publishing Group UK London, p. 8914.
- [118] Felix Berkenkamp et al. “Safe model-based reinforcement learning with stability guarantees”. In: *Advances in neural information processing systems* 30 (2017).
- [119] Mohammed Alshiekh et al. “Safe reinforcement learning via shielding”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. Issue: 1. 2018.
- [120] Krishnan Srinivasan et al. “Learning to be safe: Deep rl with a safety critic”. In: *arXiv preprint arXiv:2010.14603* (2020).
- [121] Tom Haider et al. “Out-of-distribution detection for reinforcement learning agents with probabilistic dynamics models”. In: *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*. 2023, pp. 851–859.
- [122] Frederik Träuble et al. “Representation Learning for Out-of-distribution Generalization in Reinforcement Learning”. In: *ICML 2021 Workshop on Unsupervised Reinforcement Learning*. 2021.
- [123] Mohamad H Danesh and Alan Fern. “Out-of-Distribution Dynamics Detection: RL-Relevant Benchmarks and Results”. In: *arXiv preprint arXiv:2107.04982* (2021).
- [124] Aaqib Parvez Mohammed and Matias Valdenegro-Toro. “Benchmark for out-of-distribution detection in deep reinforcement learning”. In: *arXiv preprint arXiv:2112.02694* (2021).

- [125] Miruna Pislari et al. “When should agents explore?” In: *arXiv preprint arXiv:2108.11811* (2021).
- [126] Zhuangdi Zhu et al. “Transfer learning in deep reinforcement learning: A survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023). Publisher: IEEE.
- [127] Aravind Rajeswaran et al. “Learning complex dexterous manipulation with deep reinforcement learning and demonstrations”. In: *arXiv preprint arXiv:1709.10087* (2017).
- [128] Takayuki Osa et al. “An algorithmic perspective on imitation learning”. In: *Foundations and Trends® in Robotics* 7.1-2 (2018). Publisher: Now Publishers, Inc., pp. 1–179.
- [129] Alain Andres et al. “Using offline data to speed-up reinforcement learning in procedurally generated environments”. In: *arXiv preprint arXiv:2304.09825* (2023).
- [130] Ishita Mediratta et al. “The Generalization Gap in Offline Reinforcement Learning”. In: *arXiv preprint arXiv:2312.05742* (2023).
- [131] Max Jaderberg et al. “Reinforcement learning with unsupervised auxiliary tasks”. In: *arXiv preprint arXiv:1611.05397* (2016).
- [132] Adam Stooke et al. “Decoupling representation learning from reinforcement learning”. In: *International conference on machine learning*. PMLR, 2021, pp. 9870–9879.
- [133] Nicklas Hansen et al. “Self-supervised policy adaptation during deployment”. In: *arXiv preprint arXiv:2007.04309* (2020).
- [134] Max Schwarzer et al. “Data-efficient reinforcement learning with self-predictive representations”. In: *arXiv preprint arXiv:2007.05929* (2020).
- [135] Marc Bellemare et al. “Unifying count-based exploration and intrinsic motivation”. In: *Advances in neural information processing systems* 29 (2016).
- [136] Deepak Pathak et al. “Curiosity-driven exploration by self-supervised prediction”. In: *International conference on machine learning*. PMLR, 2017, pp. 2778–2787.
- [137] Yuri Burda et al. “Exploration by random network distillation”. In: *arXiv preprint arXiv:1810.12894* (2018).
- [138] Roberta Raileanu and Tim Rocktäschel. “Ride: Rewarding impact-driven exploration for procedurally-generated environments”. In: *arXiv preprint arXiv:2002.12292* (2020).
- [139] Yannis Flet-Berliac et al. “Adversarially guided actor-critic”. In: *arXiv preprint arXiv:2102.04376* (2021).
- [140] Alain Andres, Esther Villar-Rodriguez, and Javier Del Ser. “An evaluation study of intrinsic motivation techniques applied to reinforcement learning over hard exploration environments”. In: *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer, 2022, pp. 201–220.

- [141] Jitendra Parmar et al. “Open-world machine learning: applications, challenges, and opportunities”. In: *ACM Computing Surveys* 55.10 (2023). Publisher: ACM New York, NY, pp. 1–37.
- [142] Charles Packer et al. “Assessing generalization in deep reinforcement learning”. In: *arXiv preprint arXiv:1810.12282* (2018).
- [143] Robert Kirk et al. “A survey of zero-shot generalisation in deep reinforcement learning”. In: *Journal of Artificial Intelligence Research* 76 (2023), pp. 201–264.
- [144] Pamul Yadav et al. “A survey on deep reinforcement learning-based approaches for adaptation and generalization”. In: *arXiv preprint arXiv:2202.08444* (2022).
- [145] Andreas Sedlmeier et al. “Uncertainty-based out-of-distribution classification in deep reinforcement learning”. In: *arXiv preprint arXiv:2001.00496* (2019).
- [146] Maxime Chevalier-Boisvert et al. “Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks”. In: *Advances in Neural Information Processing Systems* 36 (2024).
- [147] Mhairi Dunion et al. “Temporal disentanglement of representations for improved generalisation in reinforcement learning”. In: *arXiv preprint arXiv:2207.05480* (2022).
- [148] Roland S Johansson and Ingvars Birznieks. “First spikes in ensembles of human tactile afferents code complex spatial fingertip events”. In: *Nature neuroscience* 7.2 (2004). Publisher: Nature Publishing Group US New York, pp. 170–177.
- [149] MGK Williams, Philipp Plank, and Sumit Bam Shrestha. *Lava-a software framework for neuromorphic computing*. 2023.
- [150] Nikola K Kasabov. *Spiking neural networks for deep learning and knowledge representation*. 2019.
- [151] Iman Abouhassan et al. “Why Use Evolving Neuro-Fuzzy and Spiking Neural Networks for incremental and explainable learning of time series? A case study on predictive modelling of trade imports and outlier detection”. In: *2022 IEEE 11th International Conference on Intelligent Systems (IS)*. IEEE, 2022, pp. 1–7.
- [152] Alexey Dosovitskiy et al. “CARLA: An open urban driving simulator”. In: *Conference on robot learning*. PMLR, 2017, pp. 1–16.
- [153] Raul Sena Ferreira et al. “Simood: Evolutionary testing simulation with out-of-distribution images”. In: *2022 IEEE 27th Pacific Rim International Symposium on Dependable Computing (PRDC)*. IEEE, 2022, pp. 68–77.