Improving Artificial Intelligence Reliability through Out-of-Distribution and Misclassification Detection

Amélioration de la Fiabilité de l'Intelligence Artificielle par la Détection des Données Hors Distribution et des Erreurs de Classification

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n°129 Sciences and Technologies of Information and Communication (STIC)
Graduate School of Computer Science

Thèse préparée à CentraleSupélec, Université Paris-Saclay

Eduardo Dadalto Câmara Gomes

Composition du Jury

Yves GRANDVALET Director of Research, Heudiasyc, UTC, CNRS	Rapporteur
Yann CHEVALEYRE Professor, LAMSADE, CNRS, Université Paris-Dauphine-PSL	Rapporteur
Nicolas VAYATIS Professor, Centre Borelli, CNRS, ENS Paris-Saclay	Président
Florence d'ALCHÉ-BUC Professor, LTCI, Télécom Paris, IP Paris	Examinatrice
Nicolas THOME Professor, MLIA, ISIR, Sorbonne Université	Examinateur
Pascal POUPART Professor, University of Waterloo, Vector Institute	Examinateur

Direction de la thèse

Pablo PIANTANIDA Professor, Director of the International Laboratory on Learning Systems (ILLS), Associate Academic Member at Mila	Directeur de thèse
Florence ALBERGE Professor, SATIE Laboratory, Université Paris-Saclay, ENS Paris-Saclay, CNRS	Co-encadrante de thèse

Thèse de doctorat

For my beloved mother and father
who taught me the values of hard work and integrity.

In memory of Prof. Calyampudi Radhakrishna Rao.

Acknowledgements

I would like to express my deepest gratitude to the following individuals whose contributions have been indispensable in shaping the research presented in this thesis.

Prof. Pablo Piantanida, the Director of the International Laboratory on Learning Systems (ILLS) and Associate Academic Member at Mila, for your mentorship, guidance, and support. I have learned a lot with you and your perspectives have broadened my horizons.

Prof. Florence Alberge, from the SATIE Laboratory at Université Paris-Saclay, for your collaboration and constructive feedback.

Prof. Pierre Duhamel, from the Laboratoire des Signaux et Systèmes at Université Paris-Saclay, for your collaboration and the numerous thought-provoking discussions we had.

Assoc. Prof. Pierre Colombo, from the MICS Laboratory, for your collaboration, sharing your knowledge in NLP, and the valuable discussions we had.

Dr. Marco Romanelli, Post Doctorate fellow at New York University, for the close collaboration, friendship, support and for sharing your rigorous scientific methodology.

Dr. Georg Pichler, Post Doctorate fellow at TU Wien, for the collaboration and sharing your vast mathematical knowledge.

I also had the pleasure of working with Dr. Maxence Ernoult (Rain), Dr. Federica Granese (IRD), Dr. Nathan Noiry (Owkin), Dr. Guillaume Staerman (INRIA Paris-Saclay), Maxime Darrin (ILLS, Mila), and others.

I would like to extend my appreciation to the funding bodies that made this research possible. BPI-France and IBM France that, through the ANR project entitled AI for Digital automatisation (AIDA), provided generous financial support, covered the costs of equipment and travel expenses. IDRIS - CNRS for the generous grant, allowing me to cover the heft computational costs. Finally, CentraleSupélec and L2S laboratory, for the essential resources and administrative support, specially from Stéphanie Douesnard, Caroline Lion, José Fonseca, and Anne-Elisabeth Kohl.

I am extremely grateful to my friends, who brought joy and laughter even to the most challenging moments. My family, who have been a constant source of emotional support. And especially my girlfriend, for her unwavering belief in me and patience at every step during this process. Your presence in my life has been an anchor, grounding me in moments of doubt and celebrating with me in moments of triumph.

I finally wish to extend my acknowledgments to the authors, contributors, and maintainers of open science platforms like ArXiv and HAL and open-source tools like Python, LaTeX, Linux, PyTorch, Transformers, timm, among others for for their transformative impact on research.

Contents

List of Figures

List of Tables

List of Symbols

Most important symbols in order of appearance:

X	Input features random variable (r.v)
Y	Concepts r.v
$\mathcal{X}$	Input features domain set
$\mathcal{Y}$	Concepts domain set
$p(X, Y)$	Joint probability measure $p_{XY}(X = \boldsymbol{x}, Y = y)$
$P(Y \mid X)$	Conditional probability mass function (pmf) of the labels given the input features or concept distribution $P_{Y\|X}(Y = y \mid X = \boldsymbol{x})$
$p(X)$	Probability density function (pdf) of the features or covariate distribution $p_X(X = \boldsymbol{x})$
$p(X \mid Y)$	Conditional pdf of the features given the labels $p_{X\|Y}(X = \boldsymbol{x} \mid Y = y)$
$P(Y)$	Pmf of the labels or prior distribution $P_Y(Y = y)$
Z	Hidden binary r.v indicating the event if an example is an outlier (out-of-distribution) or not
q_X	Unknown outliers pdf or out-of-distribution $q_X(X = \boldsymbol{x})$
$\mathcal{D}_n$	Training set with $n \in \mathbb{Z}^+$ samples
$f_{\mathcal{D}_n}$	Learned predictive rule from the features set to the concept set. Used interchangeably with f
T_l	Transformation mapping from the input features to the l-th layer outputs of a DNN
U	Latent features r.v
$\mathcal{U}$	Latent features r.v domain set
$\mathcal{A}$	Arbitrary decision set
H_0	Null hypothesis of a statistical hypothesis test
H_A	Alternative hypothesis of a statistical hypothesis test
s	Similarity score function
γ	Detection threshold to obtain a hard decision from the similarity score
d	Binary detection function
α	Precision level on inlier data
E	Hidden binary r.v indicating if an example is misclassified or not
Pe	Probability of classification error

CHAPTER 1

Introduction

Machine learning (ML) enables computers to perform intelligent tasks (Turing, 1950). The term artificial intelligence (AI), so popular today, was coined by John McCarthy in 1955 to refer to the science and engineering of making intelligent machines. According to Valiant (1984), intelligent capabilities would arise by learning from data without explicit instructions. In this sense, the goal of the artificial learner is to generalize concepts from past experiences to future interactions in other contexts (Vapnik, 1995). To achieve this, artificial neural networks (NN) (Rosenblatt, 1958; Ivakhnenko et al., 1965) have shown to be a suitable statistical model, rich enough to be able to approximate any function (Cybenko, 1989; Hornik et al., 1989) in a region, and flexible enough to incorporate inductive biases related to the downstream applications, e.g., convolutional neural networks (CNN) (LeCun et al., 1989). By incorporating non-linearities (Fukushima, 1969) and multiple layers, deep architectures learn through backpropagation (BP) (Linnainmaa, 1976) and stochastic gradient descent (SGD) (Robbins and Monro, 1951) and is the subject of study of the field entitled as Deep Learning (DL) (Dechter, 1986). Recent improvements of specialized hardware, such as graphic processing units (GPU) (Clark, 1982) and NN architectures (Vaswani et al., 2017) combined with efforts and investment from academia, industry, and open source community, resulted in an astounding advancement and proliferation of intelligent systems in the last decade. As a result, ML disrupted various fields, including–but not limited to–computer vision, natural language processing, multi-autonomous agent coordination, and multimodal reasoning. This technological progress can potentially revolutionize science, medicine, transportation, cinema, and numerous other industries, potentially reshaping workforce requirements. However, AI will likely complement rather than replace traditional approaches in the short term.

Deploying AI systems in real-world applications is not without its challenges. Alongside these remarkable achievements, we are confronted with a myriad of limitations related to privacy (Liu et al., 2021), fairness (Mehrabi et al., 2021), adversarial robustness(Chakraborty et al., 2018), explainability (Burkart and Huber, 2021), uncertainty quantification (Abdar et al., 2021) and trustworthiness (Kaur et al., 2022) in general, provoking an inevitable loss of trust in automated systems. Hence, one of the most pressing challenges in AI research is ensuring the safety and reliability of these systems, particularly when they are applied in safety-critical domains, such as autonomous driving and healthcare. Failures or accidents occur when human designers have specific objectives, but the deployed

AI systems produce unexpected outcomes. Such errors can have dire consequences, from incorrect medical diagnoses to road accidents. The source of these problems is diverse, including mismatches between training and real-world data distributions, lack of objective function specification, and more. Humans suffer from the same problem when faced with situations our previous experiences did not prepare us to deal with. For instance, when visiting a country whose culture is very different from ours, we may encounter difficulty and inevitably commit mistakes, causing cultural shock. Contrary to machines, humans were able to develop a critical skill for such a situation: admitting our ignorance rather than assuming that our pre-defined concepts translate flawlessly to every situation.

Even though an ML model may achieve great generalization for a given data distribution, the training environment does not necessarily reflect those encountered in the open world (Quionero-Candela et al., 2009). As the learning framework often assumes that the data distribution is static, i.e., does not change between the learning phase and the time we apply the model, we naturally need a mechanism to alert us. Real-life environments are usually non-stationary, and the complexities of matching the development scenario to the production one are either too high or too expensive. The failure of ML models to adapt to non-stationary environments could limit their adoption. So, models that warn for unusual examples could prevent unintended behavior in AI systems. The work presented in this manuscript contributes towards mitigating these risks.

1.1 Uncertainty Estimation at the Core of Trustworthiness in AI Applications

Uncertainty estimation is the bedrock of trustworthiness in AI applications by providing a quantifiable measure of the model's confidence in its predictions. Uncertainty-aware models facilitate risk-sensitive applications, guiding users on when to trust predictions or seek additional information. In this context, uncertainty estimation is not merely a technical detail. It emerges as a fundamental element for the responsible deployment of AI, ensuring that users can rely on AI systems as trustworthy and accountable. Uncertainty is usually split into two facets: *aleatoric* (irreducible) and *epistemic* (reducible). Estimating each of them or the aggregated uncertainty is a rich and vast study field in the literature on trustworthy AI. As a consequence, it employs multiple techniques and approaches that, unfortunately, we cannot treat all of them in depth in this thesis. This manuscript will briefly introduce some of these techniques in the following and will then focus on state-of-the-art techniques for out-of-distribution and misclassification detection.

1.1.1 Epistemic versus Aleatoric Uncertainty

Aleatoric uncertainty, also known as statistical or inherent uncertainty, arises from the inherent stochasticity or randomness in the data-generating process itself. It is related to the irreducible variability even if we had a perfect model and infinite data. This type of uncertainty can be further classified into homoscedastic uncertainty, which remains constant for different samples, and heteroscedastic uncertainty, which can vary between samples.

Epistemic uncertainty, also known as model uncertainty or knowledge uncertainty, stems from our lack of knowledge or incomplete understanding of the underlying system. Epistemic uncertainty could be reduced with more data or a more sophisticated model. In regression tasks, epistemic uncertainty represents uncertainty about the model parameters. For instance, if the model hasn't seen certain patterns in the data during training, it might be uncertain how to predict those regions. Reducing epistemic uncertainty involves improving the model architecture, collecting more relevant data, or refining the training process.

Some works focus on disentangling them. (Kotelevskii et al., 2022; Mukhoti et al., 2023), however, the work on this manuscript focuses on the combined or total uncertainty that are key for detecting inlier mistakes and novelty.

1.1.2 Conformal Prediction

An interesting take on the problem of uncertainty in AI is *conformal prediction* (Romano et al., 2020; Gibbs and Candes, 2021; Angelopoulos and Bates, 2021; Angelopoulos et al., 2021). In addition to estimating the most likely outcome, a conformal predictor provides a *prediction set* or *interval* that provably contains the ground truth with high probability, unlike traditional models that provide point predictions. This can be adapted post-hoc or by leveraging a conformal learning algorithms to create models that produce valid and informative sets. Intuitively, a larger set indicates low confidence on the prediction while a smaller set shows less uncertainty in the prediction. For a more formal introduction, please refer to Vovk et al. (2005).

1.1.3 Bayesian Learning

Bayesian Learning for uncertainty estimation is a framework grounded in Bayesian statistics that extends beyond conventional point-estimate models by placing a distribution over model parameters. It involves a probabilistic approach to reasoning about uncertainties associated with predictions grounded on Bayesian statistics. The fundamental components of Bayesian Learning include a *prior* distribution, which encapsulates prior knowledge or assumptions, and a *likelihood* function that quantifies the probability of observed data given the model parameters. Following data observation, Bayesian inference leads to the *posterior* distribution, reflecting updated beliefs about the model parameters. In the realm of deep learning, Bayesian Neural Networks (BNNs) represent a relevant application where probability distributions over weights replace fixed values, allowing for the explicit modeling of uncertainty in predictions. Some key references in Bayesian Deep Learning are Gal and Ghahramani (2016); Lakshminarayanan et al. (2017); Kendall and Gal (2017); Snoek et al. (2019); Mukhoti et al. (2021); Einbinder et al. (2022); Thiagarajan et al. (2022). For a formal and comprehensive introduction to the topic, please refer to MacKay (1992).

1.2 Background

The primary objective of this thesis is to enhance the detection capability of ML models to identify situations that deviate from the norm, increasing their reliability. We hope to achieve this by developing detection methods to identify samples from outside the training distribution, more popularly known in the literature as out-of-distribution (OOD) detection. Also, we strive to quantify uncertainty in inlier predictions for performing misclassification detection. These are crucial aspects of AI safety that identify inputs that, may threaten the system response, and may cause a drop in accuracy. Since classical learning algorithms depend strongly on the properties of their inputs, e.g., the i.i.d. assumption, they tend to fail silently when faced with shifted data.

Formally, Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a continuous feature space, and let $\mathcal{Y} = \{1, \dots, C\}$ denote the label space related to some task of interest. Also, let $X \in \mathcal{X} \sim p_X$ and $Y \in \mathcal{Y} \sim P_Y$ be the random variables (r.v) governing the realization of the features $X = \boldsymbol{x}$ and discrete concepts $Y = y$ involved in the problem. We denote by p_{XY} and q_{XY} the underlying source and target probability density functions (pdf) associated with the distributions P and Q on $\mathcal{X} \times \mathcal{Y}$, respectively. We assume that a machine learning model $f : \mathcal{X} \to \mathcal{Y}$ is trained on some training set $\mathcal{D}_n = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\} \sim p_{XY}$, which yields a model that, given an input $\boldsymbol{x} \in \mathcal{X}$, outputs a prediction on $\mathcal{Y}$, i.e., $f_{\mathcal{D}_n}(\boldsymbol{x}) = \arg\max_{y \in \mathcal{Y}} p_{\widehat{Y}|X}(y|\boldsymbol{x})$. Data shift occurs when the test data joint probability distribution differs from the distribution a model expects, i.e., $p_{XY}(\boldsymbol{x}, y) \neq q_{XY}(\boldsymbol{x}, y)$. Due to this mismatch, the model's response may suffer a drop in accuracy. We can observe the types of *distribution shift* by decomposing the joint probability density function (pdf) into

$$p(X,Y) = \underbrace{P(Y|X)}_{\text{concept}} \underbrace{p(X)}_{\text{covariate}} = p(X|Y) \underbrace{P(Y)}_{\text{prior}}. \tag{1.1}$$

Each decomposed shift happens under the condition that the accompanying decomposed probability remains unchanged. Briefly, *concept drift* is usually attributed to the presence of novel classes or concepts with covariates following the same known distribution. *Covariate shift* often happens because the input data comes from different domains, e.g., drawing of concepts, while the training features are natural pictures. Finally, a *prior shift* or label shift usually occurs when the test condition is biased towards some classes. All of these shifts may have negative impacts on the model. Shifts that do not affect the detector's performance are referred to *virtual* shifts. The primary emphasis in OOD detection will be on concept shift detection. In Chapter 4, we will also treat the problem of identifying covariate shifts. Prior shift detection is left for future work.

1.3 Out-of-Distribution Detection

Out-of-distribution detection boils down to a binary classification problem by one-sided error estimation but with a caveat: data from just one class is available in training time. According to this hypothesis, it is impossible to collect enough data to learn the outlier distribution satisfactorily. To model this detection problem, we introduce an artificial hidden binary r.v $Z \in \{0, 1\}$ indicating with $z = 1$ that the input sample $\boldsymbol{x}$ is an outlier and $z = 0$ it is an inlier. The open-world data can be modeled by a *mixture* distribution defined by

$$p_{X|Z}(\boldsymbol{x}|z=0) \triangleq p_X(\boldsymbol{x}), \text{ and } p_{X|Z}(\boldsymbol{x}|z=1) \triangleq q_X(\boldsymbol{x}). \tag{1.2}$$

To further ground it formally, we have to make the hypothesis that q_X and p_X are *sufficiently different* from each other (this will be clarified later on) so that we can separate them.

One of the main difficulties behind the problem is that little can be assumed about the unknown distribution q_X from which no samples are available during training. Furthermore, there is no reason to suppose that the observed change in p_X induces a change in $P_{Y|X}$ and thus, we will assume that $P_{Y|X} \equiv Q_{Y|X}$. However, this equality should not be mistaken as being equivalent to say that the learned rule $f_{\mathcal{D}_n}$ predicting Y from X should not be re-adapted to the new input distribution q_X. This is because learning a predictor from finite data may be tuned to functions that fit well the empirical data distribution according to regions where p_X has a high probability which may be somewhat different to those in q_X. Indeed, such an assumption will be critical to detecting the underlying drift based not only on the testing sample but, in particular, on the behavior of the trained predictor when evaluating it on such samples. Features and pattern that are rarely observed during training could be identified as being caused by OOD data. The next subsection will dive into the optimal discriminator when using features as a dense proxy for the input data.

1.3.1 Optimal OOD Detection and Performance of the Oracle

This section details the mathematical model and detection performances of an Oracle discriminator accessing the underlying distributions. Basic properties and proof are relegated to Appendix A.1. Since different methods rely on different parts of a NN to perform detection, let $T_l : \mathcal{X} \to \mathcal{U}_l$ be a Borel-measurable mapping denoting the transformation between input features in $\mathcal{X}$ and the l-th layer in $\mathcal{U}_l \subseteq \mathbb{R}^{d_l}$ with $l = \{1, \ldots, L\}$. The last layer $\mathcal{U}_L = [0,1]^K$ and $T_L(\boldsymbol{x}) = [P_{\widehat{Y}|X}(1|\boldsymbol{x}), \ldots, P_{\widehat{Y}|X}(K|\boldsymbol{x})]$ indicates the soft-probabilities. Then T_l induces a probability measure $P^{T_l}_{U_l|Z}$ on the Borel σ-field $\mathcal{B}^{d_l}$ as follows:

$$P^{T_l}_{U_l|Z}(\mathcal{A}|z) = \int_{T_l^{-1}(\mathcal{A})} P_{X|Z}(\boldsymbol{dx}|z), \ \ \mathcal{A} \in \mathcal{B}^{d_l}. \tag{1.3}$$

Furthermore, we will assume the existence of pdf $p_{U_1|Z}, \ldots, p_{U_L|Z}$ corresponding to the change of measure induced by transformations at each layer. For the special case of the last layer, the probability distribution is given by

$$P_{\widehat{Y}|Z}(y|z) = \int_{\mathcal{X}} P_{\widehat{Y}|X}(y|\boldsymbol{x}) P_{X|Z}(\boldsymbol{dx}|z). \tag{1.4}$$

For simplicity, we will use a generic $p_{U|Z}$ which should be interpreted as the available information to perform the decision, e.g., if it is the last layer outputs, $p_{U|Z}$ should be understood as $P_{\widehat{Y}|Z}$ with $\boldsymbol{u} \equiv y \in \mathcal{Y}$ or the specific pdf at the given l-th layer with $\boldsymbol{u} \in \mathbb{R}^{d_l}$.

We begin by stating the optimal rejection region of an Oracle, which has access to all involved distributions.

Definition 1.3.1 (Most efficient test)**.** Let $\mathcal{A}(\gamma, \mathcal{D}_n) \subseteq \mathcal{U}$ be the set containing all the $\boldsymbol{u} \in \mathcal{U}$ to be detected as being out-of-distribution samples and thus, $\mathcal{A}^c(\gamma, \mathcal{D}_n)$ contain all samples must be declared to be in-distribution. The following decision region achieves the most efficient test:

$$\mathcal{A}(\gamma, \mathcal{D}_n) \triangleq \left\{ \boldsymbol{u} \in \mathcal{U} : \frac{p_{U|Z}\big(\boldsymbol{u}|z=1; \mathcal{D}_n\big)}{p_{U|Z}\big(\boldsymbol{u}|z=0; \mathcal{D}_n\big)} > \gamma \right\} \tag{1.5}$$

$$= \left\{ \boldsymbol{u} \in \mathcal{U} : \frac{P_Z(z=0)}{P_Z(z=1)} \cdot \frac{P_{Z|U}\big(1|\boldsymbol{u}; \mathcal{D}_n\big)}{1 - P_{Z|U}\big(1|\boldsymbol{u}; \mathcal{D}_n\big)} > \gamma \right\}, \tag{1.6}$$

where $0 < \gamma < \infty$ and $\mathcal{D}_n$ denotes the implicit dependence of the involved transformations across layers with the training set.

Proposition 1 (Detection tradeoffs)**.** *Let $\mathcal{A} \subseteq \mathcal{U}$ be any decision set, and let*

$$\epsilon_0(\mathcal{A}, \mathcal{D}_n) = \int_{\mathcal{A}} P_{U|Z}(d\boldsymbol{u}|z=0; \mathcal{D}_n), \tag{1.7}$$

$$\epsilon_1(\mathcal{A}^c, \mathcal{D}_n) = \int_{\mathcal{A}^c} P_{U|Z}(d\boldsymbol{u}|z=1; \mathcal{D}_n) \tag{1.8}$$

be the average Type-I and Type-II error probabilities, respectively. Then,

$$\epsilon_0(\mathcal{A}, \mathcal{D}_n) + \epsilon_1(\mathcal{A}^c, \mathcal{D}_n) \geq \int_{\mathcal{U}} \min \big\{ P_{U|Z}(d\boldsymbol{u}|z=1; \mathcal{D}_n), P_{U|Z}(d\boldsymbol{u}|z=0; \mathcal{D}_n) \big\} \tag{1.9}$$

$$= 1 - \big\| P_{U|Z}(\cdot|z=1; \mathcal{D}_n) - P_{U|Z}(\cdot|z=0; \mathcal{D}_n) \big\|_{TV}. \tag{1.10}$$

Equality is achieved by setting $\mathcal{A}^\star \equiv \mathcal{A}(1; \mathcal{D}_n)$. Moreover, if the hypothesis is equally distributed, then the minimum average Bayesian error satisfies

$$\inf_{\psi} \boldsymbol{P}_{\mathcal{D}_n} \{\psi(\boldsymbol{U}) \neq Z\} = \frac{1}{2} \Big[1 - \big\| P_{U|Z}(\cdot|1; \mathcal{D}_n) - P_{U|Z}(\cdot|0; \mathcal{D}_n) \big\|_{TV} \Big]. \tag{1.11}$$

The proof is relegated to Appendix A.1.

1.3.2 Statistical Hypothesis Testing Framework

Statistical hypothesis testing can be designed to test if a given sample is distributed accordingly to p_X or q_X. Traditionally, these tests are conducted in the asymptotic regime, where the number of samples tends to infinity. Efforts towards more efficient tests are recently being developed in the literature. In special, Valiant and Valiant (2017) proposes an efficient test and a lower bound of number of samples to detect OOD distributions. They leverage symmetric properties of probability distributions to design this test. Thus, out-of-distribution detection can be seen as a hypothesis test. Formally, the null and alternative hypothesis writes:

$$H_0 : X \sim p_X \text{ and } H_A : X \sim q_X. \tag{1.12}$$

However, since the true distributions are unknown in practice, it is common to rely on a detection score function to make the test. Let $s : (\boldsymbol{x}, f_{\mathcal{D}_n}) \to \mathbb{R}^+$ be a similarity score function that measures how adapted the input is to the model, i.e., a low score indicates the sample is untrustworthy, and a high value indicates otherwise. Hence, we frame the statistical hypothesis test as a *left-tailed test* (Lehmann and Romano, 2005) that will detect an input sample $\boldsymbol{x}$ according to the magnitude of $s(\boldsymbol{x})$ and a threshold $\gamma \in \mathbb{R}$.

Definition 1.3.2 (OOD detector)**.** Given s, $f_{\mathcal{D}_n}$, and γ, an out-of-distribution detector is defined by

$$d(\boldsymbol{x}) \triangleq \mathbb{1}\left[s(\boldsymbol{x}, f_{\mathcal{D}_n}) \leq \gamma\right] = \begin{cases} 1 & \text{if } s(\boldsymbol{x}) \leq \gamma \\ 0 & \text{if } s(\boldsymbol{x}) > \gamma \end{cases} \tag{1.13}$$

Hence, the role of the classifier-detector system $(f_{\mathcal{D}_n}, d)$ is to keep a prediction if the input sample $\boldsymbol{x}$ is not *rejected* by the detector d, i.e., if $\widehat{z} = d(\boldsymbol{x}) = 0$. A remaining key step is finding the detection threshold γ. In practice, it is calibrated with the help of a validation set and a desired level of precision $\alpha \in [0, 1]$ on inlier data. Let $F(s(X))$ be the cumulative density function of the r.v. $s(X)$,

$$\gamma = \inf\{s(X) \in \mathbb{R} : 1 - \alpha \leq F(s(X))\}. \tag{1.14}$$

1.4 Misclassification Detection

This subsection recalls misclassification events and links them with OOD detection. Let us consider a discrete r.v. expressed by $E \triangleq \mathbb{1}[f_{\mathcal{D}_n}(X) \neq Y]$, i.e., the *misclassification* event is denoted by $\{E = 1\} \equiv \{(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y} : \ f_{\mathcal{D}_n}(\boldsymbol{x}) \neq y\}$. We can express the joint probability density function p_{XY} as a mixture:

$$p_{XY}(\boldsymbol{x}, y) = p_{XY|E}(\boldsymbol{x}, y|E = 1)P_E(1) + p_{XY|E}(\boldsymbol{x}, y|E = 0)P_E(0). \tag{1.15}$$

By taking the marginal of Eq. (1.15) over Y, we obtain:

$$p_X(\boldsymbol{x}) = p_{X|E}(\boldsymbol{x}|1)P_E(1) + p_{X|E}(\boldsymbol{x}|0)P_E(0), \tag{1.16}$$

where $p_{X|E}(\boldsymbol{x}|1)$ denotes the pdf truncated to the error event and $p_{X|E}(\boldsymbol{x}|0)$ the pdf truncated to the event of correct classification.

We finally define the *probability of classification error* as:

$$\mathrm{Pe}(\boldsymbol{x}) \triangleq P_{E|X}(1|\boldsymbol{x}) = 1 - P_{Y|X}\left(f_{\mathcal{D}_n}(\boldsymbol{x})|\boldsymbol{x}\right). \tag{1.17}$$

Similar to the test defined in Eq. (1.12), we define the following hypothesis test for the misclassification detection problem:

$$H_0 : X \sim p_{X|E=0} \text{ and } H_A : X \sim p_{X|E=1}. \tag{1.18}$$

1.5 Literature Review

This section will briefly describe and extensively cite key references on OOD and misclassification detection, finally shedding light to a new object of study that combines both domains into a single detection framework.

Out-of-distribution detection. With roots in one-class novelty detection (Pimentel et al., 2014), out-of-distribution is also referred to in the literature as open-set recognition (Geng et al., 2021) and semantic anomaly detection (Pang et al., 2021). OOD detection became popular among the DL community when Hendrycks and Gimpel (2017) introduced a baseline method to detect OOD examples on different classification tasks, such as image recognition, text categorization, and speech recognition. The maximum softmax probability is used to score test samples, and inputs with low confidence are detected as OOD. Nevertheless, Nguyen et al. (2014) demonstrates that neural networks can produce arbitrarily high softmax scores for inputs far from the training data.

Overall, detection methods are taxonomized into *confidence-based* (Hein et al., 2019; Hendrycks and Gimpel, 2017; Liang et al., 2018b; Hsu et al., 2020; Liu et al., 2020; Hendrycks et al., 2022; Sun and Li, 2022), which rely on the logits and softmax outputs of the network to identify patterns that distinguish in-distribution from OOD samples. *Feature-based* (Sastry and Oore, 2020; Sun et al., 2021; Huang et al., 2021; Zhu et al., 2022b; Colombo et al., 2022; Dong et al., 2021; Song et al., 2022; Lin et al., 2021; Djurisic et al., 2023a; Lee et al., 2018b; Fort et al., 2021; Darrin et al., 2024; Sun et al., 2022; Du et al., 2022a; Ming et al., 2023; Djurisic et al., 2023b). Exploring latent representations typically involves designing methods that measure the dissimilarity between the input sample and the training dataset or prototypes. *generative-based* methods (Schlegl et al., 2017; Vernekar et al., 2019; Xiao et al., 2020; Ren et al., 2019; Kirichenko et al., 2020; Nalisnick et al., 2019; Choi and Chung, 2020) fit a generative model to the in-distribution and test the likelihood of testing samples as an OOD criterion. *Mixed feature-logits* (Dadalto et al., 2022; Wang et al., 2022; Dadalto et al., 2023b) leverage information from the outputs and the latent representations, managing to combine the advantages of both approaches. *Uncertainty-regression* (Lakshminarayanan et al., 2017; DeVries and Taylor, 2018; Lee et al., 2018a), aims to learn an uncertainty score in training time. Finally, *learning with outlier exposure* (Hendrycks et al., 2019; Du et al., 2022b) utilizes outlier samples to regularize training and shape outlier-aware decision boundaries. Recent benchmarks (Zhang et al., 2023) reveal no clear single winner, which is a direct consequence of the constraints and challenges imposed on the problem.

Misclassfication detection. The goal of misclassification detection is to create techniques that can evaluate the reliability of decisions made by classifiers and determine whether they can be trusted. Liang et al. (2018a) proposes applying temperature scaling (Guo et al., 2017) and perturbing the input samples to the decision boundary's direction to better detect misclassifications. A line of research trains auxiliary parameters to estimate a detection score (Corbière et al., 2019) directly, following the idea of *learning to reject* (Chow, 1970; Geifman and El-Yaniv, 2017). Exposing the model to outliers or severe augmentations during training has been explored in previous work (Zhu et al., 2023) to evaluate if these heuristics are beneficial for this particular task apart from improving robustness to outliers. Granese et al. (2021) proposes a mathematical framework and a simple detection method based on the estimated probability of error. We show that their proposed detection metric is a special case of ours. Zhu et al. (2022a) study the phenomenon that *calibration* methods are often useless or harmful for failure prediction and provide insights into why. Cen et al. (2023) discusses how training settings such as pre-training or outlier exposure impact misclassification and open-set recognition performance. Related sub-fields are *predictive uncertainty estimation* via Bayesian Neural Networks estimation (Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017; Mukhoti et al., 2021; Einbinder et al., 2022; Snoek et al., 2019; Thiagarajan et al., 2022) and *conformal predictions* (Gibbs and Candes, 2021).

A new line of research proposes combining misclassification detection or rejection option and out-of-distribution detection (Narasimhan et al., 2023; Katz-Samuels et al., 2022; Xia and Bouganis, 2022), also known as **selective classification with out-of-distribution detection (SCOD)**. In this case, at test time, the objective is to reject misclassified ($E = 1$) and OOD ($Z = 1$) samples at the same time, with a single rejection criterion.

1.6 Popular Detection Methods

Below, several essential detection methods using a standard notation that will be an accessory in understanding the techniques developed in the literature are introduced

Maximum Softmax Probability (MSP). The Maximum Softmax Probability method, as outlined in (Hendrycks and Gimpel, 2017), serves as a common baseline for out-of-distribution (OOD) detection. Provided an input $\boldsymbol{x}$ and a pre-trained neural network $f_{\mathcal{D}_n}(\cdot)$, the classifier's most confident class probability serves as similarity score.

$$s(\boldsymbol{x}; f_{\mathcal{D}_n}) = \max_{y \in \mathcal{Y}} \frac{e^{f_y(\boldsymbol{x})}}{\sum_{j=1}^{K} e^{f_j(\boldsymbol{x})}} \tag{1.19}$$

A limitation of this method is that the actual conditional probability function $P_{Y|X}$ is unknown, leading to reliance on $P_{\widehat{Y}|X}(y|\boldsymbol{x}; \mathcal{D}_n)$, an estimate derived from the training data $\mathcal{D}_n$.

ODIN. The work by Liang et al. (2018b) enhances the MSP baseline by adjusting the softmax outputs using a temperature scaling parameter

$$s(\boldsymbol{x}; T, f_{\mathcal{D}_n}) = \max_{y \in \mathcal{Y}} \frac{e^{f_y(\boldsymbol{x})/T}}{\sum_{j=1}^{K} e^{f_j(\boldsymbol{x})/T}}, \tag{1.20}$$

where $T \in \mathbb{R}^+$ is the temperature. Additionally, they introduce a small adversarial noise perturbation to the inputs based on their observation that perturbed out-of-distribution data exhibits lower confidence than perturbed in-distribution samples, indicating that OOD data are present on flatter regions of the loss landscape. The perturbation writes:

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \eta \cdot \operatorname{sign}\left\{-\nabla_{\boldsymbol{x}} \log s(\boldsymbol{x}; T, f_{\mathcal{D}_n})\right\}, \tag{1.21}$$

where η is the perturbation magnitude. The hyperparameters T and η can be optimized using random noise input, such as a Gaussian or uniform distribution. Importantly, this optimization process does not rely on prior knowledge of the out-of-distribution dataset, as described in (Hsu et al., 2020) and is a common practice in the literature.

Energy based OOD detector. In Liu et al. (2020), they propose an energy-based detector that relies on discrepancies in free energies between in-distribution and out-of-distribution examples for distinguishing between them. Instead of utilizing soft probability outputs, the energy-based model employs the Helmholtz free energy equation, following the concept from Lecun et al. (2006). Formally, the free energy of an example $\boldsymbol{x}$ is defined as $E(\boldsymbol{x}; f) = -T \cdot \log \sum_{j=1}^{K} e^{f_j(\boldsymbol{x})/T}$. A data point with low energy has a higher likelihood of being in-distribution and vice versa. So, the log-likelihood writes

$$\log p(\boldsymbol{x}) = -E(\boldsymbol{x}; f)/T - \underbrace{\log \int_{\mathcal{X}} e^{-E(\boldsymbol{x}; f)/T} \boldsymbol{dx}}_{\text{constant w.r.t. } \boldsymbol{x}} \tag{1.22}$$

showing that $-E(\boldsymbol{x}; f)$ is linearly aligned with the log-likelihood function, which is desirable for OOD detection. The work also proposes an energy-based cost function for energy-bounded learning.

Mahalanobis distance-based score. The Mahalanobis method (Lee et al., 2018b) models the embedding of a DNN as a Gaussian mixture model with a tied covariance matrix and parameters estimated on the training dataset. The modes of the mixture are estimated with samples from a single class. They use the outputs of every DNN latent block to leverage useful information for discrimination. For a test sample $\boldsymbol{x}$, the confidence score from the ℓ-th layer is calculated based on the Mahalanobis distance Mahalanobis (1936) between $f^{(\ell)}(\boldsymbol{x})$ and the closest class-conditional distribution:

$$s_\ell(\boldsymbol{x}) = \max_y - \left(f^{(\ell)}(\boldsymbol{x}) - \widehat{\boldsymbol{\mu}}_y^{(\ell)} \right)^\top \widehat{\Sigma}_\ell^{-1} \left(f^{(\ell)}(\boldsymbol{x}) - \widehat{\boldsymbol{\mu}}_y^{(\ell)} \right), \tag{1.23}$$

where $f^{(\ell)}(\cdot)$ is the ℓ-th latent feature extractor, and $\widehat{\boldsymbol{\mu}}_y^{(\ell)}$ and $\widehat{\Sigma}_\ell$ are, the empirical class conditional mean and global covariance matrix estimates, respectively. The covariance matrix is often not full rank, so the pseudo-inverse is calculated instead of the inverse. In addition, input pre-processing and feature ensemble are also used to boost performance. A logistic regression model fits the multiplicative weights α_ℓ for each layer score. Finally, the Mahalanobis-based discriminator is given by thresholding the expression $\sum_\ell \alpha_\ell s_\ell(\boldsymbol{x})$. The negative sign is to transform a distance-based score (i.e., larger values indicate a higher likelihood of being OOD) to a confidence-based score (i.e., smaller values indicate a higher likelihood of being OOD).

Relative Mahalanobis distance. Fort et al. (2021) introduces an adaptation of the Mahalanobis distance method focusing on improving near-OOD detection. It involves fitting a global Gaussian distribution to the training set without considering class information. This process computes the global mean ($\widehat{\mu}_{\text{global}}$) and covariance ($\widehat{\boldsymbol{\Sigma}}_{\text{global}}$) of the data. The similarity score is the difference between the original Mahalanobis distance (Lee et al., 2018b) and the Mahalanobis distance concerning the global Gaussian distribution:

$$\begin{aligned} s_\ell(\boldsymbol{x}) = \max_y - \left(f^{(\ell)}(\boldsymbol{x}) - \widehat{\mu}_y \right) \widehat{\boldsymbol{\Sigma}}^{-1} \left(f^{(\ell)}(\boldsymbol{x}) - \widehat{\mu}_y \right)^\top - \\ \left(f^{(\ell)}(\boldsymbol{x}) - \widehat{\mu}_{\text{global}} \right) \widehat{\boldsymbol{\Sigma}}_{\text{global}}^{-1} \left(f^{(\ell)}(\boldsymbol{x}) - \widehat{\mu}_{\text{global}} \right)^\top . \end{aligned} \tag{1.24}$$

which is equivalent to the log-likelihood ratio between the class mode of a mixture of Gaussians with a common covariance matrix and a global Gaussian distribution estimated using the entire training dataset.

Data depth. Data depths extend the notion of a median to the multivariate setting (Tukey, 1975b). Multivariate data depths are nonparametric statistics that measure the centrality of any element of $\mathbb{R}^d$, where $d \geq 2$, w.r.t. a probability distribution (respectively a random variable) defined on any subset of $\mathbb{R}^d$. Formally, a data depth is defined as follows:

$$\begin{array}{rccl} D: & \mathbb{R}^d \times \mathcal{P}(\mathbb{R}^d) & \longrightarrow & [0,1], \\ & (\boldsymbol{x}, P_X) & \longmapsto & D(\boldsymbol{x}, P_X). \end{array} \tag{1.25}$$

The higher $D(\boldsymbol{x}, P_X)$, the deeper $\boldsymbol{x}$ is in P_X. Colombo et al. (2022) propose to leverage the Integrated Rank-Weighted (IRW) depth (Staerman et al., 2021). The IRW depth of $\boldsymbol{x} \in \mathbb{R}^d$ w.r.t. to a probability distribution P_X on $\mathbb{R}^d$ is given by:

$$D_{\mathrm{IRW}}(\boldsymbol{x}, P_X) = \int_{\mathbb{S}^{d-1}} \min\left\{F_u\left(\langle \mathbf{u}, \boldsymbol{x}\rangle\right), 1 - F_u\left(\langle \mathbf{u}, \boldsymbol{x}\rangle\right)\right\} \mathrm{d}\mathbf{u},$$

where $F_u(t) = \Pr(\langle \mathbf{u}, X\rangle \leq t)$ and $\mathbb{S}^{d-1}$ is the unit hypersphere. In practice, the expectation is approximated using Monte-Carlo. Given a sample $\mathcal{S}_n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, the approximation of the IRW depth is defined as:

$$\widetilde{D}_{\mathrm{IRW}}(\boldsymbol{x}, \mathcal{S}_n) = \frac{1}{n_{\mathrm{proj}}} \sum_{k=1}^{n_{\mathrm{proj}}} \min\left\{\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left\{\langle \mathbf{u}_k, \boldsymbol{x}_i - \boldsymbol{x}\rangle \leq 0\right\}, \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left\{\langle \mathbf{u}_k, \boldsymbol{x}_i - \boldsymbol{x}\rangle > 0\right\}\right\}, \tag{1.26}$$

where $\mathbf{u}_k \in \mathbb{S}^{d-1}$ and n_{proj} is the number of direction sampled on the hypersphere. It has the advantage of not supposing any underlying distribution for the embedding features.

Deep k-nearest neighbors. Sun et al. (2022) leverages non-parametric k-nearest neighbors approach (Fix and Hodges, 1989) in the latent space for OOD detection. The noteworthy feature of this approach, akin to data depth, lies in its lack of assumptions about the underlying distribution. Given an input sample $\boldsymbol{x}$, they compute the normalized embedding $\boldsymbol{z} = f^{(\ell)}(\boldsymbol{x}) / \|f^{(\ell)}(\boldsymbol{x})\|_2$ and the ordered set

$$\mathcal{Z} = \{s_i = \|\boldsymbol{z}_i - \boldsymbol{z}\|_2 \mid s_1 \leq s_2 \leq \cdots \leq s_n\}, \tag{1.27}$$

where $\boldsymbol{z}_i$, $i = \{1, \ldots, n\}$, are the normalized embeddings of the training samples. The decision function is finally given by $d(\boldsymbol{x}) = \mathbb{1}[-s_k \geq \gamma]$.

Max cosine similarity. Techapanurak et al. (2020); Zhou et al. (2021) calculates the maximum cosine similarity between the features of a test sample and class conditional average embedding vectors denoted as $\widehat{\mu}_y$, sometimes referred to as prototype vectors. The max cosine score writes:

$$s(\boldsymbol{x}) = \max_{y \in \mathcal{Y}} \frac{\widehat{\mu}_y^\top f^{(\ell)}(\boldsymbol{x})}{\|\widehat{\mu}_y^\top\|_2}. \tag{1.28}$$

Usually, ℓ is the index of the penultimate layer of the network.

Activation clipping. Sun et al. (2021) propose a feature truncation technique, denoted as ReAct for rectified activations, where the feature vector extracted from the penultimate layer of the network, $\boldsymbol{z}$, is truncated element-wise using a threshold r to obtain $\overline{\boldsymbol{z}}$. These truncated features are transformed into rectified logits using $\overline{f(\boldsymbol{x})} = \boldsymbol{W}^\top \overline{\boldsymbol{z}} + \boldsymbol{b}$, where $\boldsymbol{W}$ and $\boldsymbol{b}$ are the weights of the linear classifier on top of the penultimate features of a DL model. Their proposed score is the rectified free Energy, computed as:

$$s(\boldsymbol{x}) = -T \log \sum_{j=1}^{K} \exp\left(\frac{\overline{f(\boldsymbol{x})}}{T}_j\right). \tag{1.29}$$

The intuition is that OOD data activations demonstrate higher variance with skewness towards activation peaks on residual networks, making the resulting predictions overconfident. By truncating the activations, this problem is alleviated, and the separation between in- and out-of-distribution samples increases.

Kullback-Leibler (KL) divergence matching. Hendrycks et al. (2022) computes class probabilities prototyes on a validation set, i.e., $v_k = \mathbb{E}_{x' \sim \mathcal{X}_{\text{val}}}\left[f\left(\boldsymbol{x}'\right)\right]$ and computes the KL divergence (Kullback and Leibler, 1951) between the test sample's softmax probabilities and the class-conditional prototype. The minimal divergence is used as a similarity score:

$$s(\boldsymbol{x}) = \min_{y \in \mathcal{Y}} \mathrm{KL}\left[\mathrm{softmax}(f(\boldsymbol{x})) \| v_k\right]. \tag{1.30}$$

The KL divergence is a common dissimilarity measure between probability distributions with nice properties and a natural choice for OOD detection.

Gradient norm. Huang et al. (2021) takes as reference probability to the KL divergence the uniform probability distribution over the classes of the classifier, i.e., $v_k = [1/K, \ldots, 1/K]^\top$. They compute as similarity score the L_p-norm of the gradient of the KL divergence between the test sample's output probabilities and the uniform reference.

$$s(\boldsymbol{x}) = \left\| \frac{\partial \, \mathrm{KL}\left[v_k \| \, \mathrm{softmax}(f(\boldsymbol{x}))\right]}{\partial \boldsymbol{W}} \right\|_p \tag{1.31}$$

where $\boldsymbol{W}$ is the weight matrix of the last linear layer from a DL model. They observed that, for some NNs, the gradient norm of OOD samples would be abnormally high compared to inlier samples.

Training with outlier exposure. Training with outlier exposure (Hendrycks et al., 2019) leverages unlabeled auxiliary examples represented as $\boldsymbol{x}'$ drawn from the distribution q_X to enhance out-of-distribution (OOD) detection. The method involves training a new classifier by minimizing an objective function:

$$\mathcal{L} = \mathbb{E}_{(\boldsymbol{x},y) \sim p_{XY}}\left[-\log f_y(\boldsymbol{x})\right] + \lambda \cdot \mathbb{E}_{\boldsymbol{x}' \sim q_X}\left[L_{\mathrm{CE}}\left(f(\boldsymbol{x}'), v_k\right)\right] \tag{1.32}$$

where L_{CE} is the cross entropy loss. The objective aims to minimize the classical cross-entropy loss and the cross-entropy between the classifier's predictions on the auxiliary data $\boldsymbol{x}'$ and the uniform distribution $v_k = [1/K, \ldots, 1/K]^\top$. Additionally, a regularization factor λ is introduced to account for different scales between the terms of the loss.

Doctor. Granese et al. (2021) proposes a simple and flexible framework to detect whether a decision made by a model is likely to be correct or not. A key ingredient of the Doctor score is to fully exploit all available information contained in the soft-probabilities of the predictions. Since the true probability of error Pe is unknown and cannot be learned from samples, they rely on the following approximation:

$$1 - s(\boldsymbol{x}) = \sum_{y \in \mathcal{Y}} p_{\widehat{Y}|X}(y|\boldsymbol{x}) \Pr(\widehat{Y} \neq y|\boldsymbol{x}) = 1 - \sum_{y \in \mathcal{Y}} p^2_{\widehat{Y}|X}(y|\boldsymbol{x}). \tag{1.33}$$

They also use temperature scaling and input pre-processing. We show in Dadalto et al. (2024a) that this criterion can also be used for selective classification, and we propose a tight relationship between the optimal misclassification detector and the optimal rejection criterion.

1.7 Experimental Setup

The classical OOD detection experimental setup consists of choosing a pre-trained classifier with an associated in-distribution dataset and running an inference of a test set composed of a mixture of an in-distribution validation dataset and a different dataset without semantic or class overlap with the in-distribution one. The similarity scores are computed for every test sample, and their performance is measured. In the following, we will introduce the evaluation metrics in Section 1.7.1, models in Section 1.7.2, and datasets in Section 1.7.3.

1.7.1 Evaluation Metrics

In this subsection, we introduce the metrics used to evaluate OOD and misclassification detection performance that are borrowed from standard binary classification evaluation metrics. Two main quantities allow us to measure the performance of a method. The *false alarm rate* is the proportion of samples that are detected as being positive (OOD) while they are negative (IND). Mathematically, $\mathrm{FAR}(X) = \Pr\left(s(X) \leq \gamma | Z = 0\right)$. The *true detection rate* (or true positive rate (TPR)) is the proportion of positive (OOD) samples that are correctly classified as being positive (OOD). It is computed as $\mathrm{TPR}(X) = \Pr\left(s(X) \leq \gamma | Z = 1\right)$. The following derived metrics are the most common ones encountered in the literature that express these effects:

- **True Negative Rate at 95% True Positive Rate (TNR at TPR-95% (%)).** This metric measures the true negative rate (TNR) at a specific true positive rate (TPR). The operating point is chosen such that the TPR of the in-distribution test set is fixed to some value. Let TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative, respectively. In practice, we measure $\mathrm{TNR} = \mathrm{TN}/(\mathrm{FP} + \mathrm{TN})$, when $\mathrm{TPR} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$ is fixed. More rigorously, for a desired detection rate r, this fixes a threshold γ_r such that the corresponding TPR equals r. At this threshold, one then computes:

$$\Pr(s(X) \leq \gamma_r | z = 0) \quad \text{with} \quad \gamma_r \quad \text{s.t.} \quad \mathrm{TPR}(\gamma_r) = r. \tag{1.34}$$

  In the literature, it is common practice to set $r = 0.95$. Values are multiplied by 100 so that it is in percentage (%) in the benchmarks.

- **False Positive Rate at 95% True Positive Rate (FPR at TPR-95% (%)).** This metric measures the complementary event of the TNR at 95% TPR, or,

$$\Pr(s(X) \geq \gamma_r | z = 0) \quad \text{with} \;\; \gamma_r \;\; \text{s.t.} \;\; \text{TPR}(\gamma_r) = r. \tag{1.35}$$

The metric is also referred to as FPR for short. In practice, we measure FPR $= \text{FP}/(\text{FP} + \text{TN})$ for the confusion metrics obtained with threshold γ_r such that $r = 0.95$. Values are multiplied by 100 so that it is in percentage (%) in the benchmarks.

- **Area Under the Receiver Operating Characteristic (ROC) curve (AUROC).** The ROC curve is constructed by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold values. The area under this curve tells how much the detector can distinguish positive (OOD) and negative (IND) samples in a threshold-independent manner (Bradley, 1997). More rigorously, the AUROC corresponds to the probability that a randomly drawn negative sample ($X_{Z=0}$) has a higher score than a randomly drawn sample positive sample ($X_{Z=1}$): AUROC $= \Pr(s(X_{Z=0}) > s(X_{Z=1}))$. The ROC curve is parametrized by the threshold γ:
$$\gamma \mapsto (\Pr\big(s(X) \leq \gamma | Z = 0\big), \Pr\big(s(X) \leq \gamma | Z = 1\big)). \tag{1.36}$$
and the area under this curve writes in terms of the truncated cumulative distribution function (tcdf) $F_z(\gamma) = \Pr(s(X) \leq \gamma | Z = z)$ of the r.v. score $s(X)$.
$$\text{AUROC} = \int_0^1 F_0\left(F_1^{-1}(v)\right) dv \tag{1.37}$$
The estimation is usually done with trapezoidal integration, and the AUROC is often multiplied by 100 so that its values are given in percentage (%) in the benchmarks.

- **Area Under the Precision-Recall (PR) curve (AUPR (%)).** The PR curve (Davis and Goadrich, 2006) plots the precision, which is the actual proportion of positive (OOD) samples amongst all the samples predicted as positive (OOD), i.e., P $= \text{TP}/(\text{TP}+\text{FP})$ against the recall. The recall is equal to the true detection rate or true positive rate, i.e., R $= \text{TP}/(\text{TP} + \text{FN})$. The curve can be parametrized by the threshold γ in the form
$$\gamma \mapsto \big(\Pr\big(s(X) \leq \gamma | Z = 1\big), \Pr\big(Z = 1 | s(X) \leq \gamma\big)\big). \tag{1.38}$$
The AUPR is more relevant to unbalanced situations where the amount of positive or negative samples is much larger than the other.

- **Detection error of the best classifier (Err(%)).** Refers to the lowest classification error possible obtained by the detector, or Err $= (\text{FP} + \text{FN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN})$.

1.7.2 Models

In computer vision classification tasks, popular model architectures are Residual Convolutional Neural Networks (ResNet) (He et al., 2016), Vision Transformers (ViT) (Dosovitskiy et al., 2021), MobileNet (Howard et al., 2017) for fast inference and DenseNet (Huang et al., 2017) for a reduced number of parameters. For textual classification in natural language processing, transformer-based encoders like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) are prevalent choices. In this thesis, we will concentrate on the computer vision benchmark exclusively. Our works (Colombo et al., 2022; Darrin et al., 2024) proposes new methods for OOD detection in NLP benchmarks.

1.7.3 Datasets

Image Classification. To this date, two primary benchmarks exist for OOD detection on natural image classification: one is founded on CIFAR, and the other on ImageNet.

- **CIFAR.** The CIFAR-10 dataset (Krizhevsky et al., 2009) comprises 32x32 pixel natural images categorized into 10 distinct classes, such as airplanes, ships, birds, and more. Similarly, the CIFAR-100 dataset consists of natural images akin to those in CIFAR-10 but spanning 100 categories. Both datasets feature a training set containing 50,000 images and a test set of 10,000 images. The distribution of classes in both training and testing data is uniform. These datasets are made available under the MIT license.

- **ImageNet-1K.** The ImageNet-1K, also known as ILSVRC2012 (Deng et al., 2009), represents a challenging and realistic mid-sized dataset, encompassing approximately 1.28 million training examples and 50,000 labeled test instances distributed across 1000 distinct classes. This dataset is available under the BSD 3-Clause license.

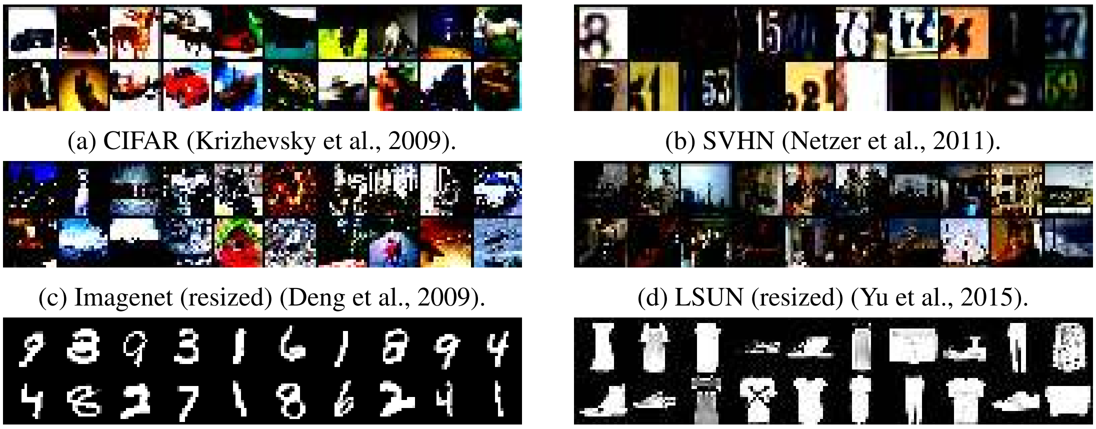
For the CIFAR benchmark, SVHN (Netzer et al., 2011), Tiny-ImageNet (Le and Yang, 2015), LSUN (Yu et al., 2015), iSUN (Xu et al., 2015), Textures (Cimpoi et al., 2014), Chars74K (de Campos et al., 2009), Places365 (Zhou et al., 2017), Gaussian noise, and Uniform noise are commonly used as OOD datasets. Fig. 1.1 show some examples of the images encountered in these datasets. For the large-scale benchmark, in addition to Textures and Places365, Species (Hendrycks et al., 2022), OpenImage-O (Wang et al., 2022), iNaturalist (Huang and Li, 2021), Sun (Huang and Li, 2021), and Semantic Shift Benchmark (Vaze et al., 2022) datasets are considered with the curated splits introduced by (Bitterwolf et al., 2023). Further details on the datasets are relegated to Appendix A.2.

1.8 Overview of the Manuscript

This section outlines the structure of the manuscript and the contents of each chapter.

Chapter 2 presents a method for OOD detection by building on the geodesic (Fisher-Rao) distance between the inlier and a proxy of the outlier data distributions. The discriminator combines confidence scores from the logits and features of a deep neural network through a unified formulation.

Chapter 3 introduces a simple unsupervised layer representations projection to perform OOD detection, identifying trajectories that are atypical from the behavior characterized by the training set. Often, methods that explore the multiple layers require a special architecture or a supervised objective. The presented method, on the other hand, is completely unsupervised and does not require any hyperparameter tuning.



(a) CIFAR (Krizhevsky et al., 2009).

(b) SVHN (Netzer et al., 2011).

(c) Imagenet (resized) (Deng et al., 2009).

(d) LSUN (resized) (Yu et al., 2015).

(e) MNIST (LeCun and Cortes, 2010).

(f) Fashion MNIST (Xiao et al., 2017).

Figure 1.1: Randomly selected images from a few datasets resized to 32×32.

Chapter 4 explores generalized data distribution drift detection. Most ML systems are evaluated in static scenarios, while, in practice, they encounter a dynamic and evolving environment. We propose a universal method for ensembling existing detectors by effectively transforming the problem into a multi-variate hypothesis test and leveraging established meta-analysis tools, resulting in a more effective detector with consolidated decision boundaries.

Chapter 5 proposes a new method to quantify uncertainty in machine learning predictions, especially focused on misclassification detection. Conventional uncertainty measures such as Shannon entropy do not provide an effective way to infer the real uncertainty associated with the model's predictions. We introduce a novel data-driven measure of uncertainty relative to an observer for misclassification detection by learning patterns in soft-predictions distribution.

Finally, Chapter 6 will conclude the research journey by summarizing this work's major findings and contributions. Also, we will offer insights into potential avenues for future research to continue advancing our understanding of out-of-distribution and misclassification detection.

1.9 List of Contributions and Publications

The main content of the dissertation will be based on the following works:

1. Chapter 2 is based on the results presented in Dadalto et al. (2022) entitled **Igeood: An Information Geometry Approach to Out-of-Distribution Detection**, that was accepted to the NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications and appeared in the proceedings of the International Conference on Learning Representations (ICLR) in 2022. The full list of authors is *Eduardo Dadalto Câmara Gomes*, Florence Alberge, Pierre Duhamel, and Pablo Piantanida, and it is available at ArXiv, abs/2203.07798.

2. Chapter 3 is based on the results presented in Dadalto et al. (2023b) entitled **Neural Trajectories for Out-of-Distribution Detection**, under submission. The full list of authors is *Eduardo Dadalto Câmara Gomes*, Pierre Colombo, Guillaume Staerman, Nathan Noiry, and Pablo Piantanida, and it is available at ArXiv, abs/2306.03522.

3. Chapter 4 is based on the results presented in Dadalto et al. (2023a) entitled **Combine and Conquer: A Meta-Analysis on Data Shift and Out-of-Distribution Detection**, that has been recently submitted to the Transactions on Machine Learning Research journal. The full list of authors is *Eduardo Dadalto Câmara Gomes*, Florence Alberge, Pierre Duhamel, and Pablo Piantanida.

4. Chapter 5 is based on the results presented in Dadalto et al. (2024b) entitled **A Data-Driven Measure of Relative Uncertainty for Misclassification Detection**, that has was accepted to the Neural Information Processing Systems Workshop entitled Mathematics of Modern Machine Learning (NeurIPS 2023 M3L) and to the proceedings of the International Conference on Learning Representations (ICLR) in 2024. The full list of authors is *Eduardo Dadalto Câmara Gomes**[1], Marco Romanelli*, Georg Pichler*, and Pablo Piantanida, and it is available at ArXiv, abs/2306.01710.

Other works I have contributed to during my PhD are:

5. **Beyond Mahalanobis-Based Scores for Textual OOD Detection** (Colombo et al., 2022), that appeared in the 35th Advances in Neural Information Processing Systems proceedings in 2022. The full list of authors is Pierre Colombo, *Eduardo Dadalto Câmara Gomes*, Guillaume Staerman, Nathan Noiry, and Pablo Piantanida, and it is available at ArXiv, abs/2211.13527.

6. **Unsupervised Layer-wise Score Aggregation for Textual OOD Detection** (Darrin et al., 2024), that has been accepted to the 38th Annual AAAI Conference on Artificial Intelligence (AAAI 2024). The full list of authors is Maxime Darrin, Guillaume Staerman, *Eduardo Dadalto Câmara Gomes*, Jackie Cheung, Pablo Piantanida, and Pierre Colombo, and it is available at ArXiv, abs/2302.09852.

7. **Trusting the Untrustworthy: A Cautionary Tale on the Pitfalls of Training-based Rejection Option** (Dadalto et al., 2024a) , that has been recently submitted to the journal Pattern Recognition Letters. The full list of authors is *Eduardo Dadalto Câmara Gomes**, Marco Romanelli*, Federica Granese, Siddharth Garg, and Pablo Piantanida.

During this thesis, I also collaborated closely with IBM-France for the ANR project entitled AIDA, a joint venture between public and private institutions towards Artificial Intelligence for Digital Automation. Notably, we delivered a practical demonstration in the format of an API (Application Programming Interface) of some of the techniques that will be presented in the following chapters for detecting OOD data in a real-world scenario. An URL[2] is provided to access the public version of the API. An open source library for accelerating research on generalized out-of-distribution (OOD) detection is publised alongside this thesis[3] (Dadalto, 2023).

[1] * indicates equal contribution.

[2] https://huggingface.co/spaces/edadaltocg/ood-detection

[3] https://github.com/edadaltocg/detectors

CHAPTER 2

IGEOOD: An Information Geometry Approach to Out-of-Distribution Detection

2.1 Introduction

In this chapter, we propose IGEOOD, a new unified and effective method to perform OOD detection by rigorously exploring the information-geometric properties of the feature space on various depths of a DNN. IGEOOD provides a flexible framework that applies to any pre-trained softmax neural classifier. A key ingredient of IGEOOD is the Fisher-Rao distance. This distance is used as an effective differential geometry tool for clustering, as a distance in the context of multivariate Gaussian pdfs (Pinele et al., 2020; Strapasson et al., 2016), among others applications.

We measure the dissimilarity between probability distributions (in and out) as the length of the shortest path within the manifold induced by the underlying class of distributions (i.e., the softmax probabilities of the neural classifier or the densities modeling the learned representations across the layers). By doing so, we can explore statistical invariances of the geometric properties of the learned features (Bronstein et al., 2021). Our method adapts to the various scenarios depending on the level of information access of the DNN. It uses only in-distribution samples but can also benefit (if available) from OOD samples or artificially generated outliers.

The contents of this chapter will be based on the papers Dadalto et al. (2021, 2022), a joint work with Florence Alberge, Pierre Duhamel, and Pablo Piantanida. The code is available at the url[1].

2.2 Summary of Contributions

Our work investigates the problem of OOD detection and advances state-of-the-art in different ways.

1. To the best of our knowledge, this is the first work studying *information geometry* tools to devise a unified metric for OOD detection. We derive an explicit characterization of the

[1] `https://www.github.com/edadaltocg/igeood`

Fisher-Rao distance based on the information-geometric properties of the softmax probabilities of the neural classifier and the class of multivariate Gaussian pdfs. In general terms, our Fisher-Rao-based metric measures the mismatch–in the geometry space–between the probability density functions of the pre-trained DNN classifier conditioned on test and in-distribution samples. Section 2.4 details IGEOOD.

2. Experiments on BLACK-BOX and GREY-BOX setups using various datasets, architectures, and classification tasks show that IGEOOD is competitive with state-of-the-art methods. In the BLACK-BOX setup, we assume that only the outputs, i.e., the logits of the DNN, are available. In the GREY-BOX setup, we allow access to all parameters of the network; however, the detection must be performed using only the output softmax probabilities. The latter permits input pre-processing, which introduces a small (additive) noise in the direction of the gradients w.r.t the test sample. This pre-processing allows for further discrimination between in- and out-of-distribution samples. Our benchmark contains two DNN architectures, three in-distribution datasets, and nine OOD datasets.

3. In a WHITE-BOX setting, we combine the logits with the low-level features of the DNN to leverage further useful statistical information of the encoded in-distribution data. We model the pre-trained latent representations as a mixture of Gaussian pdfs with a diagonal covariance matrix. Under this assumption, we derive a confidence score based on the Fisher-Rao distance between conditional pdfs corresponding to the test and the closest in-distribution samples. Experiments based on various datasets, architectures, and classification tasks clearly show consistent improvement of IGEOOD, achieving new state-of-the-art performance on a couple of benchmarks. In particular, we increased the average TNR at 95% TPR by 11.2% with tuning on OOD data and by 2.5% with tuning on adversarial data compared to Lee et al. (2018b).

2.3 Review of the Fisher-Rao Distance

In this section, we review some results from references Atkinson and Mitchell (1981); Pinele et al. (2020). We intend to clarify some basic concepts surrounding the Fisher-Rao distance (FRD) while motivating this measure in the context of OOD detection.

In a few words, Fisher-Rao's distance is given by the geodesic distance, i.e., the shortest path between points in a Riemannian space induced by a parametric family. Consider the family $\mathcal{C}$ of probability distributions over the class of discrete concepts or labels: $\mathcal{Y} = \{1, \dots, C\}$, denoted by $\mathcal{C} \triangleq \left\{q_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^C\right\}$. We are interested in measuring the distance between probability distributions $q_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})$ with respect to the testing input $\boldsymbol{x}$ and a population of inputs drawn accordingly to the in-distribution data set. To this end, we first need to characterize the Fisher-Rao distance for two inputs or for two probability distributions $q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}} \in \mathcal{C}$. Assume that the following regularity conditions hold (Atkinson and Mitchell, 1981):

(i) $\nabla_{\boldsymbol{x}}\, q_{\boldsymbol{\theta}}(y|\boldsymbol{x})$ exists for all $\boldsymbol{x}, y$ and $\boldsymbol{\theta} \in \Theta$;

(ii) $\sum\limits_{y \in \mathcal{Y}} \nabla_{\boldsymbol{x}}\, q_{\boldsymbol{\theta}}(y|\boldsymbol{x}) = 0$ for all $\boldsymbol{x}$ and $\boldsymbol{\theta} \in \Theta$;

(iii) $\boldsymbol{G}(\boldsymbol{x}) = \mathbb{E}_{Y\sim q_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})}\big[\nabla_{\boldsymbol{x}} \log q_{\boldsymbol{\theta}}(Y|\boldsymbol{x})\nabla_{\boldsymbol{x}}^{\top} \log q_{\boldsymbol{\theta}}(Y|\boldsymbol{x})\big]$ is positive definite for any $\boldsymbol{x}$ and $\boldsymbol{\theta} \in \Theta$.

Notice that if (i) holds, (ii) also holds immediately for discrete distributions over finite spaces (assuming that $\sum_{y\in\mathcal{Y}}$ and $\nabla_{\boldsymbol{x}}$ are interchangeable operations) as in our case. When (i)-(iii) are met, the variance of the differential form $\nabla_{\boldsymbol{x}}^{\top} \log q_{\boldsymbol{\theta}}(Y|\boldsymbol{x})d\boldsymbol{x}$ can be interpreted as the square of a differential arc length ds^2 in the space $\mathcal{C}$, which yields

$$ds^2 = \langle d\boldsymbol{x}, d\boldsymbol{x}\rangle_{\boldsymbol{G}(\boldsymbol{x})} = d\boldsymbol{x}^{\top}\boldsymbol{G}(\boldsymbol{x})d\boldsymbol{x}. \tag{2.1}$$

Thus, $\boldsymbol{G}$, which is the Fisher Information Matrix (FIM), can be adopted as a metric tensor. We now consider a curve $\boldsymbol{\gamma} : [0,1] \to \mathcal{X}$ connecting a pair of arbitrary points $\boldsymbol{x}$, $\boldsymbol{x}'$ in the input space $\mathcal{X}$, i.e., $\boldsymbol{\gamma}(0) = \boldsymbol{x}$ and $\boldsymbol{\gamma}(1) = \boldsymbol{x}'$. Notice that any curve $\boldsymbol{\gamma}$ induces a curve $q_{\boldsymbol{\theta}}(\cdot|\boldsymbol{\gamma}(t))$ for $t \in [0,1]$ in the space $\mathcal{C}$. The Fisher-Rao distance between the distributions $q_{\boldsymbol{\theta}} = q_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})$ and $q'_{\boldsymbol{\theta}} = q_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x}')$ will be denoted as $d_{R,\mathcal{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}})$ and is formally defined by the expression:

$$d_{R,\mathcal{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) \triangleq \inf_{\boldsymbol{\gamma}} \int_0^1 \sqrt{\frac{d\boldsymbol{\gamma}^{\top}(t)}{dt}\boldsymbol{G}(\boldsymbol{\gamma}(t))\frac{d\boldsymbol{\gamma}(t)}{dt}}, \tag{2.2}$$

where the infimum is taken over all piecewise smooth curves. This means that the FRD is the length of the *geodesic* between points $\boldsymbol{x}$ and $\boldsymbol{x}'$ using the FIM as the metric tensor. In general, the minimization of the functional in equation 2.2 is a problem that can be solved using the well-known Euler-Lagrange differential equation.

2.3.1 Derivation of Fisher-Rao Distance for the Class of Softmax Probability Distributions

The direct computation of the FIM of the family $\mathcal{C}$ with $q_{\boldsymbol{\theta}}(y|\boldsymbol{x})$ in the form of the softmax probability distribution function given by equation 2.16 can be shown to be singular, i.e., rank$(\boldsymbol{G}(\boldsymbol{x})) \leq C-1$, where $C-1$ is the number of degrees of freedom of the manifold $\mathcal{C}$. To overcome this issue, we introduce the probability simplex $\mathcal{P}$ defined by

$$\mathcal{P} = \left\{ q : \mathcal{Y} \to [0,1]^C : \sum_{y\in\mathcal{Y}} q(y) = 1 \right\}. \tag{2.3}$$

Next, we consider the following parametrization for any distribution $q \in \mathcal{P}$:

$$q(y|\boldsymbol{z}) = \frac{z_y^2}{4}, \quad y \in \{1,\dots,C\}. \tag{2.4}$$

From this expression, we consider the statistical manifold $\mathcal{D} = \big\{q(\cdot|\boldsymbol{z}) : \|\boldsymbol{z}\|^2 = 4, z_y \geq 0, \forall y \in \mathcal{Y}\big\}$. Note that the parameter vector $\boldsymbol{z}$ belongs to the positive portion of a sphere of radius 2 and is centered at the origin in $\mathbb{R}^C$. The computation of the FIM for $\boldsymbol{z}$ on $\mathcal{D}$ yields:

$$\begin{aligned}
\boldsymbol{G}(\boldsymbol{z}) &= \mathbb{E}_{q(y|\boldsymbol{z})}\left[\nabla_{\boldsymbol{z}} \log q(y|\boldsymbol{z}) \nabla_{\boldsymbol{z}}^{\top} \log q(y|\boldsymbol{z})\right] \\
&= \sum_{y \in \mathcal{Y}} \frac{z_y^2}{4}\left(\frac{2}{z_y}\boldsymbol{e}_y\right)\left(\frac{2}{z_y}\boldsymbol{e}_y^{\top}\right) \\
&= \sum_{y \in \mathcal{Y}} \boldsymbol{e}_y \boldsymbol{e}_y^{\top} \\
&= \boldsymbol{I},
\end{aligned} \tag{2.5}$$

where $\{\boldsymbol{e}_y\}$ are the canonical basis vectors in $\mathbb{R}^C$ and $\boldsymbol{I}$ is the identity matrix. From equation 2.5 we can conclude that the Fisher-Rao metric in this parametric space is equal to the Euclidean metric. Also, since the parameter vector lies on a sphere, the FRD between the distributions $q = q(\cdot|\boldsymbol{z})$ and $q' = q\left(\cdot|\boldsymbol{z}'\right)$ can be written as the radius of the sphere times the angle between the vectors $\boldsymbol{z}$ and $\boldsymbol{z}'$. Which leads to the expression:

$$d_{R,\mathcal{D}}\left(q, q'\right) = 2 \arccos\left(\frac{\boldsymbol{z}^{\top}\boldsymbol{z}'}{4}\right) = 2 \arccos\left(\sum_{y \in \mathcal{Y}} \sqrt{q(y|\boldsymbol{z}) q\left(y|\boldsymbol{z}'\right)}\right). \tag{2.6}$$

Finally, we can compute the FRD for softmax distributions in $\mathcal{C}$ as

$$d_{\mathrm{FR-Logits}}\left(q_{\boldsymbol{\theta}}, q_{\boldsymbol{\theta}}'\right) = 2 \arccos\left(\sum_{y \in \mathcal{Y}} \sqrt{q_{\boldsymbol{\theta}}(y|\boldsymbol{x}) q_{\boldsymbol{\theta}}\left(y|\boldsymbol{x}'\right)}\right), \tag{2.7}$$

obtaining the same form of equation 2.17. Notice that $0 \leq d_{\mathrm{FR-Logits}}\left(q_{\boldsymbol{\theta}}, q_{\boldsymbol{\theta}}'\right) \leq \pi$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X} \subseteq \mathbb{R}^C$, being zero when $q_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x}) = q_{\boldsymbol{\theta}}\left(\cdot|\boldsymbol{x}'\right)$ and maximum when the vectors $\left(q_{\boldsymbol{\theta}}(1|\boldsymbol{x}), \ldots, q_{\boldsymbol{\theta}}(C|\boldsymbol{x})\right)$ and $\left(q_{\boldsymbol{\theta}}\left(1|\boldsymbol{x}'\right), \ldots, q_{\boldsymbol{\theta}}\left(C|\boldsymbol{x}'\right)\right)$ are orthogonal.

2.3.2 Derivation of Fisher-Rao Distance for Multivariate Gaussian Distributions

Consider a broader statistical manifold $\mathcal{S} \triangleq \{p_{\boldsymbol{\theta}} = p(\boldsymbol{x}; \boldsymbol{\theta}) : \boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_m) \in \Theta\}$ of multivariate differential probability density functions. The Fisher information matrix $\boldsymbol{G}(\boldsymbol{\theta}) = [g_{ij}(\boldsymbol{\theta})]$ in this parametric space is provided by:

$$\begin{aligned}
g_{ij}(\boldsymbol{\theta}) &= \mathbb{E}_{\boldsymbol{\theta}}\left(\frac{\partial}{\partial \theta_i} \log p(\boldsymbol{x}; \boldsymbol{\theta}) \frac{\partial}{\partial \theta_j} \log p(\boldsymbol{x}; \boldsymbol{\theta})\right) \\
&= \int \frac{\partial}{\partial \theta_i} \log p(\boldsymbol{x}; \boldsymbol{\theta}) \frac{\partial}{\partial \theta_j} \log p(\boldsymbol{x}; \boldsymbol{\theta}) p(\boldsymbol{x}; \boldsymbol{\theta}) dx.
\end{aligned} \tag{2.8}$$

Next, consider a multivariate Gaussian distribution:

$$p(\boldsymbol{x}; \boldsymbol{\mu}, \Sigma) = \frac{(2\pi)^{-\left(\frac{n}{2}\right)}}{\sqrt{\mathrm{Det}(\Sigma)}} \exp\left(-\frac{(\boldsymbol{x} - \boldsymbol{\mu})^{\top} \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})}{2}\right), \tag{2.9}$$

where $\boldsymbol{x} \in \mathbb{R}^k$ is the variable vector, $\boldsymbol{\mu} \in \mathbb{R}^k$ is the mean vector, $\Sigma \in P_k(\mathbb{R})$ is the covariance matrix, and $P_k(\mathbb{R})$ is the space of k positive definite symmetric matrices. We can define the statistical manifold composed by these distributions as $\mathcal{M} = \{p_{\boldsymbol{\theta}}; \boldsymbol{\theta} = (\boldsymbol{\mu}, \Sigma) \in \mathbb{R}^k \times P_k(\mathbb{R})\}$. By substituting equation 2.9 in equation 2.8, we can derive the Fisher information matrix for this parametrization, obtaining:

$$g_{ij}(\boldsymbol{\theta}) = \frac{\partial \boldsymbol{\mu}^\top}{\partial \theta_i} \Sigma^{-1} \frac{\partial \boldsymbol{\mu}}{\partial \theta_j} + \frac{1}{2} \operatorname{tr}\left(\Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_i} \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_i} \right), \tag{2.10}$$

which induces the following square differential arc length in $\mathcal{M}$:

$$ds^2 = d\boldsymbol{\mu}^\top \Sigma^{-1} d\boldsymbol{\mu} + \frac{1}{2} \operatorname{tr}\left[\left(\Sigma^{-1} d\Sigma \right)^2 \right]. \tag{2.11}$$

Here, $d\boldsymbol{\mu} = (d\mu_1, \ldots, d\mu_n) \in \mathbb{R}^k$ and $d\Sigma = [d\sigma_{ij}] \in P_k(\mathbb{R})$. We observe that this metric is invariant to affine transformations (Pinele et al., 2020), i.e., for any $(\boldsymbol{c}, Q) \in \mathbb{R}^k \times GL_k(\mathbb{R})$, with $GL_k(\mathbb{R})$ the space of non-singular order k matrices, the map $(\boldsymbol{\mu}, \Sigma) \mapsto \left(Q\boldsymbol{\mu} + \boldsymbol{c}, Q\Sigma Q^\top\right)$ is an isometry in $\mathcal{M}$. Thus, the Fisher-Rao distance between two multivariate normal distributions with parameters $\boldsymbol{\theta}_1 = (\boldsymbol{\mu}_1, \Sigma_1)$ and $\boldsymbol{\theta}_2 = (\boldsymbol{\mu}_2, \Sigma_2)$ in $\mathcal{M}$ satisfies:

$$d_{R,\mathcal{M}}\left(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\right) = d_{R,\mathcal{M}}\left(\left(Q\boldsymbol{\mu}_1 + \boldsymbol{c}, Q\Sigma_1 Q^\top \right), \left(Q\boldsymbol{\mu}_2 + \boldsymbol{c}, Q\Sigma_2 Q^\top \right) \right). \tag{2.12}$$

Unfortunately, a closed-form solution for the Fisher-Rao distance remains unknown. This is still an open problem for an arbitrary covariance matrix Σ and mean vector μ. Fortunately, the FRD is known for the univariate case and, hence, for the submanifold where Σ is diagonal. Notice that in this case, equation 2.11 admits an additive form.

From Pinele et al. (2020), we obtain the analytical expression of the Fisher-Rao in the 2-dimensional submanifold of univariate Gaussian probability distributions $\mathcal{M}_2 = \{p_{\boldsymbol{\theta}} : \boldsymbol{\theta} = (\mu, \sigma^2) \in \mathbb{R} \times (0, +\infty)\}$:

$$\rho_{\mathrm{FR}}\left(\left(\mu_1, \sigma_1^2\right), \left(\mu_2, \sigma_2^2\right) \right) = \sqrt{2} \log \frac{\left| \left(\frac{\mu_1}{\sqrt{2}}, \sigma_1 \right) - \left(\frac{\mu_2}{\sqrt{2}}, -\sigma_2 \right) \right| + \left| \left(\frac{\mu_1}{\sqrt{2}}, \sigma_1 \right) - \left(\frac{\mu_2}{\sqrt{2}}, \sigma_2 \right) \right|}{\left| \left(\frac{\mu_1}{\sqrt{2}}, \sigma_1 \right) - \left(\frac{\mu_2}{\sqrt{2}}, -\sigma_2 \right) \right| - \left| \left(\frac{\mu_1}{\sqrt{2}}, \sigma_1 \right) - \left(\frac{\mu_2}{\sqrt{2}}, \sigma_2 \right) \right|}, \tag{2.13}$$

where $|\cdot|$ is the Euclidian norm in $\mathbb{R}^2$ and σ denotes the standard deviation. Consequently, the FRD for Gaussian distributions with diagonal covariance matrix $\Sigma = \operatorname{diag}\left(\sigma_1^2, \sigma_2^2, \ldots, \sigma_k^2\right)$ in the $2k$-dimensional statistical submanifold $\mathcal{M}_D = \left\{ p_{\boldsymbol{\theta}} : \boldsymbol{\theta} = (\boldsymbol{\mu}, \Sigma), \Sigma = \operatorname{diag}\left(\sigma_1^2, \sigma_2^2, \ldots, \sigma_k^2\right), \sigma_i > 0, i = 1, \ldots, k \right\}$ is

$$d_{\mathrm{FR-Gauss}}\left(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\right) = \sqrt{\sum_{i=1}^{k} d_{R,\mathcal{M}_2}\Big(\left(\mu_{1i}, \sigma_{1i}\right), \left(\mu_{2i}, \sigma_{2i}\right) \Big)^2}. \tag{2.14}$$

2.3.3 Fisher-Rao vs. Mahalanobis Distance

There is an intricate relationship between the FRD for multivariate Gaussian distributions and the Mahalanobis distance. We borrow the result from Pinele et al. (2020), which states that in the

k-dimensional submanifold $\mathcal{M}_\Sigma$ of $\mathcal{M}$ where Σ is constant, i.e., $\mathcal{M}_\Sigma = \{p_{\boldsymbol{\theta}} : \boldsymbol{\theta} = (\boldsymbol{\mu}, \Sigma), \Sigma = \Sigma_0 \in P_k(\mathbb{R})\}$, the Fisher-Rao distance $d_{R,\mathcal{M}_\Sigma}$ between two distributions is given by the Mahalanobis distance (Mahalanobis, 1936):

$$d_{R,\mathcal{M}_\Sigma}\big(\mathcal{N}(\boldsymbol{\mu}_1, \Sigma), \mathcal{N}(\boldsymbol{\mu}_2, \Sigma)\big) = \sqrt{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \Sigma^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)}. \tag{2.15}$$

The Mahalanobis distance is also used for OOD detection (Lee et al., 2018b), and its performance is compared to the FRD through several experiments in Section 2.5. Since the covariance matrix for the hidden layers' outputs is often not full rank, the pseudo-inverse is calculated instead of the inverse.

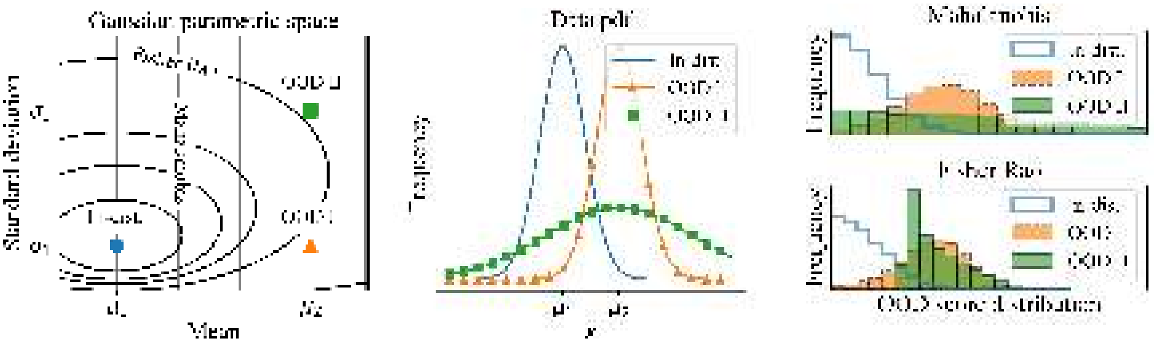
2.4 IGEOOD: OOD Detection Using the Fisher-Rao Distance

This section introduces IGEOOD, a flexible framework for OOD detection. IGEOOD is implemented in two ways: at the level of the logits using temperature scaling (Section 2.4.2), which mitigates the high-confidence scores assigned to OOD examples, and layer-wise level (Section 2.4.3). The key ingredient of IGEOOD is the Fisher-Rao distance that allows for effective differentiation between in-distribution and out-of-distribution samples. This distance measures the dissimilarity between two probability models within a class of probability distributions by calculating the geodesic distance between two points on the learned manifold. This measure connects information geometry and differential geometry through the R. Fisher information matrix (Fisher, 1922). Closed-form expressions of this distance are known to multivariate normal distributions under certain assumptions, among others distributions (Pinele et al., 2020).

2.4.1 Empirical Motivation

We introduce a simple example to demonstrate conceptually how Fisher-Rao distance is instrumental to OOD detection. It should be noted that this example is limited to one dimension. However, we expect similar behavior with more complex data under the Gaussianity assumptions.

Consider the case where we try to distinguish between samples from distinct Gaussian distributions on 1D. Assume that the in-distribution data follows a Gaussian $\mathcal{N}(\mu_1, \sigma_1)$ while OOD data is drawn according to either $\mathcal{N}(\mu_2, \sigma_1)$ or $\mathcal{N}(\mu_2, \sigma_2)$. These distributions are illustrated in Figures 2.1a and 2.1b. In this setup, distance-based approaches, which are invariant to the variance of the distributions, would have the performance limited to the information given by the difference between the means of the underlying distributions. For instance, in the case of the Mahalanobis distance, we would rely on our discrimination on the difference between the sample and the in-distribution mean, rescaled by the in-distribution standard deviation only, but nothing further could be obtained. However, suppose we can estimate OOD standard deviations from actual or pseudo-OOD data. In that case, we expect the Fisher-Rao distance between Gaussian distributions to be more effective in distinguishing between distributions. Figure 2.1c shows that the Fisher-Rao distance distinguishes better between “In-dist." and “OOD II" samples, while the other distances fail.



(a) Contour lines. (b) Synthetic data distribution. (c) OOD detection score histogram.

Figure 2.1: Example comparing Fisher-Rao and Mahalanobis distances to distinguish between 1D Gaussian distributions, showcasing the motivation to use of Fisher-Rao metric for OOD detection.

2.4.2 IGEOOD Score Using the Softmax Outputs

The Fisher-Rao distance (Atkinson and Mitchell, 1981) takes as input two probability distributions. For the classification problem, we can take the temperature T scaled softmax function (Eq. (2.16)) as an approximation of a class-conditional probability distribution:

$$q_{\boldsymbol{\theta}}\left(y|f(\boldsymbol{x});T\right) \triangleq \frac{\exp\left(f_y(\boldsymbol{x})/T\right)}{\sum_{y'\in\mathcal{Y}}\exp\left(f_{y'}(\boldsymbol{x})/T\right)}, \tag{2.16}$$

where $f:\mathcal{X}\to\mathbb{R}^C$ is a vectorial function with $f\triangleq\left(f_1,f_2,\ldots,f_C\right)$ and $f_y(\cdot)$ denotes the y-th logits output value of the DNN classifier. The Fisher-Rao distance $d_{\mathrm{FR-Logits}}$ between two distributions resulting from the softmax probability evaluated at two data points is (see Section 2.3):

$$d_{\mathrm{FR-Logits}}\big(q_{\boldsymbol{\theta}}(\cdot|f(\boldsymbol{x})),q_{\boldsymbol{\theta}}(\cdot|f(\boldsymbol{x}'))\big) \triangleq 2\arccos\left(\sum_{y\in\mathcal{Y}}\sqrt{q_{\boldsymbol{\theta}}\big(y|f(\boldsymbol{x})\big)q_{\boldsymbol{\theta}}\big(y|f(\boldsymbol{x}')\big)}\right). \tag{2.17}$$

Class conditional centroid estimation. We model the training dataset class-conditional posterior distribution by calculating the centroid of the logits representations of this set. Precisely, we compute the *empirical centroid* for the logits of each class $y\in\mathcal{Y}=\{1,\ldots,C\}$ of the in-distribution training dataset $\mathcal{D}_N$ corresponding to the Fisher-Rao distance, i.e.,

$$\boldsymbol{\mu}_y \triangleq \min_{\boldsymbol{\mu}\in\mathbb{R}^C}\frac{1}{N_y}\sum_{\forall\, i\,:\, y_i=y} d_{\mathrm{FR-Logits}}\big(q_{\boldsymbol{\theta}}(\cdot|f(\boldsymbol{x}_i)),q_{\boldsymbol{\theta}}(\cdot|\boldsymbol{\mu})\big), \tag{2.18}$$

where N_y is the amount of training examples with label y. We optimize this expression offline using SGD algorithm, where the parameter to be tuned is $\boldsymbol{\mu}$ in the logits space. This is equivalent to finding the centroid of a cluster using the Fisher-Rao distance after each example has been assigned to a cluster.

OOD and confidence score. Using the softmax probability, we can define a confidence score to be the minimum of the Fisher-Rao distance between $f(\boldsymbol{x})$ and the class-conditional centroids. Thus, the estimated class $\widehat{y}_{\mathrm{FR}}$ follows as:

$$\widehat{y}_{\mathrm{FR}}(\boldsymbol{x}) \triangleq \arg\min_{y \in \mathcal{Y}} d_{\mathrm{FR-Logits}}\big(q_{\boldsymbol{\theta}}(\cdot|f(\boldsymbol{x})), q_{\boldsymbol{\theta}}(\cdot|\boldsymbol{\mu}_y)\big). \tag{2.19}$$

However, we obtained slightly better OOD detection performance by using Eq. (2.20) instead of the minimal value. A likely explanation would be that this metric uses extra information from the other logits dimensions. Thus, we propose the Fisher-Rao distance-based OOD detection score $\mathrm{FR}_0(\boldsymbol{x})$ for the logits to be the sum of the distances between $f(\boldsymbol{x})$ and each individual class conditional centroid $\boldsymbol{\mu}_y$ given by Eq. (2.18). By taking the sum instead of the minimal distance, we leverage useful information related to the example's confidence score for each class y. We denote it by

$$\mathrm{FR}_0(\boldsymbol{x}) \triangleq \sum_{y \in \mathcal{Y}} d_{\mathrm{FR-Logits}}\big(q_{\boldsymbol{\theta}}(\cdot|f(\boldsymbol{x})), q_{\boldsymbol{\theta}}(\cdot|\boldsymbol{\mu}_y)\big). \tag{2.20}$$

Input pre-processing. In consonance with the literature (Liang et al., 2018b; Liu et al., 2020; Lee et al., 2018b), we also perform input pre-processing to enhance the detection between in-distribution and OOD samples and potentially improve OOD detection performance for the GREY-BOX discriminator. We add small magnitude perturbations ε in a Fast Gradient-Sign Method-style (FGSM) (Goodfellow et al., 2015) to each test sample $\boldsymbol{x}$ to increase the proposed metric, that is:

$$\widetilde{\boldsymbol{x}} = \boldsymbol{x} + \varepsilon \odot \mathrm{sign}\left[\nabla_{\boldsymbol{x}} \mathrm{FR}_0(\boldsymbol{x})\right]. \tag{2.21}$$

The OOD detector. The detector consists of a threshold-based function for discriminating between in-distribution and OOD data. This threshold δ and parameters are set so that the true positive rate, i.e., the in-distribution samples correctly classified as in-distribution, becomes 95%. Mathematically, the BLACK-BOX OOD detector g_{BB} and the GREY-BOX OOD detector g_{GB} writes:

$$g_{\mathrm{BB}}(\boldsymbol{x};\delta,T) = \begin{cases} 1 & \text{if } \mathrm{FR}_0\left(\boldsymbol{x}\right) \leq \delta \\ 0 & \text{if } \mathrm{FR}_0\left(\boldsymbol{x}\right) > \delta \end{cases} \quad \text{and} \quad g_{\mathrm{GB}}(\widetilde{\boldsymbol{x}};\delta,T,\varepsilon) = \begin{cases} 1 & \text{if } \mathrm{FR}_0\left(\widetilde{\boldsymbol{x}}\right) \leq \delta \\ 0 & \text{if } \mathrm{FR}_0\left(\widetilde{\boldsymbol{x}}\right) > \delta \end{cases}. \tag{2.22}$$

2.4.3 IGEOOD Score Leveraging Latent Features

For each layer, we define a set of class-conditional Gaussian distributions with diagonal standard deviation matrix $\boldsymbol{\sigma}^{(\ell)}$ and class-conditional mean $\boldsymbol{\mu}_y^{(\ell)}$, where $y \in \{1,\dots,C\}$ and ℓ is the index of the latent feature. We compute the empirical estimates of these parameters according to

$$\boldsymbol{\mu}_y^{(\ell)} = \frac{1}{N_y}\sum_{\forall i\,:\,y_i=y} f^{(\ell)}\left(\boldsymbol{x}_i\right), \quad \text{and} \quad \boldsymbol{\sigma}^{(\ell)} = \mathrm{diag}\left(\sqrt{\frac{1}{N}\sum_{y\in\mathcal{Y}}\sum_{\forall i\,:\,y_i=y}\left(f_j^{(\ell)}\left(\boldsymbol{x}_i\right) - \mu_{y,j}^{(\ell)}\right)^2}\right), \tag{2.23}$$

where $j \in \{1,\dots,k\}$, k is the size of feature ℓ, and $f^{(\ell)}(\cdot)$ is the output of the network for feature ℓ. The Fisher-Rao distance ρ_{FR} between two arbitrary *univariate* Gaussian pdfs $\mathcal{N}(\mu_1,\sigma_1^2)$ and

$\mathcal{N}(\mu_2, \sigma_2^2)$ is given by Section 2.3 above.

$$\rho_{\mathrm{FR}}\left((\mu_1, \sigma_1), (\mu_2, \sigma_2)\right) = \sqrt{2} \log \frac{\left|\left(\frac{\mu_1}{\sqrt{2}}, \sigma_1\right) - \left(\frac{\mu_2}{\sqrt{2}}, -\sigma_2\right)\right| + \left|\left(\frac{\mu_1}{\sqrt{2}}, \sigma_1\right) - \left(\frac{\mu_2}{\sqrt{2}}, \sigma_2\right)\right|}{\left|\left(\frac{\mu_1}{\sqrt{2}}, \sigma_1\right) - \left(\frac{\mu_2}{\sqrt{2}}, -\sigma_2\right)\right| - \left|\left(\frac{\mu_1}{\sqrt{2}}, \sigma_1\right) - \left(\frac{\mu_2}{\sqrt{2}}, \sigma_2\right)\right|}. \tag{2.24}$$

Similarly, the Fisher-Rao distance $d_{\mathrm{FR-Gauss}}$ between two *multivariate* Gaussian pdfs with diagonal standard deviation matrix is derived from the univariate case and is given by

$$d_{\mathrm{FR-Gauss}}\big((\boldsymbol{\mu}, \boldsymbol{\sigma}), (\boldsymbol{\mu}', \boldsymbol{\sigma}')\big) = \sqrt{\sum_{i=1}^{k} \rho_{\mathrm{FR}}\left((\mu_i, \sigma_{i,i}), \left(\mu_i', \sigma_{i,i}'\right)\right)^2}, \tag{2.25}$$

where k is the cardinality of the distributions $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ and $\mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\sigma}')$, μ_i is the i-th component of the vector $\boldsymbol{\mu}$, and $\sigma_{i,i}$ is the entry with index (i, i) of the standard deviation matrix $\boldsymbol{\sigma}$.

Experimental support for a diagonal Gaussian mixture model. It is known that intermediate features of a DNN can be valuable for detecting abnormal samples as demonstrated by Lee et al. (2018b). Nonetheless, we observed that the latent features covariance matrices are often *ill-conditioned* and are diagonal dominant. In other words, the condition number of the covariance matrix often diverges, and the magnitude of the diagonal entry in a row is greater than or equal to the sum of all the other entries in that row for most rows. Thus, a diagonal covariance matrix will be a favorable compromise for OOD detection.

Fisher-Rao distance-based feature-wise confidence score. We derive a confidence score by applying the Fisher-Rao distance between the test sample $\boldsymbol{x}$ and the closest class-conditional diagonal Gaussian distribution. Contrarily to the logits, taking the sum did not improve results, so we kept the minimal distance. We can consider two scenarios: **(i)** We do not have access to any validation OOD data whatsoever. In this case, the natural choice is to model the test samples as Gaussian distribution with the same diagonal standard deviation as the learned representation, i.e.,

$$\mathrm{FR}_\ell(\boldsymbol{x}) = \min_{y \in \mathcal{Y}} d_{\mathrm{FR-Gauss}}\big((\boldsymbol{x}, \boldsymbol{\sigma}^{(\ell)}), (\boldsymbol{\mu}_y^{(\ell)}, \boldsymbol{\sigma}^{(\ell)})\big); \tag{2.26}$$

and **(ii)** we dispose of a validation OOD dataset on which the features' diagonal standard deviation matrices $\boldsymbol{\sigma}'^{(\ell)}$ and the means $\boldsymbol{\mu}'^{(\ell)}$ can be estimated, as well as the quantity:

$$\mathrm{FR}_\ell'(\boldsymbol{x}) = \min_{y \in \mathcal{Y}} d_{\mathrm{FR-Gauss}}\big((\boldsymbol{x}, \boldsymbol{\sigma}^{(\ell)}), (\boldsymbol{\mu}'^{(\ell)}, \boldsymbol{\sigma}'^{(\ell)})\big). \tag{2.27}$$

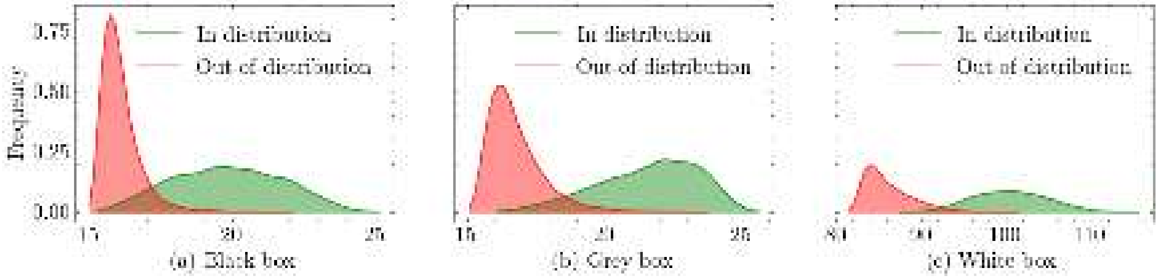
This validation dataset could be obtained from a synthetic dataset, a dataset different from the testing one, or even by adversarially creating OOD data by attacking the classifier model on the training dataset.

Feature ensemble. To further improve performance, we combine the confidence scores of the logits and the ones from the low-level features through a linear combination. Similarly to the strategy in Lee et al. (2018b), we choose the weights α_0, α_ℓ and $\alpha_\ell' \in \mathbb{R}$ by training a logistic regression detector using validation samples. Thus, we ensure that the metric emphasizes features that demonstrate a greater capacity for detecting abnormal samples. IGEOOD score for the WHITE-BOX setting is:

$$\mathrm{FR}(\boldsymbol{x}) \triangleq \alpha_0 \mathrm{FR}_0(\boldsymbol{x}) + \sum_{\ell} \alpha_\ell \cdot \mathrm{FR}_\ell(\boldsymbol{x}) + \alpha'_\ell \cdot \mathrm{FR}'_\ell(\boldsymbol{x}), \tag{2.28}$$

where FR_0 is given by equation 2.20, FR_ℓ is given by equation 2.26 and FR' considers a different validation diagonal covariance matrix for the test samples (equation 2.27). We also apply input pre-processing similarly to the GREY-BOX setting (equation 2.21), obtaining $\mathrm{FR}(\widetilde{\boldsymbol{x}})$ as final score.

Unified metric. For the three settings, the metric is the same but has different formulations given the family of the distributions. For the DNN outputs, we use the softmax posterior probability distribution formulation. The intermediate layers it is under the model of diagonal Gaussian pdfs. *Therefore, we have derived a unified OOD detection framework that combines a single distance for both the softmax outputs and the latent features of a neural network.* Figure 2.2 illustrates how each technique contributes to separating in-distribution and OOD samples.



(a) Black box (b) Grey box (c) White box

Figure 2.2: Probability distributions of the IGEOOD score under three different settings for a pre-trained DenseNet on CIFAR-10 for in-distribution and OOD data (TinyImageNet downsampled).

2.5 Experimental Results

2.5.1 Setup

The experimental setup follows the setting established by Hendrycks and Gimpel (2017), Liang et al. (2018b) and Lee et al. (2018b). We use two *pre-trained* deep neural networks architectures for image classification tasks: a Dense Convolutional Network (DenseNet-BC-100) (Huang et al., 2017) and a Residual Neural Network (ResNet-34) (He et al., 2016). We take as *in-distribution data* images from CIFAR-10 (Krizhevsky et al., 2009), CIFAR-100 and SVHN (Netzer et al., 2011) datasets.

We measure the effectiveness of the OOD detectors with three standard *evaluation metrics*: (i) The true negative rate at 95% true positive rate (TNR at TPR-95%); (ii) the area under the receiving operating curve (AUROC); and (iii) the area under the precision-recall curve (AUPR). For the BLACK-BOX and GREY-BOX experimental settings, we *tune hyperparameters* for all of the OOD detectors only based on the DNN classifier architecture, the in-distribution dataset and a validation dataset. The iSUN (Xu et al., 2015) dataset is chosen as a source of OOD validation data, independently from OOD test data. We choose the parameters that maximize the TNR at TPR-95% on the validation OOD dataset. For the WHITE-BOX framework, we allow both the benchmark and our method to tune either on adversarially generated data from in-distribution training samples or a separate validation dataset containing $1,000$ images from the OOD test dataset with feature ensemble described in Section 2.4.3. In this case, we evaluate the performance of the remaining test samples.

2.5.2 Results for the Black-Box and the Grey-Box Setups

For comparing Igeood under the hypothesis of a Black-Box scenario, we consider the Baseline (Hendrycks and Gimpel, 2017) method, ODIN (Liang et al., 2018b) with temperature scaling only, and the free-energy-based metric (Liu et al., 2020) with temperature scaling only. The results for the Black-Box setting are available in Table 2.1, where we show the average and one standard deviation OOD detection performance for each of the eight OOD detection method in six different image classification contexts (couple DNN model and in-distribution dataset). For comparison under the Grey-Box assumption, we consider ODIN and the free-energy-based methods, both with input pre-processing. The results for the Grey-Box setup are provided in Table 2.2. For the Black-Box setting, Igeood slight improves the benchmark by less than 1% in TNR at TPR-95%. While for the Grey-Box setting, results show Igeood is outperformed by $<1\%$ in a few benchmarks by ODIN, which is greatly improved by input pre-processing techniques.

Table 2.1: Average and standard deviation OOD detection performance across eight OOD datasets for each model and in-distribution dataset in a Black-Box setting.

		TNR at TPR-95%	AUROC
Model	In-dist.	Baseline / ODIN / Energy / Igeood (ours)	
DenseNet	C-10	52.5±16/**66.8**±20/65.3±23/65.6±23	91.8±3.2/**92.8**±4.6/92.1±5.3/92.3±5.1
	C-100	15.9±6.8/20.5±9.5/20.3±9.6/**20.7**±9.8	69.1±15/71.6±20/71.6±20/**73.2**±17
	SVHN	68.4±14/68.8±20/70.2±17/**72.1**±15	**92.3**±4.0/87.3±14/90.1±5.9/90.9±5.3
ResNet	C-10	41.7±16/51.9±15/56.3±13/**56.7**±13	89.6±3.1/90.4±3.1/90.4±3.0/**90.5**±3.0
	C-100	15.0±5.5/16.0±6.3/16.3±7.1/**16.4**±6.8	74.0±1.9/75.2±1.7/**75.5**±1.9/**75.5**±1.7
	SVHN	76.2±7.8/77.7±7.9/78.0±7.9/**78.3**±8.0	**92.2**±2.9/91.4±3.2/91.4±3.2/91.7±3.2
Average and Std.		44.9±24/50.3±24/51.1±24/**51.6**±24	84.8±9.5/84.8±8.3/85.2±8.4/**85.7**±8.0

Table 2.2: Average and standard deviation OOD detection performance across eight OOD datasets for each model and in-distribution dataset in a Grey-Box setting.

		TNR at TPR-95%	AUROC
Model	In-dist.	ODIN / Energy / Igeood	
DenseNet	C-10	**66.8**±23/64.8±25/65.3±24	**91.9**±6.2/91.5±6.4/**91.9**±6.0
	C-100	**25.5**±14/24.8±13/25.0±13	76.6±12/76.4±12/**78.2**±8.2
	SVHN	**75.4**±15/70.6±17/72.4±16	**91.6**±5.4/89.2±6.9/90.0±6.3
ResNet	C-10	57.3±20/57.7±19/**57.8**±19	**89.2**±5.4/88.7±5.3/89.0±5.2
	C-100	**31.1**±22/30.2±22/30.2±22	**76.9**±11/74.4±12/74.3±12
	SVHN	78.5±7.8/78.5±7.9/**78.8**±7.8	90.4±3.4/**90.9**±3.4/90.7±3.3
Average and Std.		**55.8**±21/54.4±20/54.9±20	**86.1**±6.7/85.2±7.0/85.7±6.8

2.5.3 Hyperparameters Tuning

Temperature scaling and input pre-processing. For temperature T, we ran a Bayesian optimization for 500 epochs in the interval of temperature values between 1 and 1000, where the objective function was to maximize the TNR at TPR-95$\%$ metric for the validation set. We took the best temperature among five runs with different random seeds. For the input pre-processing noise magnitude ε tuning, we ran a grid search optimization with 21 equally spaced values in the interval $[0, 0.002]$. Table 2.3 shows the best hyperparameters we found for the methods in the BLACK-BOX, GREY-BOX, and WHITE-BOX settings.

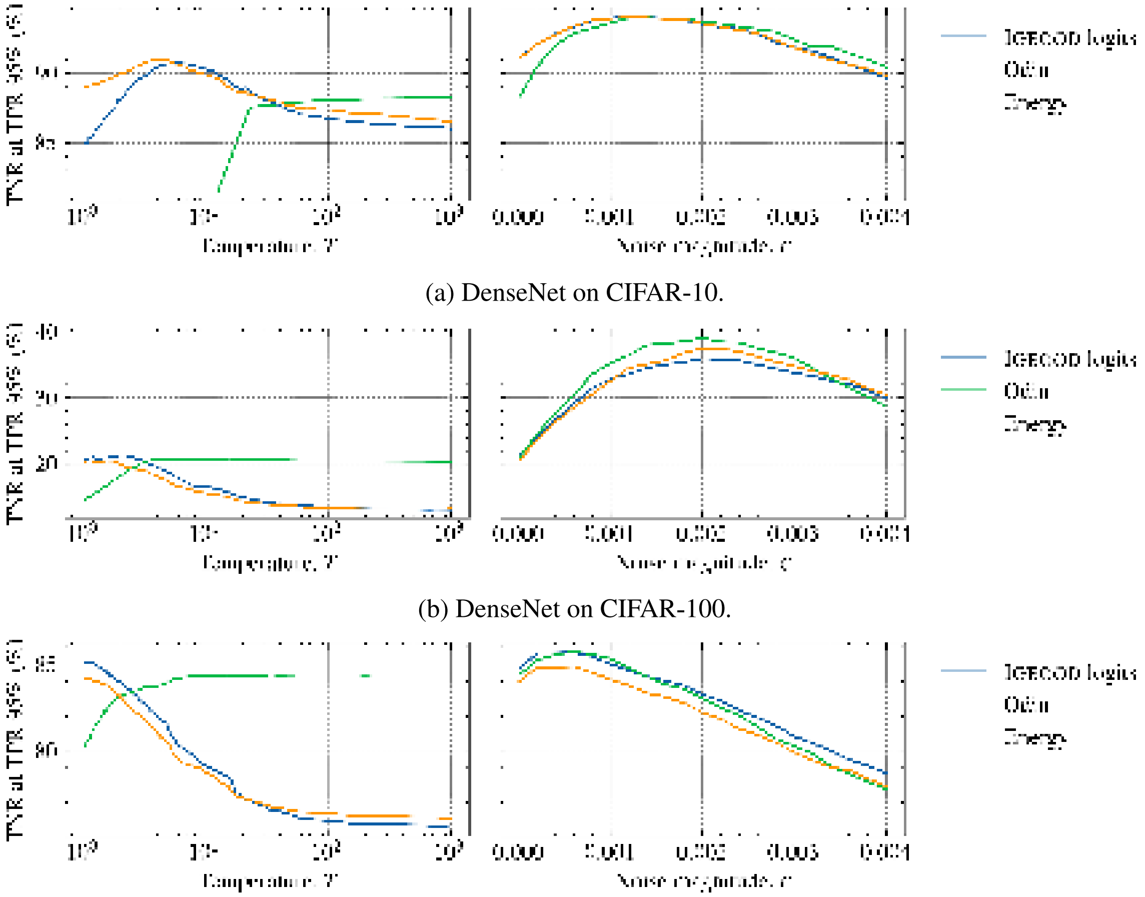
Table 2.3: Best temperatures T for the BLACK-BOX setup, best temperature and noise magnitude (T, ε) for the GREY-BOX setup, and best ε for the Mahalanobis score and $(T,\ \varepsilon)$ for IGEOOD and IGEOOD+ in the WHITE-BOX setup with adversarial tuning.

	In-dist.	BLACK-BOX			GREY-BOX			WHITE-BOX	
Model	dataset	ODIN	Energy	IGEOOD	ODIN	Energy	IGEOOD	Maha.	IGEOOD,+
DenseNet	C-10	1000	4.6	5.3	(1000, 0.0014)	(4.6, 0.0012)	(5.3, 0.0012)	0	(5, 0.0015)
	C-100	1000	1.1	2.1	(1000, 0.0020)	(1.1, 0.0020)	(2.1, 0.0020)	0	(5, 0)
	SVHN	1	1.1	1.1	(1, 0.0010)	(1.1, 0.0006)	(1.1, 0.0006)	0.001	(5, 0.0015)
ResNet	C-10	1000	5.4	5.3	(1000, 0.0014)	(5.4, 0.0012)	(5.3, 0.0012)	0.0005	(2, 0)
	C-100	1000	1	1	(1000, 0.0020)	(9.1, 0.0024)	(12.7, 0.0024)	0.0005	(1, 0)
	SVHN	1000	1.7	1	(1000, 0.0004)	(1.7, 0.0002)	(1.0, 0.0004)	0	(5, 0)

In Figure 2.3, we plot on the left-hand side column the effect of the temperature parameter in the performance for the BLACK-BOX setup. We set the noise magnitude to zero and measured the TNR at TPR-95$\%$ for 500 different temperature values found by a Bayesian optimization for various DNN models. The performance is evaluated on the iSUN dataset. The right-hand side column of Figure 2.3 shows the effect of the noise magnitude parameter in the performance of IGEOOD score in the GREY-BOX setup. We set the temperature to the best found in the BLACK-BOX case. Then, we measured the OOD performance for 21 values of noise magnitude ε equally spaced in the interval $[0, 0.004]$. The best couple (T, ε) for each method and model is used to evaluate the GREY-BOX performances. The best hyperparameters found are detailed in Table 2.3.

We observed that low values of temperature and moderate noise magnitude yield better detection performance for IGEOOD on the logits. For most models and datasets, we obtained better results for temperatures between 1 and 6 and noise magnitudes below 0.002. Detailed results and the best hyperparameters found for each configuration, as well as figures of their impact on performance, are delegated to the appendix of this chapter.

How the choice of validation dataset impacts performance. To verify the consistency of IGEOOD and other methods to the choice of validation data, we measured the TNR at TPR-95$\%$ after tuning our method in a BLACK-BOX and GREY-BOX scenario on nine validation datasets. In Table 2.4, the first column shows the validation dataset, while we used the remaining OOD datasets to evaluate performance. We obtained consistent results, ranging from 63.4% to 72.0% the average TNR at TPR-95$\%$ in the BLACK-BOX case and from 65.0% to 73.4% in the GREY-BOX setting. We show that input pre-processing provides mild amelioration for our method and can be considered a



(a) DenseNet on CIFAR-10.

(b) DenseNet on CIFAR-100.

(c) DenseNet on SVHN.

Figure 2.3: OOD detection performance against temperature and noise magnitude parameters for ODIN (Liang et al., 2018b), Energy (Liu et al., 2020) and IGEOOD (ours) on the iSUN (Xu et al., 2015) OOD dataset for a DenseNet-100 architecture.

fine-tuning step.

We show that the average TNR at TPR-95% for IGEOOD ranges between 63% and 72% on a BLACK-BOX scenario and between 65% and 74% on a GREY-BOX scenario. The performances among the compared methods are consistent across validation datasets.

2.5.4 Results for the WHITE-BOX Setting

For benchmarking IGEOOD on the WHITE-BOX setting, we compare results to the Mahalanobis (Lee et al., 2018b) method with input pre-processing and feature ensemble. For both of them, we extract features from every output of the dense (or residual) block of the DenseNet (or ResNet) model and the first convolutional layer. The size of each feature is reduced by average pooling in the spatial dimensions. Thus, the initial dimension $\mathcal{F}_\ell \times \mathcal{W}_\ell \times \mathcal{H}_\ell$ is reduced to $\mathcal{F}_\ell$, where $\mathcal{F}_\ell$ is the number of channels in block ℓ. For DenseNet, this reduction translates to features of sizes $\boldsymbol{\mathcal{F}}_1 = \{24, 108, 150, 342\}$; and for ResNet, to features of sizes $\boldsymbol{\mathcal{F}}_2 = \{64, 64, 128, 256, 512\}$.

We consider two scenarios for tuning hyperparameters for both Mahalanobis and IGEOOD: one

Table 2.4: Black-Box and Grey-Box settings average performance across different OOD datasets for validation. The hyperparameters are tuned using one validation dataset (column 1), and evaluation is done on the remaining eight OOD test datasets. The DNN is DenseNet-BC-100 pre-trained on CIFAR-10, and the values are TNR at TPR-95% in percentage.

	Black-Box				Grey-Box		
Validation set	Baseline	ODIN	Energy	Igeood	ODIN	Energy	Igeood
iSUN	52.5	64.3	64.9	**65.6**	**66.8**	64.8	65.3
Chars	55.0	70.8	71.1	**71.4**	72.5	72.0	**73.4**
CIFAR-100	55.4	68.6	69.1	**72.0**	68.6	**71.7**	71.3
Gaussian	49.4	62.8	**65.6**	63.4	**70.4**	64.0	68.0
TinyImgNet	53.0	64.7	**65.2**	63.5	**67.0**	65.0	65.5
LSUN	52.1	**63.9**	63.7	63.6	**66.6**	65.3	65.0
Places365	55.3	68.5	69.0	**71.8**	70.0	**71.5**	70.9
SVHN	55.4	68.7	69.3	**69.5**	70.0	69.4	**70.1**
Textures	55.4	71.2	**73.1**	71.4	71.5	**72.4**	71.6
average and std.	53.7±2.0	67.1±3.0	67.9±3.0	**68.0**±3.7	**69.3**±2.0	68.4±3.4	69.0±3.0

with adversarially generated (FGSM) and in-distribution data and another one with 1,000 OOD samples and in-distribution data. We derive two methods: Igeood+, which is given by equation 2.28 and considers that we can calculate the statistics from OOD data as additional information; and Igeood, which doesn't consider any prior on OOD data, i.e., set $\alpha'_\ell = 0$ on equation 2.28.

Comparison with current literature. For each DNN model and in-distribution dataset pair, we report the average and one standard deviation OOD detection performance for Mahalanobis (Lee et al., 2018b), Igeood and Igeood+. Table 2.5 validates the contributions of our techniques. We observe substantial performance improvement in all experiments for the left-hand side of the table, where we outperform Mahalanobis on average for all test cases. Igeood+ show improvements of at least 2.1% up to 23% on TNR at TPR-95%. Since the results are usually above 90%, these improvements are significant. To assess the consistency of Igeood to the choice of validation data, we measured the detection performance when all hyperparameters are tuned only using in-distribution and generated adversarial data, as observed in the right-hand side of Table 2.5. Igeood record improvements up to 10.5%, and improves by 2.5% the average TNR at TPR-95% across all datasets and models.

2.6 Discussion

2.6.1 Ablation Study

Igeood has three components, FR_0, FR_ℓ, and FR'_ℓ, that together compose the final metric of equation 2.28. The outputs of the network provide limited OOD detection capacity as observed in Table 2.1. When available, the intermediate features, i.e., FR_ℓ, are a valuable resource for OOD detection. Moreover, when few reliable OOD data are available, calculating FR'_ℓ can further improve

Table 2.5: Average and standard deviation OOD detection performance for the WHITE-BOX settings. The abbreviation TNR-95%, C-10, and C-100 stands for TNR at TPR-95%, CIFAR-10, and CIFAR-100, respectively.

		Validation on OOD data		Validation on adversarial data	
		TNR-95%	AUROC	TNR-95%	AUROC
Model	In-dist.	Mahalanobis / IGEOOD+ (ours)		Mahalanobis / IGEOOD (ours)	
DenseNet	C-10	76.6±31/**92.6**±14	92.1±12/**98.4**±3.0	75.9±30/**77.9**±29	91.7±12/**94.0**±9.0
	C-100	67.2±28/**90.2**±21	90.2±13/**97.7**±5.0	60.4±34/**70.9**±35	85.3±19/**90.8**±13
	SVHN	93.3±8.0/**98.0**±2.0	98.6±1.0/**99.6**±0.1	**93.7**±10/92.2±9.0	**98.6**±2.0/98.4±1.0
ResNet	C-10	82.5±23/**91.6**±16	96.5±4.0/**98.4**±3.0	**78.6**±24/77.3±32	**95.3**±6.0/90.0±15
	C-100	70.4±30/**86.4**±23	91.9±10/**97.1**±5.0	57.4±36/**65.1**±33	86.9±13/**88.6**±15
	SVHN	96.8±6.0/**98.9**±2.0	99.2±1.0/**99.7**±0.1	**96.3**±8.0/93.6±14	**99.1**±1.0/98.4±3.0
Average and Std.		81.1±11/**92.9**±4.0	94.8±4.0/**98.5**±1.0	77.0±15/**79.5**±10	92.8±5.4/**93.4**±3.9

the detection performance (left-hand side column of Table 2.5). Also, data from a source other than in-distribution, e.g., adversarial samples, is enough for tuning hyperparameters and combining features (right-hand side column of Table 2.5).

For both Mahalanobis and IGEOOD methods, we fitted a logistic regression model with cross-validation using 1,000 OOD and 1,000 in-distribution data samples. Each regression parameter multiplies the layer scores outputs with the objective function of maximizing the TNR at TPR-95%. We set the maximum number of iterations to 100.

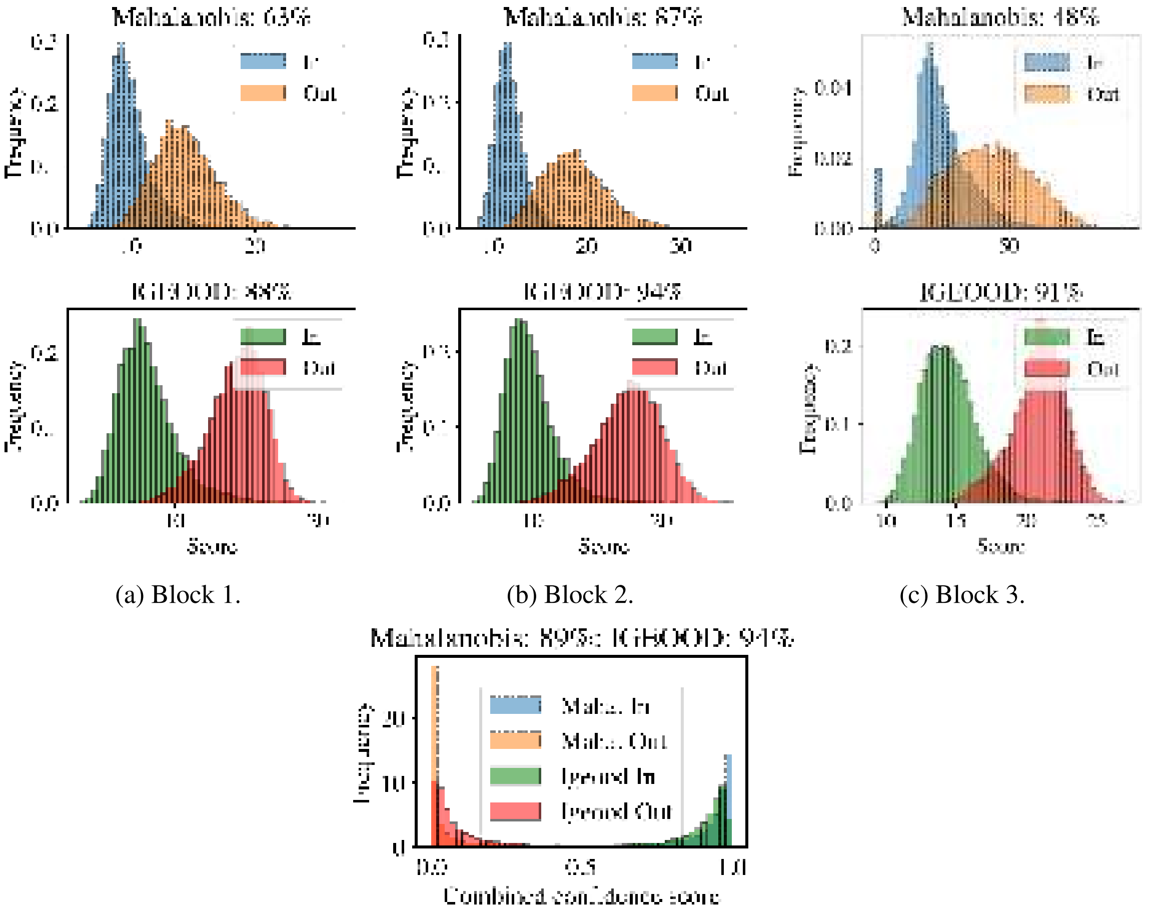
To investigate which hidden feature assists the most in OOD detection, we calculate the TNR at TPR-95% for the scores in the outputs of Blocks 1, 2, and 3 of a DenseNet pre-trained on CIFAR-10. We took as OOD data the SVHN dataset. Figure 2.4 shows the histogram and detection performance for each layer and the results from the logistic regression. Note that we did not consider the logits for the IGEOOD score in this study.

2.6.2 Adversarial Data Generation

We generate adversarial samples from the in-distribution dataset using the fast gradient sign method (FGSM). This method works by exploiting the gradients of the neural network to create a non-targeted adversarial attack. For an input image $\boldsymbol{x}_i$, the method computes the sign of the gradients of the loss function J concerning the input image to create a new image $\boldsymbol{x}_i^{\mathrm{adv}}$ that maximizes the loss as given by equation 2.29. This fabricated image is called an adversarial image, which we use for tuning the hyperparameters of the OOD detection methods in the WHITE-BOX case. Mathematically,

$$\boldsymbol{x}_i^{\mathrm{adv}} = \boldsymbol{x}_i + \varepsilon^{\mathrm{adv}} \odot \operatorname{sign}(\nabla_{\boldsymbol{x}_i} J(\boldsymbol{\theta}, \boldsymbol{x}_i, y_i)), \tag{2.29}$$

where $\varepsilon^{\mathrm{adv}} > 0$ is the additive noise magnitude parameter. Table 2.6 shows the resulting L_∞ mean perturbation and classification accuracy on adversarial samples.



(a) Block 1.

(b) Block 2.

(c) Block 3.

(d) Logistic regression result.

Figure 2.4: Histograms of the Mahalanobis and IGEOOD scores for the output of each hidden block of a DenseNet model for CIFAR-10 (in-dstribution) and SVHN (out-of-distribution). The title shows the TNR at TPR-95% considering only the scores of the outputs of the given layer. The logistic regression found as coefficients: $\boldsymbol{\alpha} = (1.0, -3.6, -0.13)$ for Mahalanobis and $\boldsymbol{\alpha} = (1.0, 1.3, 1.2)$ for IGEOOD.

2.7 Final Remarks and Summary

This chapter introduces IGEOOD, an effective and flexible method for OOD detection that applies to any pre-trained neural network. The main feature of IGEOOD relies on the geodesic distance of the probabilistic manifold of the learned latent representations that induces an effective measure for OOD detection. First, in a (GREY-) BLACK-BOX setup, we calculate the sum of the Fisher-Rao distance between the softmax output, corresponding to the test (pre-processed) sample, and a reference probability, corresponding to the conditional-class of softmax probabilities. Similarly, in a WHITE-BOX setup, we model the low-level features of a DNN as a diagonal Gaussian mixture. The Fisher-Rao distance between the pdf of the latent feature, corresponding to the test sample, and a reference pdf, corresponding to the conditional class of pdfs, provides an effective confidence score. We considered diverse testing environments where prior knowledge of OOD data may be unavailable, reflecting diverse application scenarios. It is observed that IGEOOD significantly and consistently

Table 2.6: The L_∞ mean perturbation used to generate adversarial data with FGSM algorithm and classification accuracy on adversarial samples for the DNN models and in-distribution datasets.

	CIFAR-10		CIFAR-100		SVHN	
	L_∞	Acc.	L_∞	Acc.	L_∞	Acc.
DenseNet-BC-100	0.21	19.5%	0.20	4.45%	0.32	54.7%
ResNet-34	0.21	23.7%	0.20	12.49%	0.25	50.0%

improves the accuracy of OOD detection on several DNN architectures across various datasets for a WHITE-BOX setting.

CHAPTER 3

Neural Trajectories for Out-of-Distribution Detection

3.1 Introduction

Distinguishing OOD samples is challenging. Some previous works developed detectors by combining scores at the various layers of the multi-layer pre-trained classifier (Sastry and Oore, 2020; Lee et al., 2018b; Dadalto et al., 2022; Huang et al., 2021; Colombo et al., 2022). These detectors require either a held-out OOD dataset (e.g., adversarially generated data) or ad-hoc methods to combine OOD scores computed on each layer embedding. A key observation is that existing aggregation techniques overlook the sequential nature of the underlying problem and, thus, limit the discriminative power of those methods. Indeed, an input sample passes consecutively through each layer and generates a highly correlated signature that can be statistically characterized. Our observations in this work motivate the statement:

> *The input's trajectory through a network is valuable for distinguishing typical samples from atypical ones.*

In this chapter, we cast the multi-layer scores into a sequential representation that captures the statistical trajectory of an input sample through the various layers of a neural network. Consequently, we redefine OOD detection as detecting samples whose trajectories are abnormal (or atypical) compared to reference trajectories characterized by the training set. Through a vast experimental benchmark, we showed that the vectorial representation of a sample encodes valuable information for OOD detection without the need for outlier data to tune parameters.

The contents of this chapter will be based on the work Dadalto et al. (2023b) that was conducted with my co-authors Pierre Colombo, Guillaume Staerman, Nathan Noiry, and Pablo Piantanida. The code is available at the url[1].

[1] `https://www.github.com/edadaltocg/detectors`

3.2 Summary of Contributions

1. *Computing OOD scores from trajectories.* We propose a semantically informed map from multiple embedding spaces to a simple vectorial representation. Subsequently, the simple dot product between the test sample's trajectory and the training prototype trajectory indicates how likely a sample is to belong to in-distribution.

2. *Extensive empirical evaluation.* We validate the value of the proposed method by demonstrating gains against twelve strong state-of-the-art methods on both CIFAR-10 and ImageNet on average TNR at 95% TPR and AUROC across five NN architectures.

3.3 Related Works

Efforts toward combining multiple features to improve performance were previously explored in (Lee et al., 2018b; Sastry and Oore, 2020; Dadalto et al., 2022). The strategy relies upon having additional data for tuning the detector or focusing on specific model architectures, which are limiting factors in real-world applications. For instance, MOOD (Lin et al., 2021) relies on the MSDNet architecture, which trains multiple classifiers on the output of each layer in the feature extractor, and their objective is to select the most appropriate layer in inference time to reduce the computation cost. On the other hand, we study the trajectory of an input through the network. Unlike MOOD, our method applies to any current architecture of NNs.

3.4 Preliminaries

We start by recalling the general setting of the OOD detection problem from a mathematical point of view (Section 3.4.1). Then, in Section 3.4.2, we motivate our method through a simple yet clarifying example showcasing the limitations of previous works and how we approach the problem.

3.4.1 Background

Previous work rely on a multi-layer pre-trained classifier $f : \mathcal{X} \to \mathcal{Y}$ defined as:

$$f_\theta(\cdot) = h \circ f_L \circ f_{L-1} \circ \cdots \circ f_1(\cdot),$$

with $L \geq 1$ layers, where $f_\ell : \mathbb{R}^{d_{\ell-1}} \to \mathbb{R}^{d_\ell}$ is the ℓ-th layer of the multi-layer neural classifier, d_ℓ denotes the dimension of the latent space induced by the ℓ-th layer ($d_0 = d$), and h indicates the classifier that outputs the logits. We also define $\boldsymbol{z}_\ell = (f_\ell \circ \cdots \circ f_1)(\boldsymbol{x})$ as the latent vectorial representation at the $\ell-$th layer for an input sample $\boldsymbol{x}$. We will refer to the logits as $\boldsymbol{z}_{L+1}$ and h as f_{L+1} to homogenize notation. It is worth emphasizing that the trajectory of $(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{L+1})$ corresponding to a test input $\boldsymbol{x}_0$ are dependent random variables whose joint distribution strongly depends on the underlying distribution of the input.

Therefore, the design of the binary detection function $d(\cdot)$ is typically based on the three key steps:

(i) A similarity measure $\mathrm{D}(\cdot\,;\cdot)$ (e.g., Cosine similarity, Mahalanobis distance, etc.) between a sample and a population is applied at each layer to measure the similarity (or dissimilarity) of a test input $\boldsymbol{x}_0$ at the ℓ-th layer $\boldsymbol{z}_{\ell,0} = (f_\ell \circ \cdots \circ f_1)(\boldsymbol{x}_0)$ w.r.t. the population of the training examples observed at the same layer $\big\{\boldsymbol{z}_\ell = (f_\ell \circ \cdots \circ f_1)(\boldsymbol{x}) \,:\, \boldsymbol{x} \in \mathcal{S}_N\big\}$.

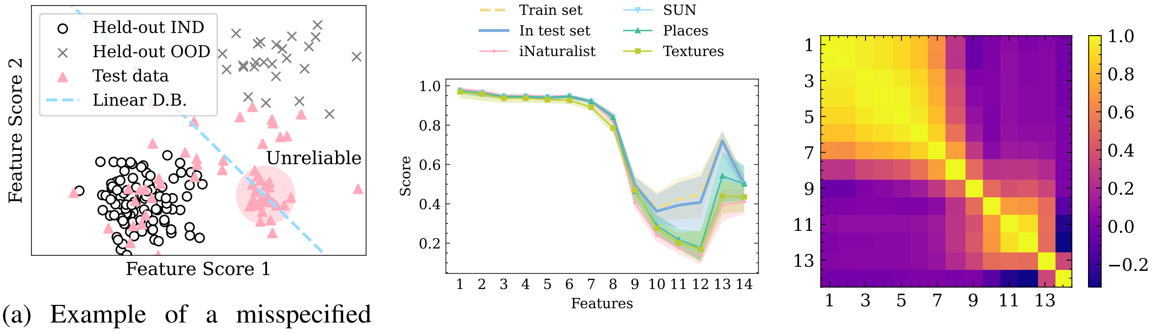
(ii) The layer-wise score obtained is mapped to the real line collecting the OOD scores.

(iii) A threshold is set to build the final decision function.

A fundamental ingredient remains in step (ii):

> *How to consistently leverage the information collected from multiple layers outputs in an unsupervised way, i.e., without resorting to OOD or pseudo-OOD samples?*

3.4.2 From Supervised Multi-Layer Scores to an Unsupervised Formulation



(a) Example of a misspecified model in a toy example in 2D caused by fitting with held-out OOD dataset.

(b) Trajectory of data through a network with 25% and 75% percentile bounds.

(c) Correlation between layers training scores in a network, highlighting structure in the trajectories.

Figure 3.1: Figure 3.1a summarizes the limitation of supervised methods for aggregating layer scores that rely on held-out OOD or pseudo-OOD data. It biases the decision boundary (D.B) that does not generalize well to other types of OOD data. We observed that in-distribution and OOD data have disparate trajectories through a network (Fig. 3.1b), especially on the last five features. These features are correlated in a sequential fashion, as observed in Fig. 3.1c.

Previous multi-feature OOD detection works treat step (ii) as a supervised learning problem (Lee et al., 2018b; Dadalto et al., 2022) for which the solution is a linear binary classifier. The objective is to find a linear combination of the scores obtained at each layer that will sufficiently separate in-distribution from OOD samples. A held-out OOD dataset is collected from true (or pseudo-generated) OOD samples. The linear soft novelty score functions s_α writes:

$$s_\alpha(\boldsymbol{x}_0) = \sum_{\ell=1}^{L} \alpha_\ell \cdot \mathrm{D}\left(\boldsymbol{x}_0; \big\{(f_\ell \circ \cdots \circ f_1)(\boldsymbol{x}) \,:\, \boldsymbol{x} \in \mathcal{S}_N\big\}\right).$$

The shortcomings of this method are the need for extra data or ad-hoc parameters, which results in decision boundaries that underfit the problem and fail to capture certain types of OOD samples. To illustrate this phenomenon, we designed a toy example (see Figure 3.1a) where scores are extracted from two features fitting a linear discriminator on held-out in-distribution (IND) and OOD samples.

As a consequence, areas of unreliable predictions where OOD samples cannot be detected due to the misspecification of the linear model arise. One could simply introduce a non-linear discriminator that better captures the geometry of the data for this 2D toy example. However, it becomes challenging as we move to higher dimensions with limited data.

By reformulating the problem from a vectorial data point of view, we can identify trends and typicality in trajectories extracted by the network from the input. Figure 3.1b shows the dispersion of trajectories coming from the in-distribution and OOD samples. These patterns are extracted from multiple latent representations and aligned on a time-series-like object. *We observed that trajectories coming from OOD samples exhibit a different shape when compared to typical trajectories from training data.* Thus, to determine if an instance belongs to in-distribution, we can test if the observed path is similar to the reference extracted from the training set.

3.5 Trajectory-Based OOD Detection

This section presents our OOD detection framework, which applies to any pre-trained multi-layer neural network with no requirements for OOD samples. We describe our method through two key steps: vector representation of the input sample (see Section 3.5.1) and test time OOD score computation (see Section Section 3.5.2).

3.5.1 Vectorial Representation

The first step to obtaining a vector representation of the data from the multivariate hidden representations is to reduce each feature map to a scalar value. To do so, we first compute the class-conditional training population prototypes defined by:

$$\boldsymbol{\mu}_{\ell,y} = \frac{1}{N_y} \sum_{i=1}^{N_y} \boldsymbol{z}_{\ell,i}, \tag{3.1}$$

where $N_y = \left|\{\boldsymbol{z}_{\ell,i} : y_i = y, \forall i \in \{1..N\}\}\right|$, $1 \leq \ell \leq L+1$ and $\boldsymbol{z}_{\ell,i} = (f_\ell \circ \cdots \circ f_1)(\boldsymbol{x}_i)$.

Given an input example, we compute the *probability weighted scalar projection*[2] between its features (including the logits) and the training class conditional prototypes, resulting in $L+1$ scalar scores:

$$\begin{aligned} \mathrm{D}_\ell(\boldsymbol{x}; \mathcal{M}_\ell) &= \sum_{y=1}^{C} \sigma_y(\boldsymbol{x}) \cdot \mathrm{proj}_{\boldsymbol{\mu}_{\ell,y}} \boldsymbol{z}_\ell & (3.2) \\ &= \sum_{y=1}^{C} \sigma_y(\boldsymbol{x}) \|\boldsymbol{z}_\ell\| \cos\big(\angle\left(\boldsymbol{z}_\ell, \boldsymbol{\mu}_{\ell,y}\right)\big), & (3.3) \end{aligned}$$

where $\mathcal{M}_\ell = \{\boldsymbol{\mu}_{\ell,y} : y \in \mathcal{Y}\}$, $\|\cdot\|$ is the ℓ_2-norm, $\angle(\cdot,\cdot)$ is the angle between two vectors, and

[2] Other metrics to measure the similarity of an input w.r.t. the population of examples can also be used.

$\sigma_y(\boldsymbol{x}; f_\theta)$ is the softmax function on the logits $f_\theta(\boldsymbol{x})$ of class y. Hence, our layer-wise scores rely on the notions of vector length and angle between vectors, which can be generalized to any n-dimensional inner product space without imposing any geometrical constraints.

It is worth emphasizing that our layer score has some advantages compared to the class conditional Gaussian model first introduced in Lee et al. (2018b) and the Gram matrix-based method introduced in Sastry and Oore (2020). Our layer score encompasses a broader class of distributions as we do not suppose a specific underlying probability distribution. We avoid computing covariance matrices, which are often ill-conditioned for latent representations of DNNs Ahuja et al. (2019). Since we do not store covariance matrices, our vectorial approach has a negligible overhead regarding memory requirements. Also, our method can be applied to any vector-based hidden representation, not being restricted to matrix-based representations as in Sastry and Oore (2020). Thus, our approach applies to a broader range of models, including transformers.

By computing the scalar projection at each layer, we define the following *vectorial neural-representation* extraction function given by Eq. 3.4. Thus, we can map sample representations to a simpler vector space while retaining information on the typicality w.r.t the training dataset.

$$\begin{aligned} \phi : \mathcal{X} &\to \mathbb{R}^{L+1} \\ \boldsymbol{x} &\mapsto \left[\mathrm{D}_1\left(\boldsymbol{x}; \mathcal{M}_1\right), \ldots, \mathrm{D}_{L+1}\left(\boldsymbol{x}; \mathcal{M}_{L+1}\right)\right] \end{aligned} \tag{3.4}$$

We apply ϕ to the training input $\boldsymbol{x}_i$ to obtain the representation of the training sample across the network $\boldsymbol{u}_i = \phi(\boldsymbol{x}_i)$. We consider the related vectors $\boldsymbol{u}_i, \forall\, i \in [1 : N]$[3] as curves parameterized by the layers of the network. We build a training reference dataset $\mathcal{U} = \{\boldsymbol{u}_i\}_{i=1}^N$ from these representations that will be useful for detecting OOD samples during test time. We then rescale the training set trajectories w.r.t the maximum value found at each coordinate to obtain layer-wise scores on the same scaling for each coordinate. Hence, for $j \in \{1, \ldots, L+1\}$, let $\max\left(\mathcal{U}\right) := [\max_i \boldsymbol{u}_{i,1}, \ldots, \max_i \boldsymbol{u}_{i,L+1}]^\top$, we can compute a reference trajectory $\bar{\boldsymbol{u}}$ for the entire training dataset defined in equation 3.5 that will serve as a global *typical reference* to test trajectories.

$$\bar{\boldsymbol{u}} = \frac{1}{N} \sum_{i=1}^{N} \frac{\boldsymbol{u}_i}{\max(\mathcal{U})} \tag{3.5}$$

3.5.2 Computing the OOD Score at Test Time

At inference time, we first re-scale the test sample's trajectory as we did with the training reference $\bar{\phi}(\boldsymbol{x}) = \phi(\boldsymbol{x}) / \max\left(\mathcal{U}\right)$. Then, we compute a similarity score w.r.t this typical reference, resulting in our OOD score. We choose as metric also the scalar projection of the test vector to the training reference. In practical terms, it boils down to the *inner product* between the test sample's trajectory and the training set's typical reference trajectory since the norm of the average trajectory is constant

[3] We observed empirically that subsampling vectors to even $N/100$ yields very good results.

for all test samples. Mathematically, our scoring function $s : \mathcal{X} \mapsto \mathbb{R}$ writes:

$$s(\boldsymbol{x}; \bar{\boldsymbol{u}}) = \langle \bar{\phi}(\boldsymbol{x}), \bar{\boldsymbol{u}} \rangle = \sum_{j=1}^{L+1} \bar{\phi}(\boldsymbol{x})_j \bar{\boldsymbol{u}}_j \tag{3.6}$$

which is bounded by Cauchy-Schwartz's inequality.

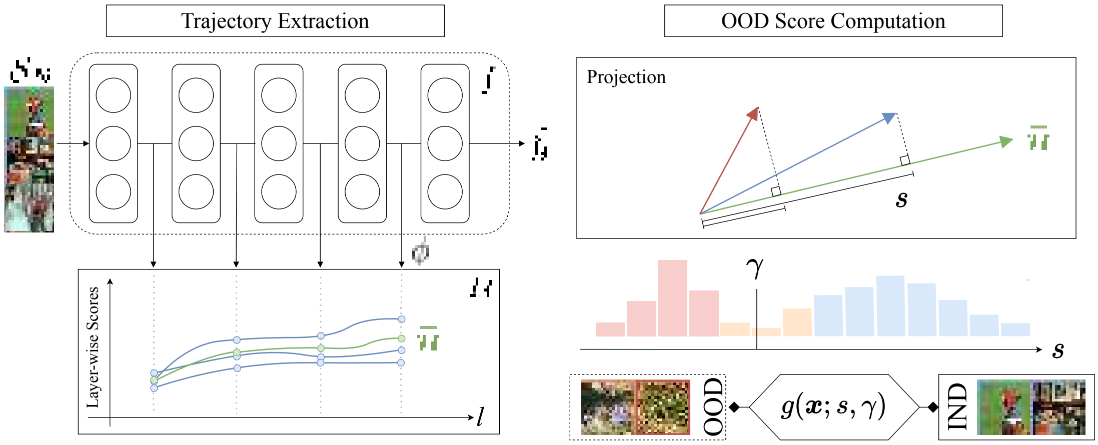
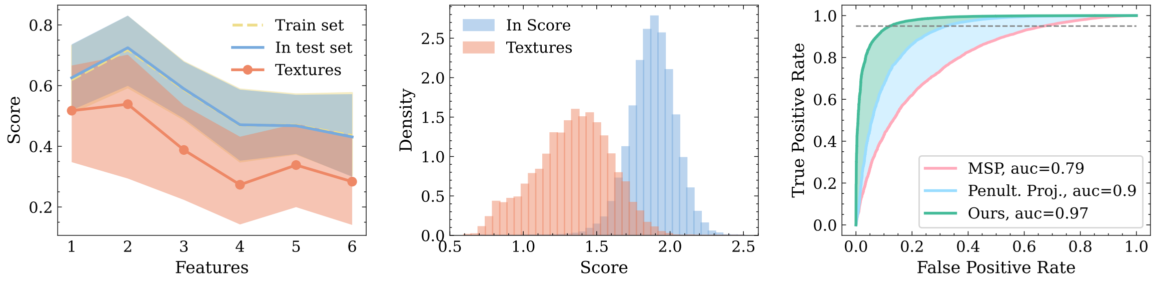


Figure 3.2: The left-hand side of the figure shows the feature extraction process of a deep neural network classifier f. The mapping of the hidden representations of an input sample into a vectorial representation is given by a function ϕ. The right-hand side of the figure shows how our method computes the OOD score s of a sample during test time. The sample's trajectory is projected to the training reference trajectory $\bar{u}$. Finally, a threshold γ is set to obtain a discriminator g.

3.6 Experimental Setup

Datasets. We set as *in-distribution* dataset *ImageNet-1K* (= ILSVRC2012; Deng et al., 2009) for our main experiments. For the *out-of-distribution* datasets, we take the same dataset splits introduced by Huang and Li (2021).



(a) Vectorial trajectories.

(b) OOD score histogram.

(c) ROC curve.

Figure 3.3: Vectorial representation with 5 and 95% quantiles (3.3a), histogram (3.3b), and ROC curve (3.3c) for our OOD score on a DenseNet-121 model with Textures as OOD dataset.

Models. We ran experiments with five models. A *DenseNet-121* (Huang et al., 2017) pre-trained on ILSVRC-2012 with 8M parameters and test set top-1 accuracy of 74.43%. A *ResNet-50* model with top-1 test set accuracy of 75.85% and 25M parameters. A *BiT-S-101* (Kolesnikov et al., 2020) model based on a ResNetv2-101 architecture with top-1 test set accuracy of 77.41% and 44M parameters. And a MobileNetV3 large, with an accuracy of 74.6% and around 5M parameters. We reduced the intermediate representations with an *max pooling* operation when needed obtaining a final vector with a dimension equal to the number of channels of each output. We also ran experiments with a Vision Transformer (*ViT-B-16*; Dosovitskiy et al., 2021), which is trained on the ILSVRC2012 dataset with 82.64% top-1 test accuracy and 70M parameters. We take the output's class tokens for each layer. We download all the checkpoint weights from PyTorch (Paszke et al., 2019) hub. All models are trained from scratch on ImageNet-1K. For all models, we compute the probability-weighted projection of the building blocks, as well as the projection of the logits of the network to form the vectorial representation. So, there is no need for a special layer selection.

		iNat	SUN	Places	Textures	IN-O	OI-O	NINCO	SSB-H	Avg.
ResNet-50	MSP	88.4	81.6	80.5	80.4	28.6	83.9	79.9	75.6	74.9
	ODIN	91.3	84.7	82.0	84.9	40.7	87.4	80.4	73.0	78.0
	Energy	90.6	86.6	84.0	86.7	41.8	87.1	79.7	72.2	78.6
	MaxLogits	91.1	86.4	84.0	86.4	40.7	87.4	80.4	73.0	78.7
	KLMatching	89.7	80.4	78.9	82.5	39.0	85.7	80.8	74.0	76.4
	IGEOOD	90.1	85.0	82.8	85.7	39.5	86.7	80.2	71.5	77.7
	Mahalanobis	63.0	50.8	50.4	89.8	78.5	75.1	65.8	55.2	66.1
	GradNorm	93.9	90.1	86.1	90.6	47.9	80.9	74.1	65.1	78.6
	DICE	94.3	90.7	87.4	90.6	43.3	85.8	77.5	70.3	80.0
	ViM	87.4	81.0	78.3	96.8	71.0	88.8	78.9	66.4	81.1
	ReAct	**96.7**	94.3	**91.9**	88.8	52.5	**89.2**	80.2	75.0	83.6
	KNN	94.9	88.6	84.7	95.4	**76.8**	84.1	79.6	64.2	83.5
	Proj. (Ours)	95.8	**94.5**	91.2	**96.3**	62.9	82.7	**81.2**	**80.1**	**85.6**
ViT-B-16	MSP	88.2	80.9	80.4	83.0	73.6	89.0	83.2	80.5	82.4
	ODIN	86.0	75.2	76.5	81.2	74.5	89.5	83.7	83.3	81.2
	Energy	79.2	70.2	68.4	79.3	75.9	88.5	82.7	84.4	78.6
	MaxLogits	93.2	84.8	81.2	83.7	75.0	89.6	83.9	83.4	84.4
	KLMatching	93.2	85.1	83.4	84.5	78.2	90.1	83.4	78.5	84.6
	IGEOOD	94.6	**85.9**	81.8	85.0	77.2	91.1	85.1	84.4	85.6
	Mahalanobis	96.0	85.3	**84.2**	87.5	83.3	93.1	85.4	76.9	86.5
	GradNorm	91.2	85.3	83.4	86.5	44.7	47.2	51.7	68.5	69.8
	DICE	46.1	49.3	49.7	64.3	40.1	32.5	42.3	52.7	47.1
	ViM	**97.1**	85.4	81.9	85.9	83.1	93.0	83.5	75.0	85.6
	ReAct	85.6	78.8	77.3	84.5	83.2	93.8	85.4	81.2	83.7
	KNN	88.9	79.4	77.7	87.8	**84.4**	**94.2**	87.2	79.2	84.8
	Proj. (Ours)	93.3	82.1	80.7	**91.1**	81.4	93.8	**87.5**	**84.8**	**86.8**

Table 3.1: Comparison against post-hoc state-of-the-art methods for OOD detection on the ImageNet benchmark in terms of AUROC. Values are in percentage, and higher is better.

3.7 Results and Discussion

3.7.1 Main Results

We report our main results in Table 3.1, which includes the performance for two out of five models (see Section 3.A for the remaining three), four OOD datasets, and twelve detection methods. On ResNet-50, we achieve a gain of 2% in average AUROC compared to ReAct. For the ViT-B-16, the gap between methods is small, and our method exhibits a comparable TNR and AUROC to previous state-of-the-art. For BiT-S-101, we outperform GradNorm by 18.9% TNR and 5.4% AUROC. For DenseNet-121 (see Section 3.A), we improved on ReAct by 16% and 3.9% in TNR and AUROC, respectively. Finally, on MobileNet-V3 Large, we registered gains of around 20% TNR and 9.2% AUROC. We observed that activation clipping benefits our method on convolution-based networks but hurts its performance on transformer architectures, aligned with the results from Sun et al. (2021).

	Trajectory	iNat	SUN	Places	Text.	Avg.
ResNet-50	Penult.	96.3	93.1	89.5	95.2	93.5
ResNet-50	Last 2	98.0	96.7	94.9	94.5	**96.0**
ResNet-50	Last 3	97.7	95.3	92.7	96.5	95.5
ResNet-50	All	95.8	94.5	91.2	96.3	94.5
ViT-B-16	Penult.	97.0	88.6	86.2	91.0	90.7
ViT-B-16	Last 2	96.5	88.8	84.9	90.7	90.2
ViT-B-16	Last 3	97.6	89.9	86.7	91.2	**91.3**
ViT-B-16	All	93.3	82.1	80.7	91.1	86.8

Table 3.2: Performance for segments of the trajectory.

3.7.2 Results on CIFAR-10

We ran experiments with a ResNet-18 model trained on CIFAR-10 Krizhevsky et al. (2009). We extracted the trajectory from the outputs of layers 2 to 4 and logits. The results are displayed in Table 3.3. Our method outperforms comparable state-of-the-art methods by 2.4% on average AUROC, demonstrating that it is consistent and suitable for OOD detection on small datasets too.

	MSP	ODIN	Ener.	KNN	ReAct	Ours
C-100	88.0	88.8	89.1	89.8	89.7	89.4
SVHN	91.5	91.9	92.0	94.9	94.6	99.0
LSUNc	95.1	98.5	98.9	97.0	97.9	99.8
LSUNr	92.2	94.9	95.3	95.8	96.7	99.8
TIN	89.8	91.1	91.7	92.8	93.8	98.0
Places	90.1	92.9	93.2	93.7	94.7	93.6
Text.	88.5	86.4	87.2	94.2	93.4	97.9
Avg.	90.7	92.1	92.5	94.0	94.4	**96.8**

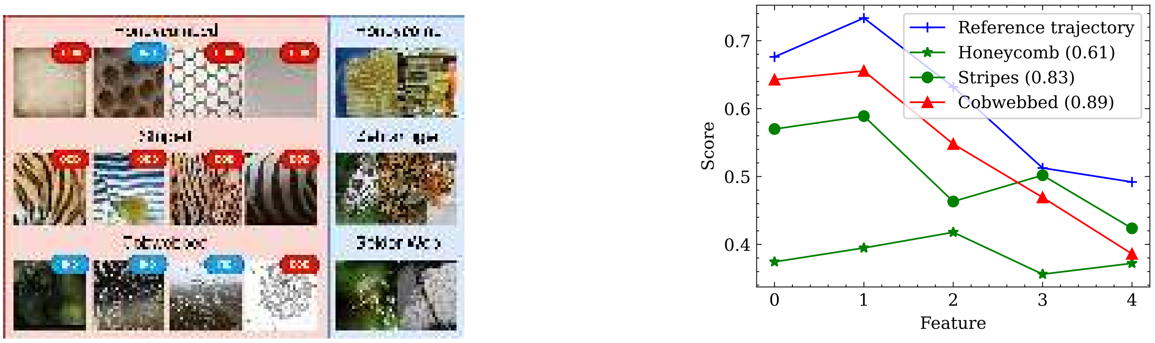
Table 3.3: CIFAR-10 benchmark results in terms of AUROC based on a ResNet-18 model.

3.7.3 Ablation Study

Table 3.2 show OOD detection results in terms of AUROC (%) on the ImageNet benchmark for the penultimate layer, for a trajectory formed by the last two and three outputs compared to the results for the entire trajectory. We demonstrated that building a trajectory with the last layers could improve detection performance, as observed in previous work that most information for OOD detection in Computer Vision benchmarks is contained in the last layers' outputs. Thus, we showed that the practitioner's inductive bias can improve detection, demonstrating that our method is flexible and can adopt smart layer selection strategies.

3.7.4 Study Case

There are a few overlaps regarding the semantics of class names in the Textures and ImageNet datasets. In particular, “honeycombed" in Textures versus “honeycomb" in ImageNet, “stripes" vs. “zebra", "tiger", and "tiger cat", and “cobwebbed" vs. “spider web". We showed in Table 3.1 that our method significantly decreases the number of false negatives in this benchmark. We designed a simple study case to understand better how our method can discriminate where baselines often fail. Take the Honeycombed vs. Honeycomb, for instance (the first row of Fig. 3.4a). The honeycomb from ImageNet references natural honeycombs, usually found in nature, while honeycombed in Textures has a broader definition attached to artificial patterns. In this class, the Energy baseline makes 108 mistakes, while we only make 20 mistakes. We noticed that some of our mistakes are aligned with real examples of honeycombs (e.g., the second example from the first row). At the same time, we confidently classify other patterns correctly as OOD. For the striped case (middle row), our method flags only 16 examples as being in-distribution, but we noticed an average higher score for the trajectories in Fig. 3.4b (Stripes). Note that, for the animal classes, the context and head are essential features for classifying them. For the Spider webs class, most examples from Textures are visually closer to ImageNet. Overall, the study shows that our scores are aligned with the semantic proximity between testing samples and the training set.



(a) A few examples from Textures dataset sharing semantic overlap with ImageNet classes.

(b) Trajectories of the leftmost examples of Fig. 3.4a and their OOD scores in parenthesis.

Figure 3.4: Detection of individual samples on classes with semantic overlap between ImageNet and Textures. The badge on each image in Fig. 3.4a shows the detection label given by our method.

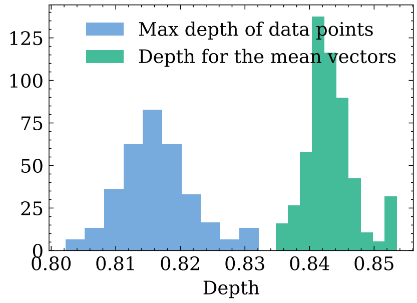


Figure 3.5: Histogram showing that the halfspace depth of the average vectors for a given class is higher than the highest depth of an embedding feature vector of the same class, demonstrating multivariate centrality.

3.7.5 On the Centrality of the Class-Conditional Features Maps

This section studies whether the mean vectors are sufficiently informative for statistically modeling the class-conditional embedding features. From a statistical point of view, the average would be informative if the data is compact. To address this point, we plotted the median and the mean for the coordinates of the feature map and measured their difference in Figure 3.6. We observed that they almost superpose in most dimensions or are separated by a minor difference, which indicates that the data is compact and central. In addition, we showed in Figure 3.5 that the halfspace depth (Tukey, 1975a) of the mean vector of a given is superior to the maximum depth of a training sample vector of the same class, suggesting the average is central or deep in the feature data distribution. From a practical point of view, the clear advantages of using only the mean as a reference are computational efficiency, simplicity, and interpretability. We believe that future work directions could be exploring a method that better models the density in the embedding features, especially as more accurate classifiers are developed.

3.7.6 Limitations

We believe this work is only the first step towards efficient post-score aggregation as we have tackled an open and challenging problem of combining multi-layer information. We rely on the class-conditional mean vectors, which might not be sufficiently informative for statistically modeling the embedding features depending on the distribution. From a statistical point of view, the average would be informative if the data is compact. To address this point, we plotted the median and the mean for several coordinates of the feature map and measured their difference in Fig. 3.6. We observed that they superpose in most dimensions, which indicates that the data is compact and central, thus, the prototypes are informative. Additionally, we showed in Fig. 3.5 that the halfspace depth (Tukey (1975a); more details in Section 3.7.5) of the mean vector of a given class is superior to the maximum depth of a training sample of the same class, suggesting that the average is deep in the feature's data distribution.

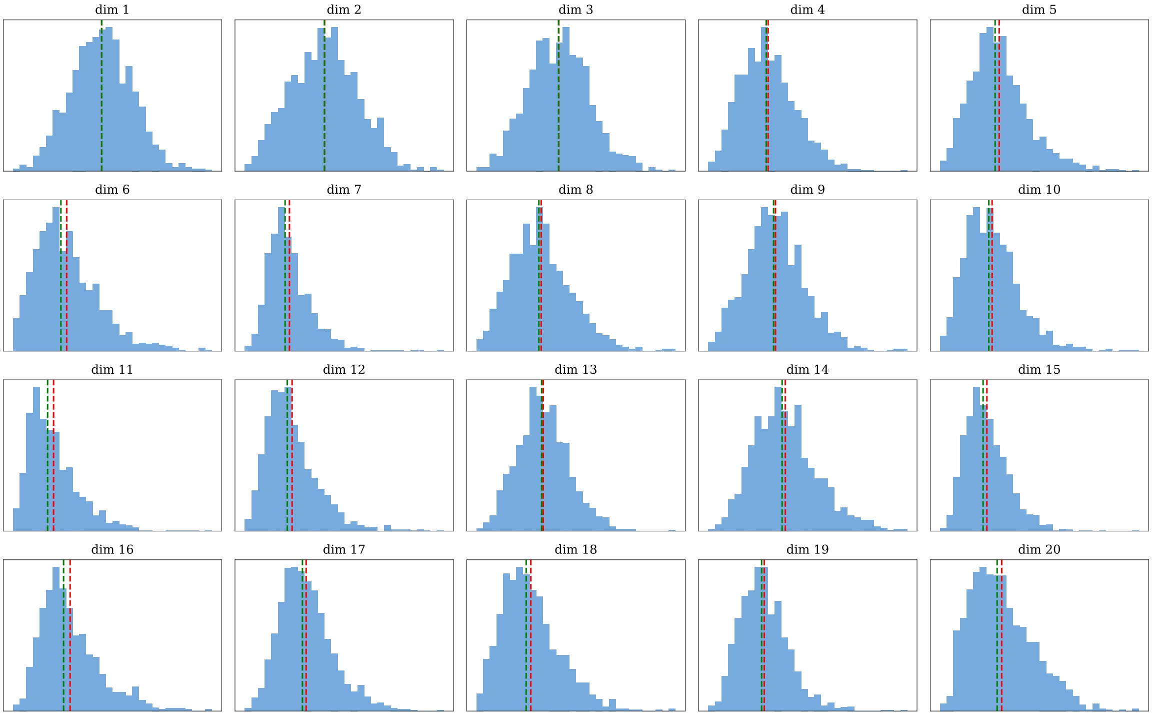


Figure 3.6: Histogram for the 20 first dimensions of the penultimate feature of a DenseNet-121 for class index 0 of ImageNet. The green line is the average, and the red line is the estimated median.

3.8 Final Remarks and Summary

In this chapter, we introduced an original approach to OOD detection that leverages the sample's trajectories through the layers of a neural network. Our method detects samples whose trajectories differ from the typical behavior characterized by the training set. The key ingredient relies on the statistical dependencies of the scores extracted at each layer, using a purely self-supervised algorithm. We empirically validate through an extensive benchmark against several state-of-the-art methods that our Projection method is consistent and achieves great results across multiple models and datasets, even requiring no special hyperparameter tuning. We hope this work will encourage future research to explore sample trajectories to enhance the safety of AI systems.

3.A Appendix to Chapter 3

We provide additional results in terms of TNR and AUROC for the BiT-S-101, DenseNet-121, and MobileNetV3-Large models in Table 3.4 for the baseline methods and ours.

Table 3.4: Extended comparison against post-hoc state-of-the-art methods for OOD detection on the ImageNet benchmark. Values are in percentage.

		iNaturalist		SUN		Places		Textures		Average	
		TNR	ROC	TNR	ROC	TNR	ROC	TNR	ROC	TNR	ROC
BiT-S-101	MSP	35.9	87.9	28.8	81.9	21.3	79.3	21.9	77.5	27.0	81.7
	ODIN	29.3	86.7	36.7	86.8	26.0	82.7	24.1	79.3	29.0	83.9
	Energy	25.0	84.5	39.8	87.3	27.2	82.7	24.2	78.8	29.0	83.3
	MaxLogits	80.8	95.9	35.8	80.2	32.3	76.8	33.9	80.6	45.7	83.4
	KLMatching	68.5	92.9	28.6	81.5	27.4	80.2	38.0	84.2	40.6	84.7
	IGEOOD	**83.8**	**96.4**	37.6	82.6	33.4	79.3	36.8	82.9	47.9	85.3
	Mahalanobis	16.5	78.3	13.2	74.5	10.5	69.6	**86.6**	**97.3**	31.7	79.9
	GradNorm	41.3	86.0	55.2	88.2	39.0	**83.3**	41.2	81.0	44.2	84.6
	ReAct	46.2	88.9	10.7	65.9	7.0	62.0	8.5	65.8	18.1	70.7
	DICE	16.3	86.4	8.2	69.1	4.6	61.9	5.2	62.9	8.6	70.1
	ViM	16.3	86.4	8.2	69.1	4.6	61.9	5.2	62.9	8.6	70.1
	KNN	39.7	88.9	22.1	77.5	20.6	75.9	54.1	89.7	34.1	83.0
	Proj. (Ours)	67.0	91.7	**59.5**	**89.4**	**40.8**	82.3	85.1	96.7	**63.1**	**90.0**
DenseNet-121	MSP	50.7	89.1	33.0	81.5	30.8	81.1	32.9	79.2	36.9	82.7
	ODIN	60.4	92.8	45.2	87.0	40.3	85.1	45.3	85.0	47.8	87.5
	Energy	60.3	92.7	48.0	87.4	42.2	85.2	47.9	85.4	49.6	87.7
	MaxLogits	58.2	92.3	44.6	86.8	38.7	84.5	45.1	84.8	46.6	87.1
	KLMatching	50.0	89.6	22.7	79.8	23.6	78.8	36.1	82.4	33.1	82.7
	IGEOOD	46.7	90.3	37.4	84.8	32.2	82.6	41.6	83.9	39.5	85.4
	Mahalanobis	3.5	59.7	4.8	57.0	4.6	54.8	54.4	88.3	16.8	64.9
	GradNorm	73.3	93.4	59.1	88.8	48.0	84.1	56.7	87.7	59.3	88.5
	ReAct	68.8	93.9	51.3	89.6	44.5	86.6	48.1	87.6	53.2	89.4
	DICE	72.0	93.9	61.0	89.8	48.8	85.8	58.6	87.8	60.1	89.3
	ViM	27.4	85.8	14.3	77.6	11.7	73.8	80.9	96.1	33.6	83.3
	KNN	57.1	92.1	33.2	83.6	26.8	79.6	81.5	**96.5**	49.7	88.0
	Proj. (Ours)	**80.4**	**96.4**	**62.2**	**91.8**	**52.3**	**88.0**	**81.8**	**96.5**	**69.2**	**93.2**
MobileNetV3 Large	MSP	**39.4**	**87.3**	25.0	79.9	23.8	79.0	29.7	80.8	29.5	81.8
	ODIN	39.1	86.9	26.1	78.8	23.4	77.5	31.5	80.6	30.0	80.9
	Energy	30.1	83.9	21.2	76.7	18.1	74.8	28.7	78.9	24.5	78.6
	MaxLogits	39.4	87.0	26.2	78.9	23.5	77.6	31.6	80.7	30.2	81.0
	KLMatching	46.6	89.4	19.8	78.7	20.8	77.7	31.9	83.2	29.7	82.2
	IGEOOD	35.4	86.8	24.8	79.2	21.8	77.6	32.6	81.5	28.7	81.3
	Mahalanobis	7.3	71.3	2.2	52.7	3.3	54.4	45.7	84.7	14.6	65.8
	GradNorm	11.6	76.2	7.9	68.3	9.3	68.7	14.6	76.7	10.9	72.5
	ReAct	15.0	82.0	8.4	70.7	10.2	70.5	38.6	85.1	18.1	77.1
	DICE	23.2	83.5	12.4	76.8	14.0	75.8	13.3	78.0	15.7	78.5
	ViM	8.0	75.1	4.4	61.5	5.4	63.7	34.6	83.1	13.1	70.8
	KNN	31.3	78.8	24.9	77.9	18.5	73.5	12.8	64.5	21.9	73.7
	Proj. (Ours)	34.1	86.3	**49.4**	**87.1**	**37.4**	**82.4**	**80.1**	**95.4**	**50.3**	**87.8**

CHAPTER 4

Combine and Conquer: A Meta-Analysis on Data Distribution Shift and Out-of-Distribution Detection

4.1 Introduction

This chapter presents a universal method to improve the *detection* of performance-degrading shifts by ensembling existing detectors in an unsupervised manner. Each detector can be formalized as a test of equivalence of the source distribution (from which training data is sampled) and target distribution (from which real-world data is sampled) through the lens of a predictive model. Our framework is highly adaptable for future developments in detection scores. It is motivated by the fact that different detection algorithms may make trivial mistakes in different parts of the data space without any assumptions on the test data distribution (Birnbaum, 1954). The challenge is to develop a widely applicable method for combining detectors to alleviate catastrophic errors, resulting in a more effective detector with consolidated decision boundaries.

Additionally, we can create a fully interpretable criterion by adjusting the final statistics of the in-distribution scores. Our framework is highly adaptable for future developments in detection scores. Through a meticulous empirical investigation, we analyze different types of shifts with varying degrees of impact on data, demonstrating that our approach significantly enhances overall robustness and performance across various domains, shift types, and out-of-distribution detection scenarios.

The contents of this chapter will be based on the work Dadalto et al. (2023a) that was a collaboration with my co-authors Florence Alberge, Pierre Duhamel, and Pablo Piantanida. The code is available at the url[1].

4.2 Summary of Contributions

Our method is inspired by *meta-analysis* (Glass, 1976), a statistical technique combining multiple studies' results to produce a single overall estimate. Even though p-value ensembling is not new,

[1] https://www.github.com/edadaltocg/detectors

this is the first time such techniques have been used in OOD detection and data distribution shift detection. We summarize our contributions as follows:

1. To the best of our knowledge, we are the first to present a simple and convenient ensembling algorithm for combining existing out-of-distribution data detectors, leading to better generalizability by incorporating effects that may not be apparent in individual detectors;

2. A probabilistic interpretable detection criterion that comes for free by correcting the final statistics into a distribution with known parameters;

3. A framework to adapt any single example detector to a window-based data shift detector;

4. And a comprehensive empirical investigation encompassing single example out-of-distribution detection and window-based data distribution shift detection.

4.3 Related Works

Window-based data shift detection. This line of work proposes methods for detecting shifts in data distribution using multiple samples. Lipton et al. (2018) presents a technique for detecting prior probability shifts. Rabanser et al. (2019) studies two-sample tests with high dimensional inputs through dimensionality reduction techniques from the input to a projected space. Cobb and Looveren (2022) explores two sample conditional distributional shift detection based on maximum conditional mean discrepancies to segment relevant contexts in which data drift is diminishing. Our work shows detection against window data shifts, for a survey on *adapting* models to these shifts, please refer to Gama et al. (2014) and Lange et al. (2022).

4.4 Methodology

This section analyzes the methodology for detecting distribution shifts in data streams inputted to deep neural networks. We define data stream in Section 4.4.1 and we formalize window-based detection in Section 4.4.2.

4.4.1 Background

At test time, an unlabeled sequence of inputs or *data stream* is expected, sampled from the marginal target distribution q_X.

Definition 4.4.1 (Data stream)**.** A data stream $\mathcal{S}$ is a finite or infinite sequence of not necessarily independent observations typically grouped into *windows* (i.e., sets $\mathcal{W}_j^m = \{x_j, \dots, x_{j+m-1}\} \sim q_X$) of same size m,

$$\mathcal{S} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_m, \dots\} = \bigcup_{j=1}^{\infty} \mathcal{W}_j^m. \tag{4.1}$$

Under the data-shift scenario, let $\beta \in [0,1]$ be a mixture coefficient, we will write the true joint test pdf q_{XY} as a mixture of pdfs p and υ^2:

$$q_{XY}(\boldsymbol{x}, y) = (1-\beta) \cdot p_{XY}(\boldsymbol{x}, y) + \beta \cdot \upsilon_{XY}(\boldsymbol{x}, y). \tag{4.2}$$

Remark 1. *When $\beta = 0$, the test distribution matches the training distribution, i.e., there is no shift. Conversely, when $\beta = 1$, we have the largest shift between training and testing environments.*

4.4.2 Window Based Detection Framework

Predictions can be made sample by sample or window by window in a data stream. We introduced the single sample detection in Chapter 1.

In a *window based detection* scenario, we make the assumptions that 1.) there are available multiple reference samples, 2.) the instance's class label *are not* available right after prediction, and 3.) the model is not updated. So, given a *reference window* $\mathcal{W}_1^r \sim p_{XY}$ with r samples and test window $\mathcal{W}_2^m = \{\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_m\} \sim q_X$ with sample size m, our task is to determine whether they are both sampled from the source distribution or, equivalently, whether $p_{XY}(\boldsymbol{x}, y)$ equals $q_{X\widehat{Y}}(\boldsymbol{x}', \hat{y}')$ where $\hat{y}' = f(\boldsymbol{x}')$. The null and alternative hypothesis of the two-sample test of homogeneity writes:

$$H_0 : p_{XY}(\boldsymbol{x}, y) = q_{X\widehat{Y}}\left(\boldsymbol{x}', \hat{y}'\right) \text{ and } H_A : p_{XY}(\boldsymbol{x}, y) \neq q_{X\widehat{Y}}\left(\boldsymbol{x}', \hat{y}'\right). \tag{4.3}$$

In this case, the null hypothesis is that the two distributions are identical for all $(\boldsymbol{x}, y)$; the alternative is that they are not identical, which is a two-sided test. As testing this null hypothesis on a continuous and high dimensional space is unfeasible, we will compute a univariate score on each sample of the windows. With a slight abuse of notation let $s\left(\mathcal{W}^m, f\right) = \{s(\boldsymbol{x}_1, f), \ldots, s(\boldsymbol{x}_m, f)\}$ be a multivariate *proxy variable* to derive a unified large-scale window-based data shift detector. To compute the final window score, we rely on the Kolmogorov-Smirnov (Massey, 1951) two-sample hypothesis test over the proxy variable. The test statistic writes:

$$\text{KS}(\mathcal{W}_1^m, \mathcal{W}_2^r) = \sup_w \left| F_{2,m}(w) - F_{1,r}(w) \right|, \tag{4.4}$$

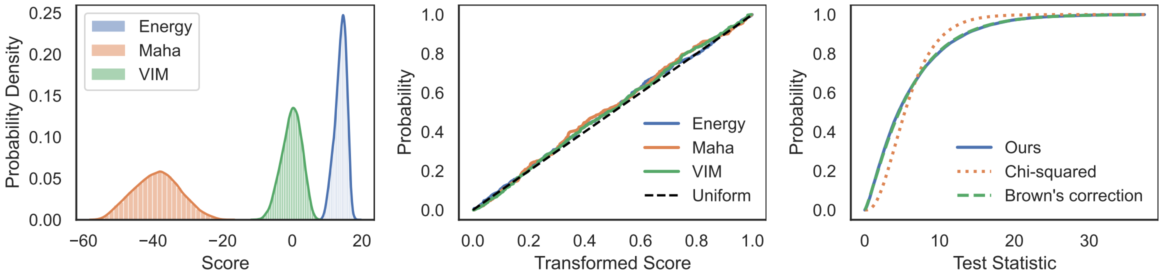
where $F_{1,r}$ and $F_{2,m}$ are the empirical cumulative distribution functions (ecdf) of the scores of each sample of the first and the second windows, respectively. Finally, The KS statistic is compared to a threshold, i.e., the window-based binary detector writes $D(\cdot) = \mathbb{1}\left[\text{KS}(\cdot, \mathcal{W}_1^r) \leq \gamma\right]$.

4.5 Main Contribution: Arbitrary Scores Combination

This section explains in detail the core contribution of the chapter. We present an algorithm to effectively combine arbitrary detection score functions inspired by *meta-analysis* (Glass, 1976), a statistical technique that combines the results of multiple studies to produce a single overall estimate. The first step is to transform the scores into p-values through a quantile normalization (Conover and Iman, 1981) (cf. Section 4.5.2). Then, with multiple detectors, the p-values can be combined using

[2] We assume that υ is unknown and differs significantly from p, i.e., $\frac{1}{2}\int_{\mathcal{X}\times\mathcal{Y}} |p(z) - \upsilon(z)| dz \geq \delta$.

a p-value combination method (cf. Section 4.5.3). Finally, we introduce an additional statistical treatment, since the p-values of the multiple tests over the same sample are not independent, to obtain better-calibrated statistics through the Brown's method (Brown, 1975) (cf. Section 4.5.4) for the Fisher's statistic. Haroush et al. (2021) treated the first step similarly and proposed a few combination methods for the second step. However, to the best of our knowledge, we are the first to propose correcting for correlated tests in the context of OOD and data shift detection.



(a) Score's pdf (in-distribution). (b) Quantile transformation's cdf. (c) Combined p-value cdf.

Figure 4.1: Illustration of the three steps of the proposed algorithm on an example with three score functions on in-distribution data. Our main experiments combine 15 scores.

4.5.1 Simple Statistics for Score Aggregation Falls Short

Common practices to combine different detection scores usually involve a notion of mean (de Carvalho, 2016) of scores accompanied with some assumptions. For instance, should the scores contribute equally to the final one, or should we consider a weighted sum, giving more importance to a few methods? Outlier score values should have preference over the other values or not? Should we take a more conservative or permissive approach in combining the scores? For example, the product of the available scores would give a low combined score if any of the individual scores is low. Using the minimum or maximum value among all anomaly scores can control if the method is more conservative or permissive. All of these combination methods are valid, but they are heavily dependent on the distributional characteristics. Since the choice of aggregating method depends on the characteristics of the data, let us focus on the special characteristics of OOD detection scores.

One intrinsic constraint on OOD detection is not having access to a sufficiently representative group of outlier data, undermining techniques such as *metalearning* Opitz and Maclin (1999) and other supervised ensembling techniques. Another undesired characteristic is that detection scores usually exhibit very distinct distribution shapes, with different moments, as displayed in Fig. 4.1a. In order to mitigate some of this effects, some simple statistical approaches are commonly used. They include normal standardization or *z-score normalization*, where the individual score r.v $S_i = s_i(X, f)$ is converted into a standard score $Z_i = (S_i - \bar{S}_i)/\sigma_{S_i}$ where the absolute value of Z_i represents the distance between the raw score and the population mean in units of the standard deviation σ_{S_i}. Even though this method correct the first two moments of the distributions, it fails to accommodate for skewness, kurtosis, and multimodality. Another common data normalization is *min-max scaling*, with statistics $Z_i = (S_i - \min S_i)/(\max S_i - \min S_i)$. While min-max scaling guarantees that the final score will be in the same range of zero and one, it also fail to correct many other characteristics.

The lack of control over the moments of the resulting distribution makes the task of combining scores much more challenging. To address this issue, we stress the importance of pre-processing the scores with a quantile normalization instead.

4.5.2 Quantile Normalization: Managing Disparate Score Distribution

Each detector's score r.v. $S_i = s_i(X, f)$ follows very different distributions depending on the model's architecture, the dataset it was trained on, and, of course, the score function s_i. In order to combine them effectively, we propose first to apply a quantile normalization (Bolstad et al., 2003), which exhibits interesting statistical properties (Gallón et al., 2013). Let $S_i : \Omega \mapsto \mathbb{R}$ be a continuous univariate r.v. captured by a cumulative density function (cdf) $F_i(\delta) = \Pr(S_i \leq \delta)$ for $i \in \{1, \ldots, k\}$ and $\delta \in \mathbb{R}$. Its *empirical cumulative density function* $\widehat{F}_i : \mathbb{R} \mapsto [0, 1]$ is defined by

$$\widehat{F}_i^r(\delta) = \frac{1}{r}\sum_{i=1}^{r} \mathbb{1}\left[S_i \leq \delta\right], \tag{4.5}$$

which converges almost surely to the true cdf for every δ by the Dvoretzky-Kiefer-Wolfowitz-Massart inequality (Massart, 1990). We are going to estimate this function using a subsample of size r of the training or validation set if available. The resulting r.v. is uniformly distributed in the interval $[0, 1]$. As a result, for each detector i and sample $\boldsymbol{x}$, we can obtain a p-value:

$$\mathrm{p}_i(\boldsymbol{x}) = P_{H_0}\left(S_i \leq s_i(\boldsymbol{x}, f)\right) = \Pr\left(S_i \leq s_i(\boldsymbol{x}, f) | H_0\right) \approx \widehat{F}_i^r\left(s_i(\boldsymbol{x}, f)\right). \tag{4.6}$$

A decision is made by comparing the p-value to a desired significance level α. If $\mathrm{p} < \alpha$, then the null hypothesis H_0 is rejected, and the sample is believed to be OOD. Even though we derived everything for the single sample case, this formulation can be extended to the window-based scenario.

4.5.3 Combining Multiple p-Values

Our objective is to aggregate a set of $k \geq 2$ scores (or p-values) in such a way that their synthesis exhibits better properties, such as improved robustness or detection performance, by consolidating each method's decision boundaries. Unfortunately, since q is not known and p is hard to estimate, designing an optimal test is unfeasible according to Neyman–Pearson's Fundamental Lemma (Lehmann and Romano, 2005). However, there are several possible empirical combination methods, such as Tippett (1931) $\min_i \mathrm{p}_i$, Neyman and Pearson (1933) $2\sum_i^k \ln(1 - \mathrm{p}_i)$, Wilkinson (1951) $\max_i \mathrm{p}_i$, Edgington (1972) $\sum_{i=1}^k \mathrm{p}_i$, and Simes (1986) $\min_i \frac{k}{i}\mathrm{p}_i$ for sorted p-values. We are going to explain in detail Fisher's method (Fisher, 1925; Mosteller and Fisher, 1948) in the main manuscript, also referred to as the chi-squared method, and Stouffer's method (Stouffer et al., 1949) in the appendix Section 4.A.1, as they exhibit good properties that will be explored in the following.

If the p-values are the independent realizations of a uniform distribution, i.e., for in-distribution data, $-2\sum_{i=1}^k \ln \mathrm{p}_i \sim \chi^2_{2k}$ follows a chi-squared distribution with $2k$ degrees of freedom. Finally, for a test input $\boldsymbol{x}$, Fisher's detector score function can be defined as

$$s_F(\boldsymbol{x}, f) = -2\sum_{i=1}^{k} \ln \widehat{F}_i(s_i(\boldsymbol{x}, f)). \tag{4.7}$$

Fisher's test has interesting qualitative properties, such as sensitivity to the smallest p-value, and it is generally more appropriate for combining positive-valued data (Heard and Rubin-Delanchy, 2017) with matches the properties of most OOD scores.

4.5.4 Correcting for Correlated p-values

It should be noted that Fisher's method depends on the assumption of independence and uniform distribution of the p-values. However, the p-values for the same input sample are not independent. Brown (1975) proposes modeling the r.v $s_F(\cdot)$ using a scaled chi-squared distribution, i.e.,

$$s_F(\cdot) \sim c\chi^2(k'), \text{ with } c = \operatorname{Var}(S_F)/(2\mathbb{E}[S_F]) \text{ and } k' = 2(\mathbb{E}[S_F])^2/\operatorname{Var}(S_F). \tag{4.8}$$

With this simple trick, we approach more interpretable results, as we know in advance the distribution followed by the in-distribution data under our combined score. As such, we can leverage calibrated confidence values given by the true cdf and leverage more powerful single-sample statistical tests for window-based data shift detection.

Remark 2. *Commonly, the binary detection threshold γ for a score is set based on a certain quantile of the score's value on an in-distribution validation set. Usually, this value is set to have 95% of entities correctly classified. By combining p-values with Fisher's method and correcting for correlation with Brown's method, we relax the need for a validation set to find γ, i.e., $\gamma = F^{-1}_{c\chi^2(k')}(\alpha)$.*

Remark 3. *Since Brown's method is simply a linear scaling, any detection evaluation metric (e.g., AUROC) computed for this method will be identical to the original Fisher's method statistic. However, we get calibrated scores given that we know the underlying data distribution, as observed in Fig. 4.1c.*

Algorithm 1 summarizes the offline steps of Combine and Conquer. Finally, at test time, the aggregated binary detection function for an input sample $\boldsymbol{x}$ writes for a given TPR desired performance $\alpha \in [0, 1]$:

$$d(\boldsymbol{x}) = \mathbb{1}\left[F_{c\chi^2(k')}\left(-2\sum_{i=1}^{k} \ln \widehat{F}_i(s_i(\boldsymbol{x}, f))\right) \leq \alpha\right] = \begin{cases} 1 \text{ data shift,} \\ 0 \text{ otherwise.} \end{cases} \tag{4.9}$$

4.6 Experimental Setup

In this section, we present and detail the experimental setup from a conceptual point of view. For all our main experiments, we set as *in-distribution* dataset *ImageNet-1K* (=ILSVRC2012; Deng et al., 2009) on ResNet (He et al., 2016) and Vision Transformers (Dosovitskiy et al., 2021) models. Our

Algorithm 1 Offline preparation for combining multiple detectors for OOD detection.

Require: Classifier f, in-distribution held-out data set $\mathcal{D}_r = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_r\}$, and $k \geq 2$ detection score functions denoted by $s_1, \dots s_k$.

```
$S \leftarrow \mathbf{0}_{r \times k}$                                   ▷ Initialize empty $r \times k$ matrix
for $\boldsymbol{x}_i \in \mathcal{D}_r$ do                               ▷ Fill the matrix with in-distribution scores
    for $j \in \{1, \dots, k\}$ do
        $S_{i,j} \leftarrow s_j(\boldsymbol{x}_i)$
    end for
end for
for $j \in \{1, \dots, k\}$ do                                  ▷ Define the empirical cdfs to compute p-values
    $\widehat{F}_j(\cdot) \leftarrow 1/r \sum_{i=1}^{r} \mathbb{1}[S_{i,j} \leq \cdot\,]$
end for
    ▷ The following steps are for the Fisher-Brown method. They can be easily adapted to other methods
for $i \in \{1, \dots, r\}$ do
    $\mathrm{p}_i \leftarrow -2 \sum_{j=1}^{k} \ln \widehat{F}_j(S_{i,j})$
end for
$\mu \leftarrow 1/r \sum_{i=1}^{r} \mathrm{p}_i, \;\; \sigma^2 \leftarrow 1/r \sum_{i=1}^{r} (\mathrm{p}_i - \mu)^2$
$c \leftarrow \sigma^2/(2\mu), \;\; k' \leftarrow 2\mu^2/\sigma^2$
return $\widehat{F}_1, \dots, \widehat{F}_k, c, k'$
```

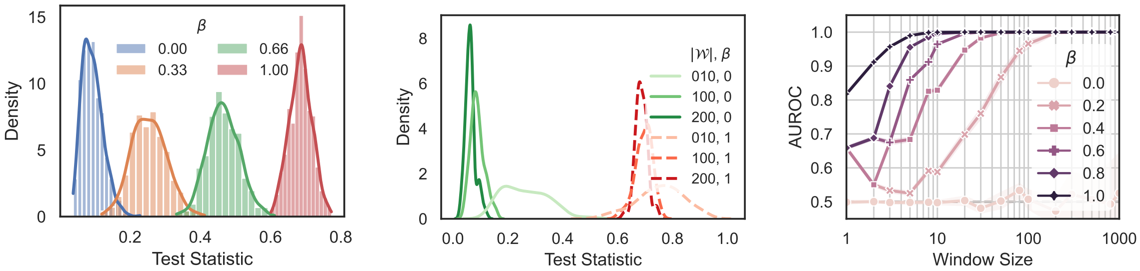
experiments encompass a full-spectrum setting on i.) classic OOD detection (Section 4.6.1), ii.) concept shift via independent window-based detection (Section 4.6.2; Par. 1), iii.) covariate shift via independent window-based detection (Section 4.6.2; Par. 2), and iv.) sequential shift detection via sequential window-based detection (Section 4.6.3).

4.6.1 Classic Out-of-Distribution Detection

We evaluate OOD detection performance on the curated **datasets** from Bitterwolf et al. (2023) that contain a clean subset of the far-OOD datasets: SSB-Easy (Vaze et al., 2022), OpenImage-O (Wang et al., 2022), Places (Zhou et al., 2017), iNaturalist (Horn et al., 2017), and Textures (Cimpoi et al., 2014); and the near-OOD datasets: SSB-Hard (Vaze et al., 2022), Species (Hendrycks et al., 2022), and NINCO (Bitterwolf et al., 2023). For the **evaluation metrics**, we consider the Area Under the Receiver Operating Characteristic curve (AUROC), which measures how well the OOD score distinguishes between out- and in-distribution data in a threshold-independent manner (higher is better). For the **baselines**, we consider the following post-hoc detection methods: MSP (Hendrycks and Gimpel, 2017), Energy (Liu et al., 2020), Maha (Lee et al., 2018b), Igeood (Dadalto et al., 2022), MaxCos (Techapanurak et al., 2020), ReAct (Sun et al., 2021), ODIN (Liang et al., 2018b), DICE (Sun and Li, 2022), VIM (Wang et al., 2022), KL-M (Hendrycks et al., 2022), Doctor (Granese et al., 2021), RMD (Fort et al., 2021), KNN (Sun et al., 2022), GradN (Huang et al., 2021). When needed, we followed the hyperparameter selection procedure suggested in the original papers. New methods can be easily integrated into our universal framework and should improve the robustness and, potentially, the performance of the group detector. In Section 4.7 we discuss the empirical results and analyze if there exists a optimal subset of detectors that boosts detection performance.

4.6.2 Independent Window-Based Detection

Concept shift. We suppose that full ID and corrupted windows formed by ID and OOD data from the OpenImage-O (OI-O) (Wang et al., 2022) dataset with mixing parameter β (Eq. (4.2)) are available. The objective of the detectors is to classify each test window as being corrupted or not by comparing it to a fixed reference window of size $r = 1000$ extracted from a validation set. We ran experiments with $\beta \in [0, 1]$ and with window sizes $|\mathcal{W}| \in \{1, \dots, 1000\}$. We use the KS two sample test described in Section 4.4.2 as window-based test statistics. Evaluation metrics and baselines are the same as described in Section 4.6.1. Fig. 4.2 shows Fisher's ensembled test statistic in different scenarios of mixture amount and window sizes. Fig. 4.2a shows the distribution of the test statistics for different mixture values from $\beta = 0$ (fully ID window) to $\beta = 1$ (fully OOD window). Fig. 4.2b displays how the distribution on the test statistic changes from flatter to peaky as we increase the window size (better seen in color). Finally, Fig. 4.2c demonstrates how the detection performance is affected by window sizes increase mixture coefficient. Note an AUROC of 0.5 for the case with $\beta = 0$, as expected. With a window size as low as 8, we can already perfectly distinguish fully corrupted from normal ones. Similar qualitative behavior is observed on all detectors.



(a) From ID to OOD window.

(b) Flat to peaky windows.

(c) Detection performance.

Figure 4.2: Test statistic distribution and detection performance w.r.t concept shift intensity and window size. Experiments ran for Fisher's method on a ResNet-50.

Model	Train	Val.	IN-R	IN-R (m)
ResNet-50	87.5	76.1	1.33	36.2
ResNet-101	90.0	77.4	1.67	39.3
ResNet-152	90.2	78.3	0.67	41.4
ViT-S-16	88.0	81.4	1.33	46.0
ViT-B-16	90.5	84.5	3.33	56.8
ViT-L-16	92.3	85.8	1.67	64.3

Table 4.1: Top-1 accuracies in percentage on the training and validation sets and on the domain drift on all and (m)asked classes outputs.

Covariate shift. We ran experiments with the ImageNet-R (IN-R) (Hendrycks et al., 2021) dataset, providing domain shift to 200 ID classes. Similarly to the novelty setup described in the previous paragraph, we suppose that the windows arrive independently from one another. We use the same reference window to compute metrics, and we vary the mix parameter and the window size in the same way. Fig. 4.8 is similar to Fig. 4.2 and shows the behavior of the combined p-values for detecting covariate shift in windows of a data stream. Similar qualitative observations are drawn. Table 4.1 display the accuracy of each model studied on the new domain. We can see that without masking only the classes present on IN-R, the drift is severe, with a top-1 accuracy of around 1% only. However, as we compute the top accuracy only on the 200 classes by masking the other 800, we can observe an amelioration in performance. In our experiments, we simulate the more realistic scenario by supposing that this mask is not available.

4.6.3 Sequential Drift Detection

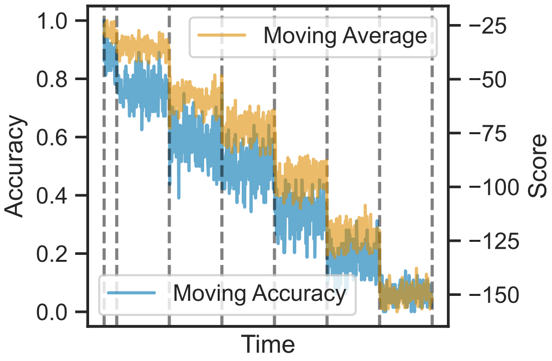


Figure 4.3: Data stream monitoring with correlation $\rho = 0.98$.

In this setup, differently from the independent window-based detection setting, we implement a sliding window of size 64 with a stride of one. We assume that the samples arrive sequentially and labels are unavailable to compute the true accuracy of the model on the current or past test windows. The objective is to see how well the moving average of the detection score will correlate with the moving accuracy of the model. By having a high correlation with accuracy, a machine learning practitioner can use the values of the score as an indicator if the system is suffering from any degrading data distribution shift. We ran experiments with the corrupted ImageNet (IN-C) (Hendrycks and Dietterich, 2019) dataset. The intensity of the drift increases over time from intensity 0 (training warmup set and part of the validation set without corruptions) to 5. Fig. 4.3 illustrates the monitoring pipeline with the moving accuracy on the left y-axis and the score's moving average on the right y-axis. The score's moving average can effectively follow the accuracy (hidden variable).

4.7 Results and Discussion

4.7.1 Out-of-Distribution Detection

Table 4.2 displays the experimental result on classic OOD detection for a ResNet-50 model on the setup described in Section 4.6.1. Fisher's method achieves state-of-the-art results on average AUROC, surpassing the previous SOTA by 1.4% (MaxCos). Also, the other six standard p-value combination strategies also achieve great results, validating our proposed meta-framework of Section 4.5. Similar tables for FPR and other architectures are available in the Section 4.A. Apart from achieving overall great performance capabilities, the most compelling observed property is the robustness compared to individual detection metrics. Fig. 4.4 shows the ranking per dataset and on average for selected methods. We can observe that, even though several detectors achieve top-1 performance in a few cases, there are several datasets in which they underperform, sometimes catastrophically. This is not true for the group methods, which can effectively combine the existing detectors to obtain a final score that successfully combines the multiple decision regions. For instance, Combine and Conquer with Fisher/Brown keeps top-4 performance in all cases on the ResNet-50 ImageNet benchmark and Stouffer/Hartung is top-5 in all cases.

Table 4.2: Numerical results in terms of AUROC (values in percentage) comparing p-value combination methods against literature for a ResNet-50 model trained on ImageNet. The left-hand side shows results on out-of-distribution detection, and the right-hand side shows results on concept (OI-O) and covariate (IN-R) shift detection with $|\mathcal{W}| = 3$ and $\beta = 1$.

	Out-of-Distribution Detection									Data Shift Detection	
Method	Avg.	SSB-H	NINCO	Spec.	SSB-E	OI-O	Places	iNat.	Text.	IN-R	OI-O
Fisher/Brown	**89.8**	75.8	84.3	88.7	91.0	**93.0**	93.1	95.9	96.4	**94.3** (0.2)	**95.7** (0.4)
Stouffer/Hartung	89.6	75.5	84.6	89.0	90.9	92.8	92.7	95.8	95.5	92.8 (0.2)	95.5 (0.4)
Edgington	89.3	75.2	84.6	89.0	**91.0**	92.5	92.1	95.5	94.4	92.5 (0.2)	95.3 (0.3)
Pearson	89.2	74.6	**84.9**	**89.4**	90.9	92.4	91.8	95.5	94.1	92.2 (0.3)	93.9 (0.4)
Simes	89.2	75.0	83.0	87.6	89.5	92.3	93.1	95.7	97.0	83.6 (0.5)	86.6 (0.7)
Tippet	88.5	74.8	80.9	86.7	87.3	91.7	93.5	95.9	97.2	82.0 (1.0)	81.5 (0.7)
Wilkinson	86.5	68.7	83.3	89.0	88.1	89.5	86.3	93.6	93.1	71.2 (1.8)	77.4 (0.9)
MaxCos	88.4	69.6	82.7	88.2	89.9	92.2	89.7	96.1	**98.4**	92.2 (0.3)	95.5 (0.4)
ReAct	87.4	75.0	80.1	87.2	82.3	90.4	**95.8**	**96.6**	91.6	92.2 (0.3)	94.5 (0.4)
ODIN	85.4	72.9	80.3	83.9	87.7	88.8	90.0	91.4	88.3	92.2 (0.5)	93.6 (0.4)
DICE	85.1	70.2	77.4	84.1	82.5	88.6	91.6	94.4	91.9	85.5 (0.3)	90.1 (0.4)
Energy	85.0	72.1	79.6	83.1	87.2	88.7	90.0	90.7	88.4	91.9 (0.3)	93.4 (0.4)
Igeood	84.7	71.4	80.1	83.0	88.8	88.0	88.8	90.2	87.6	91.0 (0.3)	93.3 (0.3)
VIM	84.3	66.4	78.9	80.7	89.3	90.3	83.7	87.9	97.5	92.2 (0.5)	95.4 (0.4)
KL-M	84.3	73.9	80.7	86.1	87.3	85.7	85.2	90.0	85.3	86.9 (0.6)	91.4 (0.9)
Doctor	84.2	75.9	80.6	85.1	87.0	85.1	86.7	89.7	83.8	85.2 (0.6)	89.9 (0.4)
RMD	83.5	**78.2**	82.7	87.7	82.9	84.9	81.3	87.6	82.7	89.9 (0.3)	93.1 (0.6)
MSP	83.5	75.5	79.9	84.5	86.1	84.1	85.9	88.7	83.0	83.6 (0.5)	89.0 (0.4)
KNN	83.4	64.3	79.6	83.3	88.0	87.2	83.0	84.1	97.6	84.6 (0.5)	89.2 (0.8)
GradN	82.6	63.3	74.4	83.1	76.2	84.4	91.1	96.0	92.5	49.7 (1.0)	67.4 (1.2)
Maha	69.6	55.3	65.7	70.3	70.6	73.9	60.0	72.7	88.4	71.2 (1.8)	77.6 (1.8)

4.7.2 Independent Window-Based Detection

Fig. 4.5 displays results on concept shift detection. Fig. 4.5a) shows the detectors' performance with the window size, showcasing a small edge in performance for Vim, Fisher's, and Stouffer's methods. Fig. 4.5b displays the impact of the mixture parameter. Fig. 4.5c shows that model size does mildly impact detection performance, with registered improvements for ResNet-152 over ResNet-50 on Fisher's method. The confidence interval bounds are computed over 10 different seeds and are quite narrow for all methods. Similar observations are drawn in the covariate shift results displayed in Fig. 4.10, except for the network scale impact, where we obtained more or less the same results for all sizes. On the right-hand side of Table 4.2, we showed that for both shifts, we demonstrated

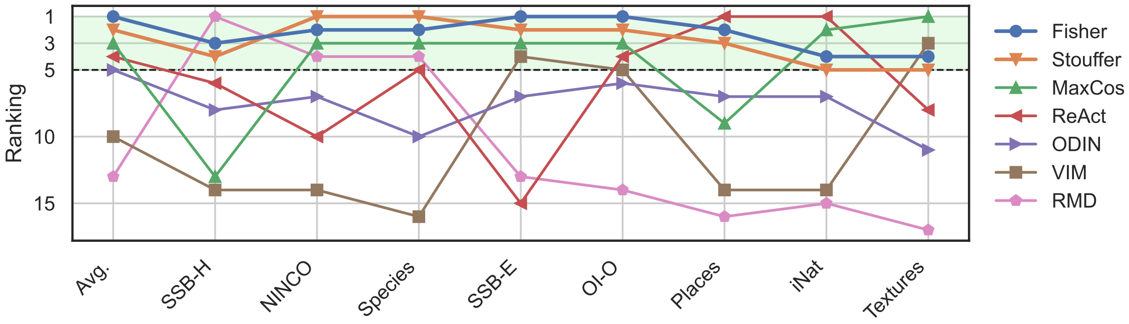
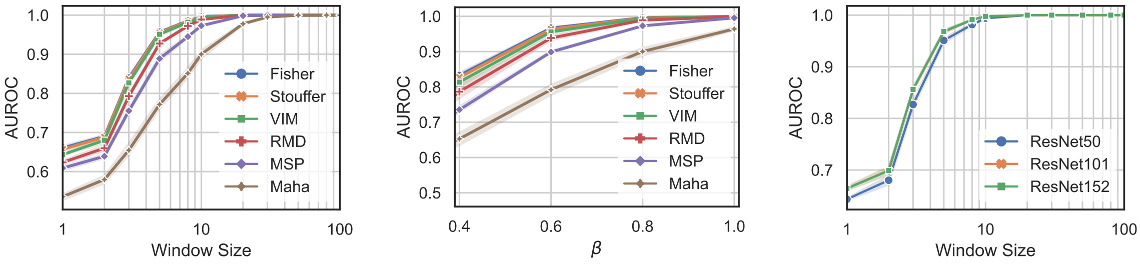


Figure 4.4: Ranking in terms of AUROC for a few selected methods for the ResNet-50 model. Note that the two displayed methods to combining tests obtain a top-5 ranking in every dataset, while state-of-the-art individual detectors vary significantly in performance.



(a) $|\mathcal{W}|$ impact with $\beta = 0.8$.

(b) β impact with $|\mathcal{W}| = 10$.

(c) Model size impact ($\beta = 0.8$).

Figure 4.5: Concept shift (OpenImage-O) detection performance on a ResNet-50 model (ImageNet).

improved performance by combining p-values, especially with Fisher's method. We also observe from the table that the concept shift benchmark is slightly easier than the covariate shift benchmark, probably biased because most OOD detectors were developed for the novel class scenario.

4.7.3 Results in a Sequential Stream

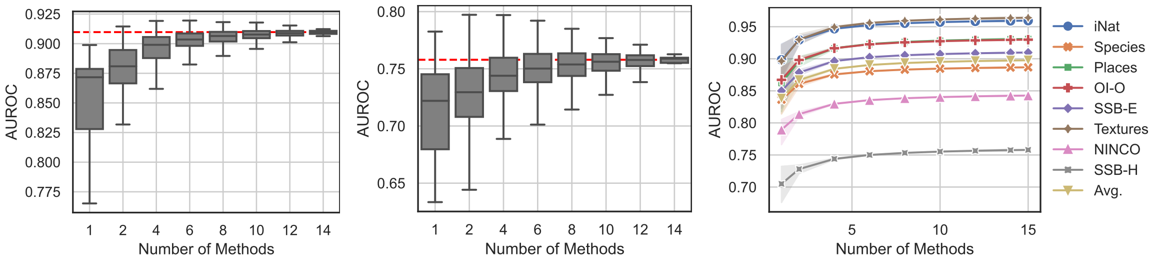
Table 4.3 displays the average results for the ImageNet-C dataset, including 19 kinds of covariate drifts. We can observe that the most performing methods are the scores function based on the softmax and logit outputs and that Fisher's method is on par with top-performing methods. We emphasize that, even though MSP and Doctor work well in this benchmark, they demonstrated poor performance on other benchmarks, notably on Table 4.2. This supports our claim that combining scores is the most effective approach for improving robustness and performance in general data shift detection.

Table 4.3: Average Pearson's correlation coefficient with the hidden accuracy with one standard deviation in parenthesis for top and bottom performing detection methods across 19 different corruptions on the sequential data shift detection scenario on a ResNet-50 model.

	Fisher	Doctor	MSP	Igeood	...	KNN	RMD	GradN	Maha
Avg.	0.96 (0.03)	0.96 (0.03)	0.96 (0.03)	0.95 (0.03)	...	0.92 (0.07)	0.92 (0.03)	0.91 (0.07)	0.81 (0.21)

4.7.4 On the Distillation of the Best Subset of Detectors

We provide a supervised study to showcase the potential impact of finding an optimal subset of detectors. We computed the performance of all possible subsets of $j < k$ methods, and we report our results in Fig. 4.6. We found out that 1.) surprisingly, removing the least performant detector from the pool does not necessarily increase performance; 2.) increasing the size of the subset improves probable detection on average and on worst performance; 3.) best subset selection benefits harder to find OOD samples; and 4.) not surprisingly, the best combination for the easy benchmark may be very different from the best subset on the harder one. We also list the best subset of four methods on average performance: {GradN, ReAct, MaxCos, RMD}, on an easy dataset (SSB-Easy): {DICE, MaxCos, KL-M, VIM}, and on a hard dataset (SSB-Hard): {MSP, GradN, ReAct, RMD}. Their AUROC and relative gain w.r.t all methods combined together are equal to 91.4 (+1.8%), 92.0 (+1.1)%, and 79.7 (+4.9%), respectively. *These observations support the main claim of this chapter that in a data-free scenario with specialized methods, combining all of them should greatly improve the safety of the underlying system.*



(a) SSB-Easy. (b) SSB-Hard. (c) Average for all datasets with 95% CI.

Figure 4.6: Evaluation of all possible subsets of detectors on the OOD detection benchmark. The dashed red line indicates the performance combining all detectors.

4.7.5 Limitations

Our study acknowledges that there is not a one-size-fits-all detector or a universally superior combination method, a finding supported by previous research (Heard and Rubin-Delanchy, 2017; Fang et al., 2022). This recognition underlines the inherent complexity of real-world ML applications. Additionally, we recognize that the empirical cumulative distribution function may be susceptible to estimation errors, and the effectiveness of individual detector score functions can influence the performance of the aggregated score. It is also important to note that, although our investigation primarily focused on computer vision applications, similar techniques can be applied to diverse scenarios and application domains.

4.8 Final Remarks and Summary

This chapter introduces a highly adaptable and efficient approach to combining detectors while effectively addressing data distribution shifts. By converting arbitrary scores into p-values and incorporating meta-analysis tools, we have demonstrated consolidated decision boundaries that prevent catastrophic collapses observed on individual detectors. We also showed that Fisher's method corrected for correlated p-values demonstrates great properties, being a fully interpretable detection criterion. Through a meticulous empirical investigation, we have thoroughly validated our approach, assessing both single-example out-of-distribution detection and window-based data distribution shift detection, gaining significant robustness and detection performance across various domains. Looking ahead, our framework offers a robust foundation for enhancing the safety of AI systems.

We believe several directions for future research are left open. A promising path involves exploring the pattern in the performance of detectors across different kinds of drifts to enable subset selection, leading to enhanced detection accuracy. However, it might need validation on held-out labeled data or domain expertise to reflect the prior importance of the p-values. Furthermore, our proposed algorithm could be integrated into incremental and online learning algorithms, thereby enhancing their adaptability to evolving data streams, representing an exciting avenue for advancing machine learning applications.

4.A Appendix to Chapter 4

4.A.1 Combining Multiple p-Values with Stouffer's Method

The Stouffer et al. (1949) test statistics for combining p-values is given by:

$$s_S(\cdot) = \sum_{i=1}^{k} \Phi^{-1}(\mathrm{p}_i(\cdot)) \tag{4.10}$$

where Φ^{-1} is the *probit*, i.e., $\Phi^{-1}(\alpha) = \sqrt{2}\,\mathrm{erf}^{-1}(2\alpha - 1)$, where erf is the Gauss error function. If the p-values are independent, $s_S(\cdot) \sim \mathcal{N}(0,1)$, where $\mathcal{N}(\mu,\sigma^2)$ is the normal distribution with mean μ and standard deviation σ.

4.A.2 Correcting for Correlated p-Values with Hartung's Method

Hartung (1999) method aims to correct Stouffer's test for correlated p-values. The group statistics writes:

$$s_H(\cdot;\boldsymbol{w},\rho) = \frac{\sum_{i=1}^{k} w_i \Phi^{-1}(\mathrm{p}_i(\cdot))}{\sqrt{(1-\rho)\sum_{i=1}^{k} {w_i}^2 + \rho\left(\sum_{i=1}^{k} w_i\right)^2}} \underset{H_0}{\sim} \mathcal{N}(0,1) \tag{4.11}$$

with ρ a real-valued parameter and $\sum_{i=1}^{k} w_i \neq 0$. Hartung showed that an unbiased estimator of ρ based on p_i under H_0 is given by:

$$\widehat{\rho} = 1 - \mathbb{E}\left[\frac{1}{k-1}\sum_{i=1}^{k}\left(\Phi^{-1}(\mathrm{p}_i) - \frac{1}{k}\sum_{i=1}^{k}\Phi^{-1}(\mathrm{p}_i)\right)^2\right]. \tag{4.12}$$

Assuming equal weights, we repeated a similar experiment as the one of Fig. 4.1, replacing the chi-squared with a standard normal to see how well the correction works. We can observe in Fig. 4.7 that the corrected statistic indeed approximates a standard normal distribution. Unlike Brown's method, Hartung's method corrects the statistics directly instead of correcting the parameters of the underlying distribution.

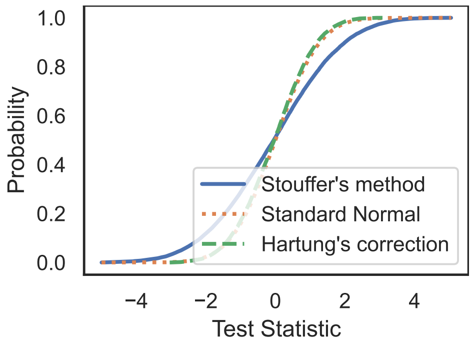
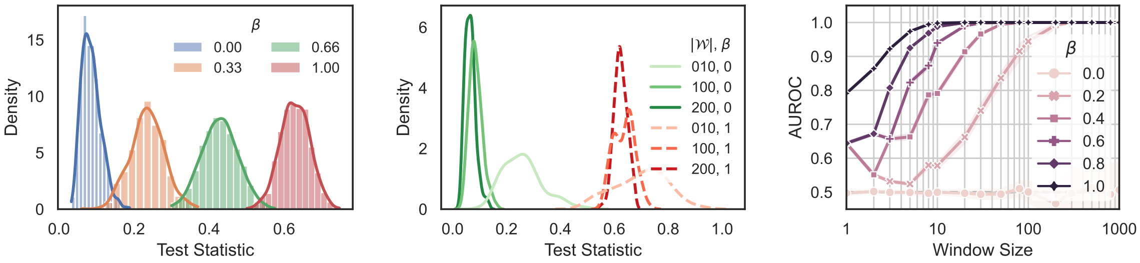


Figure 4.7: Stouffer's method corrected for correlated p-values with Hartung's method to obtain a standard normal distribution when evaluated on in-distribution data (null hypothesis), also obtaining interpretable results.

4.A.3 Additional Results

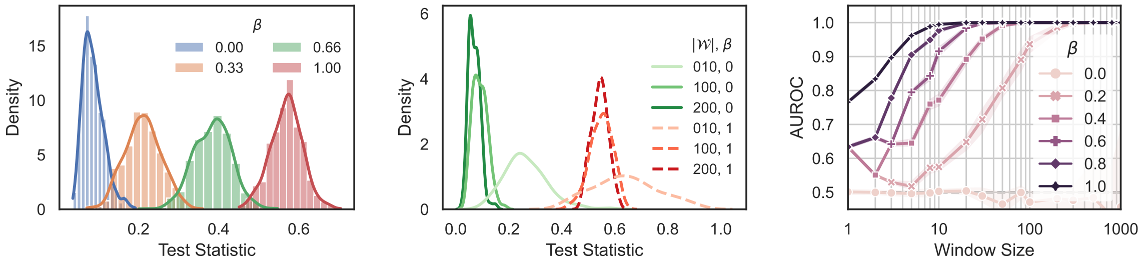


(a) From ID to OOD window.

(b) ID vs. OOD windows.

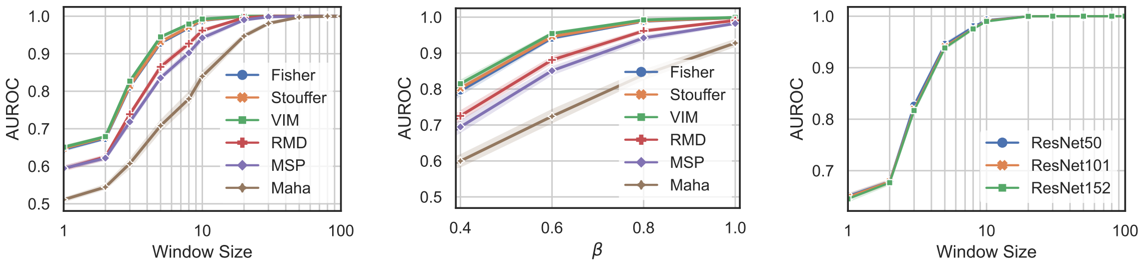
(c) Detection performance.

Figure 4.8: Test statistic behavior and detection performance in function of the covariate shift intensity and window size. Experiments ran on a ResNet-50.



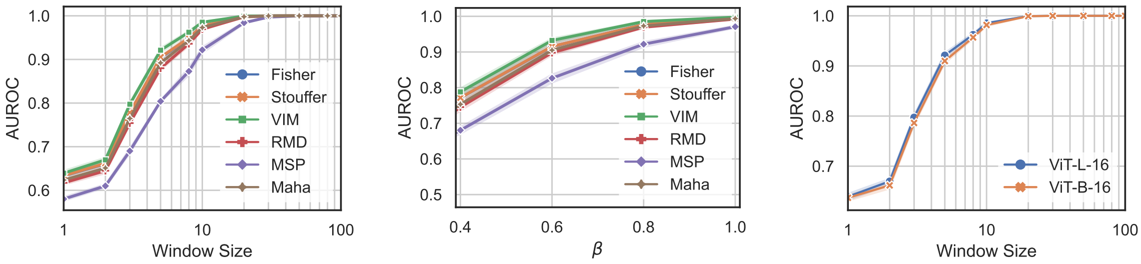
(a) From ID to OOD window. (b) ID vs. OOD windows. (c) Detection performance.

Figure 4.9: Test statistic behavior and detection performance in function of the covariate shift intensity and window size. Experiments ran on a ViT-L-16.



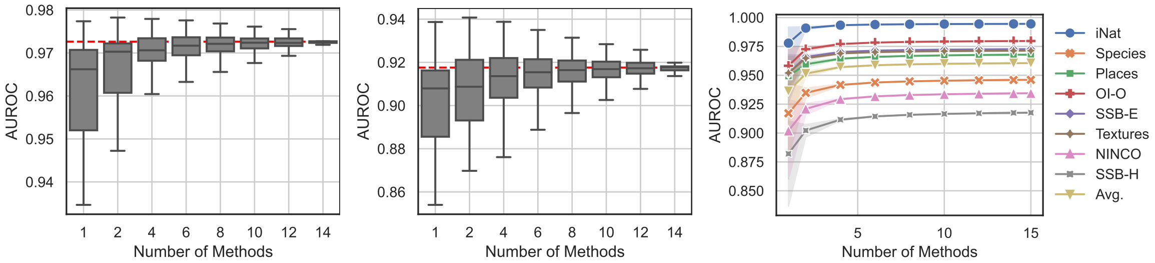
(a) $|\mathcal{W}|$ impact with $\beta = 0.8$. (b) β impact with $|\mathcal{W}| = 10$. (c) Model size impact ($\beta = 0.8$).

Figure 4.10: Covariate shift (ImageNet-R) detection performance on a ResNet-50 model (ImageNet).



(a) $|\mathcal{W}|$ impact with $\beta = 0.8$. (b) β impact with $|\mathcal{W}| = 10$. (c) Model size impact ($\beta = 0.8$).

Figure 4.11: Covariate shift (ImageNet-R) detection performance on a ViT-L-16 model (ImageNet).



(a) SSB-Easy. (b) SSB-Hard. (c) Average for all datasets with 95% CI.

Figure 4.12: Evaluation of all possible subsets of detectors on the OOD detection benchmark for a ViT-L-16 model. The dashed red line indicates the performance combining all detectors.

CHAPTER 5

A Data-Driven Measure of Relative Uncertainty for Misclassification Detection

5.1 Introduction

Usual uncertainty measures such as Shannon entropy do not provide an effective way to infer the real uncertainty associated with the model's predictions. This paper introduces a novel data-driven measure of uncertainty relative to an observer for misclassification detection inspired by Rao (1982). By learning patterns in the distribution of soft-predictions, our uncertainty measure can identify misclassified samples based on the predicted class probabilities. Interestingly, according to the proposed measure, soft predictions corresponding to misclassified instances can carry much uncertainty, even though they may have low Shannon entropy. We demonstrate empirical improvements over multiple image classification tasks, outperforming state-of-the-art misclassification detection methods.

It relies on negative and positive instances to capture meaningful patterns in the distribution of soft-predictions. It yields high and low uncertainty values for negative and positive instances, respectively. *Our measure is "relative", as it is not characterized axiomatically, but only serves the purpose of measuring uncertainty of positive instances relative to negative ones from the point of view of a subjective observer d.*. By learning to minimize the uncertainty in positive instances and to maximize it in negative instances, our metric can effectively capture meaningful information to differentiate between the underlying structure of distributions corresponding to two categories of data.

The contents of this chapter will be based on the work Dadalto et al. (2024b) that was a collaboration with my co-authors Marco Romanelli, Georg Pichler, and Pablo Piantanida. The code is available at the url[1].

[1] `https://github.com/edadaltocg/relative-uncertainty/`

5.2 Summary of Contributions

1. We leverage a novel statistical framework for categorical distributions to devise a learnable measure of relative uncertainty (REL-U) for a model's predictions, which induces large uncertainty for negative instances, even if they may lead to low Shannon entropy (cf. Section 5.3);

2. We propose a closed-form solution for training REL-U in the presence of positive and negative instances (cf. Section 5.4);

3. We report significantly favorable and consistent results over different models and datasets, considering both natural misclassifications within the same statistical population, and in case of distribution shift, or *mismatch*, between training and testing distributions (cf. Section 5.5).

5.3 A Data-Driven Measure of Uncertainty

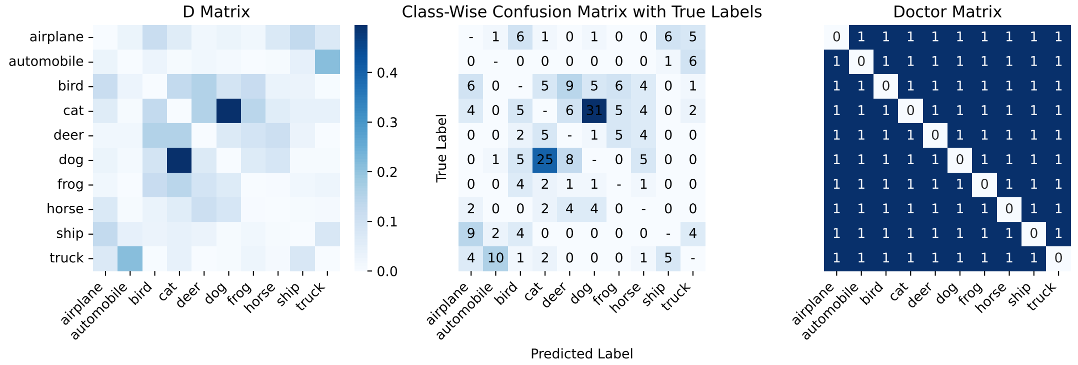


Figure 5.1: Intuitive example illustrating the advantage of REL-U compared to entropy-based methods: REL-U (left-end side heatmap) captures the real uncertainty (central heatmap) much better than Doctor (Granese et al., 2021); a detailed analysis is provided in Section 5.5.3.

Before we introduce our method, we start by recalling basic definitions and notations. Then, we describe our statistical model and some useful properties of the underlying detection problem.

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a (possibly continuous) feature space and let $\mathcal{Y} = \{1, \ldots, C\}$ denote the label space related to some task of interest. Moreover, we denote by p_{XY} be the underlying joint probability distribution on $\mathcal{X} \times \mathcal{Y}$. We assume that a machine learning model is trained on some training data, which ultimately yields a model that, given samples $\boldsymbol{x} \in \mathcal{X}$, outputs a probability mass function (pmf) on $\mathcal{Y}$, which we denote as a vector $\widehat{\mathbf{p}}(\mathbf{x})$. This may result from a soft-max output layer, for example. A predictor $f\colon \mathcal{X} \to \mathcal{Y}$ is then constructed, which yields $f(\boldsymbol{x}) = \arg\max_{y \in \mathcal{Y}} \widehat{\mathbf{p}}(\mathbf{x})_y$. We note that we may also interpret $\widehat{\mathbf{p}}(\mathbf{x}) \in \mathcal{Y}$ as the probability distribution of $\widehat{Y}$, which, given $\boldsymbol{X} = \boldsymbol{x}$, is distributed according to $p_{\widehat{Y}|\boldsymbol{X}}(y|\boldsymbol{x}) \triangleq \widehat{\mathbf{p}}(\mathbf{x})_y$.

In statistics and information theory, many measures of uncertainty were introduced, and some were utilized in machine learning to great effect. Among these are Shannon entropy (Shannon, 1948, Sec. 6), Rényi entropy (Rényi, 1961), q-entropy (Tsallis, 1988), as well as several divergence measures, capturing a notion of distance between probability distributions, such as Kullback-Leibler divergence (Kullback and Leibler, 1951), f-divergence (Csiszár, 1964), and Rényi divergence (Rényi, 1961). These definitions are well motivated, axiomatically, and/or by their use in coding theorems. While some measures of uncertainty offer flexibility by choosing parameters, e.g., α for Rényi α-entropy, they are invariant w.r.t. relabeling of the underlying label space. In our case, however, this semantic meaning of specific labels can be important, and we do not expect a useful measure of "relative" uncertainty to satisfy this invariance property.

Recall that the quantity $\widehat{\mathbf{p}}(\mathbf{x})$ is the posterior distribution output by the model given the input $\boldsymbol{x}$. The entropy measure of Shannon (Shannon, 1948, Sec. 6)

$$H(\widehat{Y}|\boldsymbol{x}) \triangleq -\sum_{y\in\mathcal{Y}} \widehat{\mathbf{p}}(\mathbf{x})_y \log\left(\widehat{\mathbf{p}}(\mathbf{x})_y\right) \tag{5.1}$$

and the concentration measure of Gini (Gini, 1912)

$$s_{\text{gini}}(\boldsymbol{x}) \triangleq 1 - \sum_{y\in\mathcal{Y}} \left(\widehat{\mathbf{p}}(\mathbf{x})_y\right)^2 \tag{5.2}$$

have commonly been used to measure the dispersion of a categorical random variable $\widehat{Y}$ given a sample $\boldsymbol{x}$. It is worth to emphasize that either measure may be used to carry out an analysis of dispersion for a random variable predicting a discrete value (e.g., a label). This is comparable to the analysis of variance for the prediction of continuous random values.

Regrettably, these measures suffer from two major inconveniences: they are invariant to relabeling of the underlying label space, and, more importantly, they lead to very low values for overconfident predictions, even if they are wrong. These observations make both Shannon entropy and the Gini coefficient unfit for our purpose, i.e., the detection of misclassification instances. Evidently, we need a novel measure of uncertainty that can operate on probability distributions $\widehat{\mathbf{p}}(\mathbf{x})$, and that allows us to identify meaningful patterns in the distribution from which uncertainty can be inferred from data. To overcome the aforementioned difficulties, we propose to construct a class of uncertainty measures that is inspired by the measure of diversity investigated in Rao (1982), defined as

$$s_d(\boldsymbol{x}) \triangleq \mathbb{E}[d(\widehat{Y},\widehat{Y}')|\boldsymbol{X}=\boldsymbol{x}] = \sum_{y\in\mathcal{Y}}\sum_{y'\in\mathcal{Y}} d(y,y')\widehat{\mathbf{p}}(\mathbf{x})_y\widehat{\mathbf{p}}(\mathbf{x})_{y'}, \tag{5.3}$$

where $d \in \mathcal{D}$ is in a class of distance measures and, given $\boldsymbol{X}=\boldsymbol{x}$, the random variables $\widehat{Y},\widehat{Y}' \sim \widehat{\mathbf{p}}(\mathbf{x})$ are independently and identically distributed according to $\widehat{\mathbf{p}}(\mathbf{x})$. The statistical framework we are introducing here offers great flexibility by allowing for an arbitrary function d that can be learned from data, as opposed to fixing a predetermined distance as in Rao (1982). *In essence, we regard the uncertainty in equation 5.3 as relative to a given observer d, which appears as a parameter in the definition.* To the best of our knowledge, this is a fundamentally novel concept of uncertainty.

5.4 From Uncertainty to Misclassification Detection

We wish to perform misclassification detection based on the statistical properties of soft-predictions of machine learning systems. In essence, the resulting problem requires a binary hypothesis test, which, given a probability distribution over the class labels (the soft-prediction), decides whether a misclassification event likely occurred. We follow the intuition that by examining the soft-prediction of categories corresponding to a given sample, the patterns present in this distribution can provide meaningful information to detect misclassified samples. For example, if a sample is misclassified, this can cause a significant shift in the soft-prediction, even if the classifier is still overconfident. From a broad conceptual standpoint, examining the structure of the population of predicted distributions is very different from the Shannon entropy of a categorical variable. We are primarily interested in the different distributions that we can distinguish from each other by means of positive (correctly classified) and negative (incorrectly classified) instances.

We first rewrite $s_d(\boldsymbol{x})$ Eq. (5.3) in order to make it amenable to learning the metric d. By defining the $C \times C$ matrix $D \triangleq (d_{ij})$ using $d_{ij} = d(i,j)$, we have $s_d(\boldsymbol{x}) = \widehat{\mathbf{p}}(\mathbf{x})\, D\, \widehat{\mathbf{p}}(\mathbf{x})^\top$. For $s_d(\boldsymbol{x})$ to yield a good detector g, we design a contrastive objective, where we would like $\mathbb{E}[s_d(\boldsymbol{X}_+)]$, which is the expectation over the positive samples, to be small compared to the expectation over negative samples, i.e., $\mathbb{E}[s_d(\boldsymbol{X}_-)]$. This naturally yields the following objective function, where we assume the usual properties of a distance function $d(y,y) = 0$ and $d(y',y) = d(y,y') \geq 0$ for all $y, y' \in \mathcal{Y}$.

Definition 5.4.1. Let us introduce our objective function with hyperparameter $\lambda \in [0,1]$,

$$\mathcal{L}(D) \triangleq (1-\lambda) \cdot \mathbb{E}\left[\widehat{\mathbf{p}}(\mathbf{X}_+)\, D\, \widehat{\mathbf{p}}(\mathbf{X}_+)^\top\right] - \lambda \cdot \mathbb{E}\left[\widehat{\mathbf{p}}(\mathbf{X}_-)\, D\, \widehat{\mathbf{p}}(\mathbf{X}_-)^\top\right] \tag{5.4}$$

and for a fixed $K \in \mathbb{R}^+$, define our optimization problem as follows:

$$\begin{cases} \text{minimize}_{D \in \mathbb{R}^{C\times C}}\, \mathcal{L}(D) & & \\ \text{subject to} & d_{ii} = 0, & \forall i \in \mathcal{Y} \\ & d_{ij} \geq 0, & \forall i,j \in \mathcal{Y} \\ & d_{ij} = d_{ji}, & \forall i,j \in \mathcal{Y} \\ & \text{Tr}(DD^\top) \leq K & \end{cases} \tag{5.5}$$

The first constraint in equation 5.5 states that the elements along the diagonal are zeros, which ensures that the uncertainty measure is zero when the distribution is concentrated at a single point. The second constraint ensures that all elements are non-negative, which is a natural condition, so the measure of uncertainty is non-negative. The natural symmetry between two elements stems from the third constraint, while the last constraint imposes a constant upper bound on the Frobenius norm of the matrix D, guaranteeing that a solution for the underlying optimization problem exists.

Proposition 2 (Closed form solution)**.** *The constrained optimization problem defined in Eq.* (5.5) *admits a closed form solution* $D^* = \frac{1}{Z}(d^*_{ij})$, *where*

$$d^*_{ij} = \begin{cases} \mathrm{ReLU}\left(\lambda \cdot \mathbb{E}\left[\widehat{\mathbf{p}}(\mathbf{X}_-)_i^\top \widehat{\mathbf{p}}(\mathbf{X}_-)_j\right] - (1-\lambda) \cdot \mathbb{E}\left[\widehat{\mathbf{p}}(\mathbf{X}_+)_i^\top \widehat{\mathbf{p}}(\mathbf{X}_+)_j\right]\right) & i \neq j \\ 0 & i = j \end{cases}. \quad (5.6)$$

The multiplicative constant Z *is chosen such that* D^* *satisfies the condition* $\mathrm{Tr}(D^*(D^*)^\top) = K$.

The proof is based on a Lagrangian approach and relegated to Section 5.A.1. Algorithm 2 in Section 5.A.2, summarizes all the main steps for the empirical evaluation, including the data preparation and the computation of the matrix D^*. Note that, apart from the zero diagonal and up to normalization,

$$D^* = \mathrm{ReLU}\left(\lambda \cdot \mathbb{E}\left[\widehat{\mathbf{p}}(\mathbf{X}_-)^\top \widehat{\mathbf{p}}(\mathbf{X}_-)\right] - (1-\lambda) \cdot \mathbb{E}\left[\widehat{\mathbf{p}}(\mathbf{X}_+)^\top \widehat{\mathbf{p}}(\mathbf{X}_+)\right]\right). \quad (5.7)$$

Finally, we define the Relative Uncertainty (REL-U) score for a given sample $\boldsymbol{x}$ as

$$s_{\text{REL-U}}(\boldsymbol{x}) \triangleq \widehat{\mathbf{p}}(\mathbf{x})\, D^*\, \widehat{\mathbf{p}}(\mathbf{x})^\top. \quad (5.8)$$

Remark 4. *Note that Eq.* (5.2) *is a special case of Eq.* (5.8) *when* $d_{ij} = 1$ *if* $i \neq j$ *and* $d_{ii} = 0$. *Thus,* $s_{1-d}(\boldsymbol{x}) = s_{\text{gini}}(\boldsymbol{x})$ *when choosing* d *to be the Hamming distance, which was also pointed out in (Rao, 1982, Note 1).*

5.5 Experiments and Discussion

In this section, we present the experiments conducted to validate our measure of uncertainty in the context of misclassification considering both the case when the training and test distributions *match*, and the case in which the two distributions *mismatch*. Although our method requires additional positive and negative instances, we show that lower amounts are needed (hundreds or few thousands) compared to methods that involve re-training or fine-tuning (hundreds of thousands).

5.5.1 Misclassification Detection on Matched Data

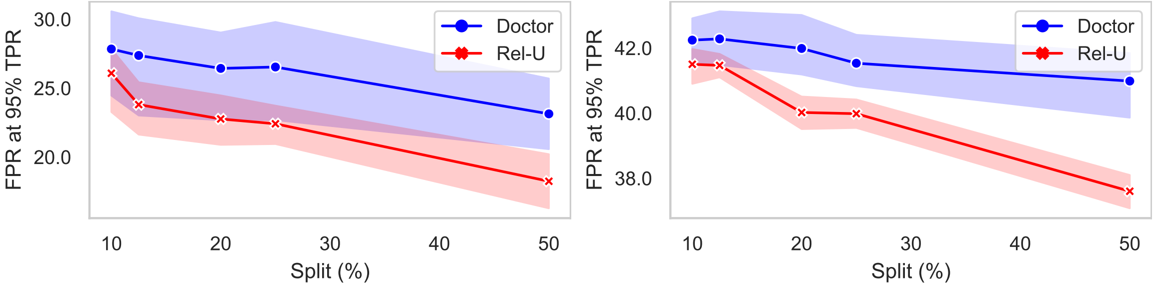
We designed our experiments as follows: for a given model architecture and dataset, we trained the model on the training dataset. We split the test set into two sets: one portion for tuning the detector (held out validation set) and the other for evaluating it. Consequently, we can compute all *hyperparameters* in an unbiased way and cross-validate performance over many splits generated from ten random seeds. For ODIN (Liang et al., 2018a) and Doctor (Granese et al., 2021), we found the best temperature (T) and input pre-processing magnitude perturbation (ϵ). For our method, we tuned the best lambda parameter (λ), T, and ϵ. For details on temperature and input pre-processing equations, see Section 5.A.5. As of *evaluation metric*, we consider the false positive rate (fraction of misclassifications detected as being correct classifications) when 95% of data is true positive (fraction of correctly classified samples detected as being correct classifications), denoted as FPR at 95% TPR (lower is better). AUROC results are similar among methods (see Fig. 5.7 in the appendix).

Table 5.1 showcases the misclassification detection performance in terms of FPR at 95% TPR of our method and the strongest baselines (MSP (Hendrycks and Gimpel, 2017), ODIN (Liang et al., 2018a), Doctor (Granese et al., 2021)) on different neural network architectures (DenseNet-121 (Huang et al., 2017), ResNet-34 (He et al., 2016)) trained on different datasets (CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009)) with different learning objectives (Cross-entropy loss, LogitNorm (Wei et al., 2022), MixUp (Zhang et al., 2018), RegMixUp (Pinto et al., 2022), OpenMix (Zhu et al., 2023)). Please refer to Section 5.A.3 for details on the baseline methods. We observe that, on average, our method performs best 11/20 experiments and is equal to the second best in 4/9 out of the remaining experiments. It works consistently better on all the models trained with cross-entropy loss and the models trained with RegMixUp objective, which achieved the best accuracy among them. We observed some negative results when training with logit normalization, but also, the accuracy of the base model decreased. Results for Bayesian methods for uncertainty estimation such as Deep Ensembles (Lakshminarayanan et al., 2017) and MCDropout (Gal and Ghahramani, 2016), as well as results for an MLP directly trained on the tuning data are reported in Table 5.4 in the Section 5.A.5. We report superior detection capabilities for the task at hand.

Table 5.1: Misclassification detection results across two different architectures trained on CIFAR-10 and CIFAR-100 with five different training losses. We report the average accuracy of these models and the detection performance in terms of average FPR at 95% TPR (lower is better) in percentage with one standard deviation over ten different seeds in parenthesis.

Model	Training	Accuracy	MSP	ODIN	Doctor	Rel-U
DenseNet-121 (CIFAR-10)	CrossEntropy	94.0	32.7 (4.7)	24.5 (0.7)	21.5 (0.2)	**18.3** (0.2)
	LogitNorm	92.4	39.6 (1.2)	**32.7** (1.0)	37.4 (0.5)	37.0 (0.4)
	Mixup	95.1	54.1 (13.4)	38.8 (1.2)	**24.5** (1.9)	37.6 (0.9)
	OpenMix	94.5	57.5 (0.0)	53.7 (0.2)	33.6 (0.1)	**31.6** (0.4)
	RegMixUp	95.9	41.3 (8.0)	30.4 (0.4)	23.3 (0.4)	**22.0** (0.2)
DenseNet-121 (CIFAR-100)	CrossEntropy	73.8	45.1 (2.0)	41.7 (0.4)	**41.5** (0.2)	**41.5** (0.2)
	LogitNorm	73.7	66.4 (2.4)	**60.8** (0.2)	68.2 (0.4)	68.0 (0.4)
	Mixup	77.5	48.7 (2.3)	41.4 (1.4)	**37.7** (0.6)	**37.7** (0.6)
	OpenMix	72.5	52.7 (0.0)	51.9 (1.3)	48.1 (0.3)	**45.0** (0.2)
	RegMixUp	78.4	49.7 (2.0)	45.5 (1.1)	43.3 (0.4)	**40.0** (0.2)
ResNet-34 (CIFAR-10)	CrossEntropy	95.4	25.8 (4.8)	19.4 (1.0)	14.3 (0.2)	**14.1** (0.1)
	LogitNorm	94.3	30.5 (1.6)	**26.0** (0.6)	31.5 (0.5)	31.3 (0.6)
	Mixup	96.1	60.1 (10.7)	38.2 (2.0)	26.8 (0.6)	**19.0** (0.3)
	OpenMix	94.0	40.4 (0.0)	39.5 (1.3)	**28.3** (0.7)	28.5 (0.2)
	RegMixUp	97.1	34.0 (5.2)	26.7 (0.1)	21.8 (0.2)	**18.2** (0.2)
ResNet-34 (CIFAR-100)	CrossEntropy	79.0	42.9 (2.5)	38.3 (0.2)	34.9 (0.5)	**32.7** (0.3)
	LogitNorm	76.7	58.3 (1.0)	55.7 (0.1)	**65.5** (0.2)	**65.4** (0.2)
	Mixup	78.1	53.5 (6.3)	43.5 (1.6)	**37.5** (0.4)	**37.5** (0.3)
	OpenMix	77.2	46.0 (0.0)	43.0 (0.9)	41.6 (0.3)	**39.0** (0.2)
	RegMixUp	80.8	50.5 (2.8)	45.6 (0.9)	40.9 (0.8)	**37.7** (0.4)

Ablation study. Fig. 5.2 displays how the amount of data reserved for the tuning split impacts the performance of the best two detection methods. We demonstrate how our data-driven uncertainty estimation metric generally improves with the amount of data fed to it in the tuning phase, especially on a more challenging setup such as on the CIFAR-100 model. Fig. 5.3 illustrates three ablation studies conducted to analyze and comprehend the effects of different factors on the experimental results. A separate subplot represents each hyperparameter ablation study, showcasing the outcomes obtained under specific conditions. *We observe that $\lambda \geq 0.5$, low temperatures, and low noise magnitude achieve better performance.* Overall, the method is shown to be robust to the choices of hyperparameters under reasonable ranges. Further discussion on hyperparameter selection is relegated to Section 5.A.4.



(a) CIFAR-10

(b) CIFAR-100

Figure 5.2: Impact of the tuning split size on the misclassification performance on a ResNet-34 model trained with supervised CrossEntropy loss for our method and the Doctor baseline. Hyperparameters are set to their default values ($T = 1.0$, $\epsilon = 0.0$, and $\lambda = 0.5$), i.e., only the impact of the validation split size is observed.

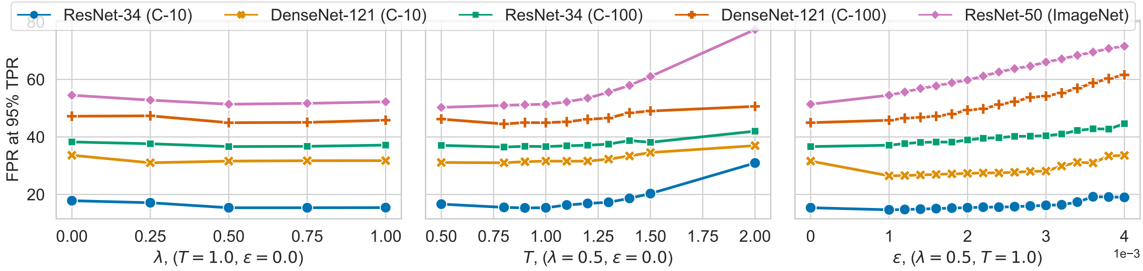


Figure 5.3: Ablation studies for temperature, lambda, and noise magnitude effects. The x-axis represents the experimental conditions, while the y-axis shows the performance metric.

Training losses or regularization is independent of detection. Previous work highlights the independence of training objectives from detection methods, which challenges the meaningfulness of evaluations. In particular, we identify three major limitations in (Zhu et al., 2023): The evaluation of post-hoc methods, such as Doctor and ODIN, lacks consideration of perturbation and temperature hyperparameters. Despite variations in accuracy and the absence of measures for coverage and risk, different training methods are evaluated collectively. Furthermore, the post-hoc methods are not assessed on these models. The primary flaw in their analysis stems from evaluating different detectors on distinct models, leading to comparisons between (models, detectors) tuples that have different misclassification rates. As a result, such an analysis may fail to determine the most performant detection method in real-world scenarios.

Does calibration improve detection? There has been growing interest in developing machine learning algorithms that are not only accurate but also well-calibrated, especially in applications where reliable probability estimates are desirable. In this section, we investigate whether models with calibrated probability predictions help improve the detection capabilities of our method or not. Previous work (Zhu et al., 2022a) has shown that calibration does not particularly help or impact misclassification detection on models with similar accuracies, however, they focused only on calibration methods and overlooked detection methods.

To assess this problem in the optics of misclassification detectors, we calibrated the soft-probabilities of the models with a temperature parameter (Guo et al., 2017). Note that this temperature is not necessarily the same value as the detection hyperparameter temperature. This calibration method is simple and effective, achieving performance close to state-of-the-art (Minderer et al., 2021). To measure how calibrated the model is before and after temperature scaling, we measured the expected calibration error (ECE) (Guo et al., 2017) before, with $T = 1$, and after calibration. We obtained the optimal temperature after a cross-validation procedure on the tuning set and measured the detection performance of the detection methods over the calibrated model on the test set. For the detection methods, we use the optimal temperature obtained from calibration, and no input pre-processing is conducted ($\epsilon = 0$), to observe precisely what is the effect of calibration. We set $\lambda = 0.5$.

Table 5.2 shows the detection performance over the calibrated models. We cannot conclude much from the CIFAR benchmark as the models are already well-calibrated out of the training, with ECE of around 0.03. In general, calibrating the models slightly improved performance on this benchmark. However, for the ImageNet benchmark, we observe that Doctor gained a lot from the calibration, while REL-U remained more or less invariant to calibration on ImageNet, suggesting that the performance of REL-U is robust under the model's calibration.

Table 5.2: Impact of model probability calibration on misclassification detection methods. The uncalibrated and the calibrated performances are in terms of average FPR at 95% TPR (lower is better) and one standard deviation in parenthesis.

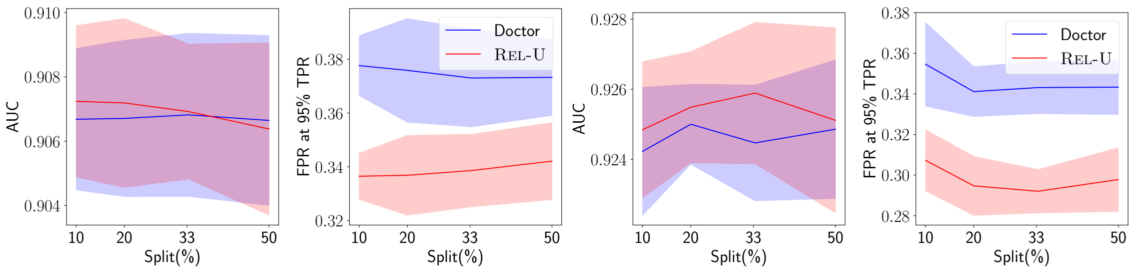
Architecture	Dataset	ECE$_1$	ECE$_T$	Uncal. Doctor	Cal. Doctor	Uncal. REL-U	Cal. REL-U
DenseNet-121	CIFAR-10	0.03	0.01	31.1 (2.4)	28.2 (3.8)	32.7 (1.7)	27.7 (2.1)
	CIFAR-100	0.03	0.01	44.4 (1.1)	45.9 (0.9)	45.7 (0.9)	46.6 (0.6)
ResNet-34	CIFAR-10	0.03	0.01	24.3 (0.0)	23.0 (1.4)	26.2 (0.0)	24.2 (0.1)
	CIFAR-100	0.06	0.04	40.0 (0.3)	38.7 (1.0)	40.6 (0.7)	38.9 (0.9)
ResNet-50	ImageNet	0.41	0.03	76.0 (0.0)	55.4 (0.7)	51.7 (0.0)	53.0 (0.3)

5.5.2 Mismatched Data

So far, we have evaluated methods for misclassification detection under the assumption that the data available to learn the uncertainty measure and that during testing are drawn from the same distribution. In this section, we consider cases in which this assumption does not hold true, leading to a mismatch between the generative distributions of the data. Specifically, we investigate two sources of mismatch: *i)* Datasets with different label domains, where the symbol sets and symbols cardinality are different in each dataset; *ii)* Perturbation of the feature space domain generated using popular distortion filters. Understanding how machine learning models and misclassification detectors perform under such conditions can help us gauge and evaluate their robustness.

5.5.2.1 Mismatch from Different Label Domains

We considered pre-trained classifiers on the CIFAR-10 dataset and evaluated their performance in detecting samples in CIFAR-10 and distinguishing them from samples in CIFAR-100, which has a different label domain. Similar experiments have been conducted in Fort et al. (2021); Zhu et al. (2023). The test splits were divided into a validation set and an evaluation set, with the validation set consisting of 10%, 20%, 33%, or 50% of the total test split and samples used for training were not reused.



(a) DenseNet-121 (b) ResNet-34

Figure 5.4: Impact of different validation set sizes (in percentage of test split) for mismatch detection.

For each split, we combine the number of validation samples from CIFAR-10 with an equal number of samples from CIFAR-100. To assess the validity of our results, each split has been randomly selected 10 times, and the results are reported in terms of mean and standard deviation in Fig. 5.4. We observe how our proposed data-driven method performs when samples are provided to describe the two groups accurately. To reduce the overlap between the two datasets, and in line with previous work (Fort et al., 2021), we removed the classes in CIFAR-100 that most closely resemble the classes in CIFAR-10.

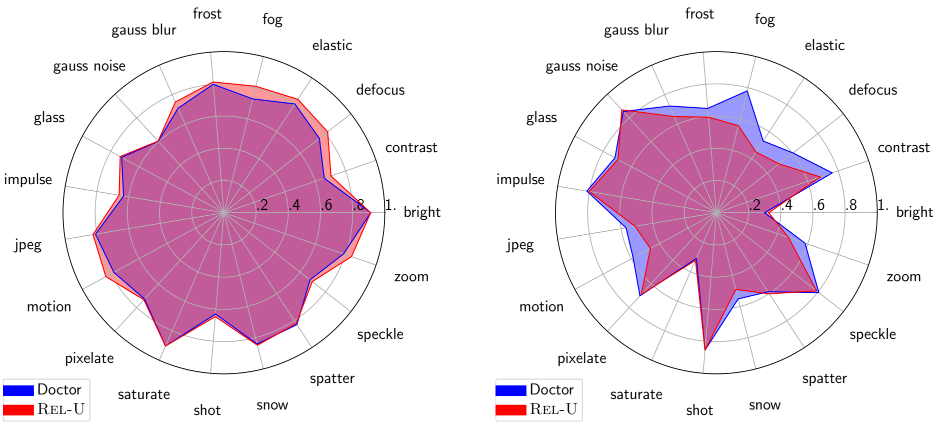
In order to reduce the overlap between the label domain of CIFAR-10 and CIFAR-100, in this experimental setup, we have ignored the samples corresponding to the following classes in CIFAR-100: bus, camel, cattle, fox, leopard, lion, pickup truck, streetcar, tank, tiger, tractor, train, and wolf.

5.5.2.2 Mismatch from feature space corruption

We trained a model on the CIFAR-10 dataset and evaluated its ability to detect misclassification on the popular CIFAR-10C corrupted dataset, which contains a version of the classic CIFAR-10 test set perturbed according to 19 different types of corruption and 5 levels of intensities. With this experiment, we aim to investigate if our proposed detector is able to spot misclassifications that arise from input perturbation, based on the sole knowledge of the misclassified patterns within the CIFAR-10 test split.

Consistent with previous experiments, we ensure that no samples from the training split are reused during validation and evaluation. To explore the effect of varying split sizes, we divide the test splits into validation and evaluation sets, with validation sets consisting of 10%, 20%, 33%, or 50% of the total test split. Each split has been produced 10 times with 10 different seeds and the average of the results has been reported in the spider plots in Fig. 5.5. In the case of datasets with perturbed feature spaces, we solely utilize information from the validation samples in CIFAR-10 to detect misclassifications in the perturbed instances of the evaluation datasets, without using corrupted data during validation. We present visual plots that demonstrate the superior performance achieved by our proposed method compared to other methods. Additionally, for the case of perturbed feature spaces, we introduce radar plots, in which each vertex corresponds to a specific perturbation type, and report results for intensity 5. This particular choice of intensity is motivated by the fact that it creates the most relevant divergence between the accuracy of the model on the original test split and the accuracy of the model on the perturbed test split. Indeed the average gap in accuracy between the original test split and the perturbed test split is reported in Table 5.3.

We observe that our proposed method outperforms Doctor in terms of AUROC and FPR, as demonstrated by the radar plots. As we can see, in the case of CIFAR-10 vs CIFAR-10C, the radar plots (Fig. 5.5) show how the area covered by the AUROC values achieves similar or larger values for the proposed method, indeed confirming that it is able to detect misclassifications in the mismatched data better. Moreover, the FPR values are lower for the proposed method. Additionally, as a particular case of a mismatch from feature space corruption, we have considered the task of detecting a mismatch between MNIST and SVHN, the results are reported in Fig. 5.6.



(a) AUROC

(b) FPR at 95% TPR

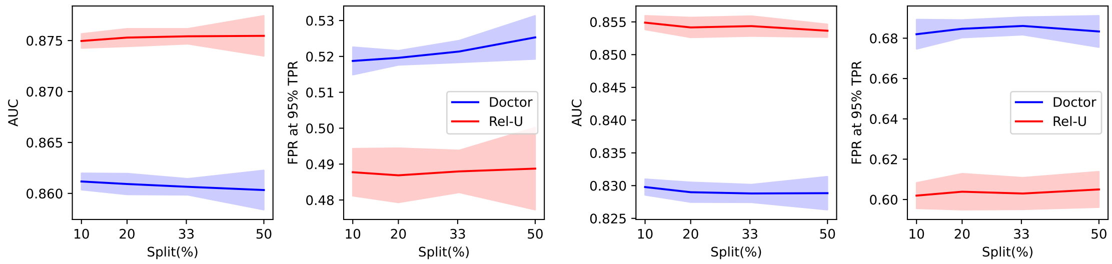
Figure 5.5: CIFAR-10 vs CIFAR-10C, ResNet-34, using 10% of the test split for validation.

5.5.3 Empirical Interpretation of the Relative Uncertainty Matrix.

Fig. 5.1 exemplifies the advantage of our method over the entropy-based methods in Eqs. (5.1) and (5.2). In particular, the left-end side heatmap represents the D matrix learned by optimizing Eq. (5.4) on CIFAR-10. Clearly, by only using the information required in Eq. (5.4) (no class labels or predictions required, only the probability vectors), our method is able to describe the uncertainty over

Table 5.3: We report the gap in accuracy between the original and the corrupted test set for the considered model. The gap is reported, and the average and standard deviation over the 19 different types of corruption for corruption intensity is equal to 5. The maximum and minimum gaps are also reported, with the relative corruption type.

Architecture	Average gap	Max gap	Min gap
DenseNet121	0.36 ± 0.18	0.66 (Gaussian Blur)	0.04 (Brightness)
ResNet34	0.35 ± 0.20	0.72 (Impulse Noise)	0.03 (Brightness)



(a) DenseNet-121.

(b) ResNet-34.

Figure 5.6: SVHN versus MNIST mismatch analysis.

different, and differently hard to predict, classes: darker shades of blue indicate higher uncertainty, while lighter shades of blue indicate lower uncertainty. The central heatmap is the predictor's class-wise true confusion matrix. The vertical axis represents the true class, while the horizontal axis represents the predicted class. For each combination of two classes ij, the corresponding cell reports the count of samples of class j that were predicted as class i. The correct matches along the diagonal are dashed for better visualization of the mistakes. The confusion matrix is computed on the same validation set used to compute the D matrix. Crucially, our uncertainty matrix can express different degrees of uncertainty depending on the specific combination of classes at hand. Let us focus for instance on the fact that most of the incorrectly classified dogs are predicted as cats, and vice-versa. The matrix D fully captures this by assigning high uncertainty to the cells at the intersection between these two classes. This is not the case for the entropy-based methods, which cannot capture such a fine-grained uncertainty, and assign the same uncertainty to all the cells, regardless of the specific combination of classes at hand.

5.5.4 Limitations

We presented machine learning researchers with a fresh methodological outlook and provided machine learning practitioners with a user-friendly tool that promotes safety in real-world scenarios. Some considerations should be put forward, such as the importance of cross-validating the hyperparameters of the detection methods to ensure their robustness on the targeted data and model. As a data-driven measure of uncertainty, to achieve the best performance, it is important to have enough samples at the disposal to learn the metric from as discussed on Section 5.5.1. As with every detection method, our method may be vulnerable to targeted attacks from malicious users.

5.6 Final Remarks and Summary

To the best of our knowledge, we are the first to propose REL-U, a method for uncertainty assessment that departs from the conventional practice of directly measuring uncertainty through the entropy of the output distribution. REL-U uses a metric that leverages higher uncertainty scores for negative data w.r.t. positive data, e.g., incorrectly and correctly classified samples in the context of misclassification detection and attains favorable results on matched and mismatched data. In addition, our method stands out for its *flexibility and simplicity*, as it relies on a closed-form solution to an optimization problem. Extensions to diverse problems present both an exciting and promising avenue for future research.

5.A Appendix to Chapter 5

5.A.1 Proof of Proposition 2

We have the optimization problem

$$\begin{cases} \text{minimize}_{D\in\mathbb{R}^{C\times C}}\ \mathcal{L}(D) & \\ \text{subject to} & d_{ii}=0,\ \forall i\in\{1,\dots,C\}; \\ & d_{ij}-d_{ji}=0,\ \forall i,j\in\{1,\dots,C\} \\ & \mathrm{Tr}(DD^\top)-K\leq 0 \\ & -d_{ij}\leq 0,\ \forall i,j\in\{1,\dots,C\} \end{cases} \tag{5.9}$$

in standard form (Boyd and Vandenberghe, 2004, eq. (4.1)) and can thus apply the KKT conditions (Boyd and Vandenberghe, 2004, eq. (5.49)). We find

$$\nabla\mathcal{L}(D^*)-\sum_{i,j}\xi^*_{ij}\nabla d^*_{ij}+\sum_i\mu^*_i\nabla d^*_{ii}+\sum_{ij}\nu^*_{ij}\nabla(d^*_{ij}-d^*_{ji})+\kappa^*\nabla(\mathrm{Tr}(D^*(D^*)^\top)-K)=0 \tag{5.10}$$

as well as the constraints

$$d^*_{ii}=0 \qquad d^*_{ij}-d^*_{ji}=0 \tag{5.11}$$

$$-d^*_{ij}\leq 0 \qquad \xi^*_{ij}\geq 0 \tag{5.12}$$

$$\xi^*_{ij}d_{ij}=0 \qquad \kappa^*\geq 0 \tag{5.13}$$

$$\kappa^*(\mathrm{Tr}(D^*(D^*)^\top)-K)=0 \tag{5.14}$$

We have

$$\nabla\mathcal{L}(D^*)=(1-\lambda)\cdot\mathbb{E}\left[\widehat{\mathbf{p}}(\mathbf{X}_+)^\top\widehat{\mathbf{p}}(\mathbf{X}_+)\right]-\lambda\cdot\mathbb{E}\left[\widehat{\mathbf{p}}(\mathbf{X}_-)^\top\widehat{\mathbf{p}}(\mathbf{X}_-)\right] \tag{5.15}$$

$$\nabla(\mathrm{Tr}(D^*(D^*)^\top)-K)=2D^* \tag{5.16}$$

and thus[2]

$$\begin{aligned} 0 &= (1-\lambda)\cdot\mathbb{E}\left[\widehat{\mathbf{p}}(\mathbf{X}_+)^\top\widehat{\mathbf{p}}(\mathbf{X}_+)\right] - \lambda\cdot\mathbb{E}\left[\widehat{\mathbf{p}}(\mathbf{X}_-)^\top\widehat{\mathbf{p}}(\mathbf{X}_-)\right] - \boldsymbol{\xi}^* + \operatorname{diag}(\boldsymbol{\mu}^*) \\ &\quad + \boldsymbol{\nu}^* - (\boldsymbol{\nu}^*)^\top + \kappa^* 2D^* \qquad (5.17) \\ D^* &= \frac{1}{2\kappa^*}\Big(-(1-\lambda)\cdot\mathbb{E}\left[\widehat{\mathbf{p}}(\mathbf{X}_+)^\top\widehat{\mathbf{p}}(\mathbf{X}_+)\right] + \lambda\cdot\mathbb{E}\left[\widehat{\mathbf{p}}(\mathbf{X}_-)^\top\widehat{\mathbf{p}}(\mathbf{X}_-)\right] + \boldsymbol{\xi}^* - \operatorname{diag}(\boldsymbol{\mu}^*) \\ &\quad - \boldsymbol{\nu}^* + (\boldsymbol{\nu}^*)^\top\Big) \qquad (5.18) \end{aligned}$$

As $\nabla\mathcal{L}(D^*)$ in Eq. (5.15) is already symmetric, we can choose $\boldsymbol{\nu}^* = \mathbf{0}$. We choose[3] $\boldsymbol{\mu}^* = \operatorname{diag}(\nabla\mathcal{L}(D^*))$ to ensures $d^*_{ii} = 0$. The non-negativity constraint can be satisfied by appropriately choosing $\mathbf{0} \leq \boldsymbol{\xi}^* = \operatorname{ReLU}(-\nabla\mathcal{L}(D^*))$. Finally, κ^* is chosen such that the constraint $\operatorname{Tr}(D^*(D^*)^\top) = K$ is satisfied. In total, this yields $D^* = \frac{1}{Z}\operatorname{ReLU}(d^*_{ij})$, where

$$d^*_{ij} = \begin{cases} -(1-\lambda)\cdot\mathbb{E}\left[\widehat{\mathbf{p}}(\mathbf{X}_+)_i^\top\widehat{\mathbf{p}}(\mathbf{X}_+)_j\right] + \lambda\cdot\mathbb{E}\left[\widehat{\mathbf{p}}(\mathbf{X}_-)_i^\top\widehat{\mathbf{p}}(\mathbf{X}_-)_j\right] & i \neq j \\ 0 & i = j \end{cases}. \qquad (5.19)$$

The multiplicative constant $Z = 2\kappa^* > 0$ is chosen such that D^* satisfies the condition $\operatorname{Tr}(D^*(D^*)^\top) = K$.

Remark 5. *A technical problem may occur when d^*_{ij} as defined in Eq.* (5.19) *is equal to zero for all $i, j \in \{1, 2, \dots, C\}$. In this case, D^* cannot be normalized to satisfy* $\operatorname{Tr}(D^*(D^*)^\top) = K$ *and the solution to the optimization problem in Eq.* (5.9) *is the all-zero matrix $D^* = \mathbf{0}$. I.e., no learning is performed in this case. We deal with this problem by falling back to the Gini coefficient Eq.* (5.2)*, where similarly, no learning is required.*

Equivalently, one may also add a small numerical correction ε to the definition of the ReLU *function, i.e.,* $\overline{\operatorname{ReLU}}(x) = \max(x, \varepsilon)$. *Using this slightly adapted definition when defining $D^* = \frac{1}{Z}\overline{\operatorname{ReLU}}(d^*_{ij})$ naturally yields the Gini coefficient in this case.*

5.A.2 Algorithm

In this section, we introduce a comprehensive algorithm to clarify the computation of the relative uncertainty matrix D^*.

At test time, it suffices to compute Eq. (5.8) to obtain the relative uncertainty of the prediction.

5.A.3 Details on Baselines and Benchmarks

In this section, we provide a comprehensive review of the baselines used on our benchmarks. We state the definitions using our notation introduced in Section 5.3.

[2] We use $\boldsymbol{X} = \operatorname{diag}(\boldsymbol{x})$ for a vector $\boldsymbol{x}$ to obtain a matrix $\boldsymbol{X}$ with $\boldsymbol{x}$ on the diagonal and zero otherwise.

[3] Slightly abusing notation, we also write $\boldsymbol{x} = \operatorname{diag}(\boldsymbol{X})$ to obtain the diagonal of the matrix $\boldsymbol{X}$ as a vector $\boldsymbol{x}$.

Algorithm 2 Offline relative uncertainty matrix computation.

```
Require: $\hat{\mathbf{p}}\colon \mathcal{X} \mapsto \mathbb{R}^C$ trained on a training set with $C$ classes, validation set $\mathcal{D}_m = \{(\mathbf{x}_j, y_j) \underset{\text{i.i.d}}{\sim} p_{XY}\}_{j=1}^{m}$, and hyperparameter $\lambda \in [0,1]$

  $\mathcal{D}_m^+ \leftarrow \varnothing,\ \ \mathcal{D}_m^- \leftarrow \varnothing$        ▷ Initialize empty positive and negative sets
  for $(\mathbf{x}, y) \in \mathcal{D}_m$ do        ▷ Fill the respective sets with positive or negative samples
      if $\arg\max_{y' \in \mathcal{Y}} \widehat{\mathbf{p}}(\mathbf{x})_{y'} = y$ then
          $\mathcal{D}_m^+ \leftarrow \mathcal{D}_m^+ \cup \{\widehat{\mathbf{p}}(\mathbf{x})\}$
      else
          $\mathcal{D}_m^- \leftarrow \mathcal{D}_m^- \cup \{\widehat{\mathbf{p}}(\mathbf{x})\}$
      end if
  end for
  $\boldsymbol{\mu}^+ \leftarrow \frac{1}{|\mathcal{D}_m^+|} \sum_{\hat{\mathbf{p}} \in \mathcal{D}_m^+} \hat{\mathbf{p}}^\top \hat{\mathbf{p}},\ \ \boldsymbol{\mu}^- \leftarrow \frac{1}{|\mathcal{D}_m^-|} \sum_{\hat{\mathbf{p}} \in \mathcal{D}_m^-} \hat{\mathbf{p}}^\top \hat{\mathbf{p}}$
  $D^* \leftarrow \mathbf{0}_{C \times C}$        ▷ $C$ by $C$ square matrix with zeroed out elements
  for $i \leftarrow 1, i \leq C, i \leftarrow i + 1$ do        ▷ Build $D^*$ according to Eq. (5.6)
      for $j \leftarrow 1, j \leq C, j \leftarrow j + 1$ do
          if $i \neq j$ then
              $d_{ij}^* \leftarrow \max\left(\lambda \mu_{ij}^- - (1 - \lambda)\, \mu_{ij}^+, 0\right)$
          end if
      end for
  end for
  return $D^*$
```

5.A.3.1 MLP

We trained an MLP with two hidden layers of 128 units with ReLU activation function and dropout with $p = 0.2$ on top of the hidden representations with a binary cross entropy objective on the validation set with Adam optimizer and learning rate equal to 10^{-3} until convergence. Results on misclassification are presented in Table 5.4.

5.A.3.2 MCDropout

Gal and Ghahramani (2016) propose to approximate Bayesian NNs by performing multiple forward passes with dropout enabled. To compute the confidence score, we averaged the logits and computed the Shannon entropy defined in Eq. (5.1). We set the number of inferences hyperparameter to $k = 10$ and we set the dropout probability to $p = 0.2$. Results on misclassification are presented in Table 5.4.

5.A.3.3 Deep Ensembles

Lakshminarayanan et al. (2017) propose to approximate Bayesian NNs by averaging the forward pass of multiple models trained on different initializations. We ran experiments with $k = 5$ different random seeds. To compute the confidence score, we averaged logits and computed the MSP response Eq. (1.19). Results on misclassification are presented in Table 5.4.

5.A.3.4 Conformal Predictions

According to **conformal learning** Angelopoulos and Bates (2021); Angelopoulos et al. (2021); Romano et al. (2020) the presence of uncertainty in predictions is dealt by providing, in addition to estimating the most likely outcome—actionable uncertainty quantification, a “prediction set” that provably “covers” the ground truth with a high probability. This means that the predictor implements an uncertainty set function, i.e., a function that returns a set of labels and guarantees the presence of the right label within the set with a high probability for a given distribution.

5.A.3.5 LogitNorm

Wei et al. (2022) observe that the norm of the logit keeps increasing during training, leading to overconfident predictions. So, they propose Training neural networks with logit normalization to hopefully produce more distinguishable confidence scores between in- and out-of-distribution data. They propose normalizing the logits of the cross entropy loss, resulting in the following loss function:

$$\ell(f(\mathbf{x}), y) = -\log \frac{\exp f_y(\mathbf{x})/(T\|\widehat{\mathbf{p}}(\mathbf{x})\|_2)}{\sum_{i=1}^{C} \exp f_i(\mathbf{x})/(T\|\widehat{\mathbf{p}}(\mathbf{x})\|_2)}. \tag{5.20}$$

5.A.3.6 MixUp

Zhang et al. (2018) propose to train a neural network on convex combinations of pairs of examples and their label to minimize the empirical vicinal risk. The mixup data is defined as

$$\tilde{\mathbf{x}} = \lambda \mathbf{x}_i + (1-\lambda)\mathbf{x}_j \text{ and } \tilde{y} = \lambda y_i + (1-\lambda) y_j \text{ for } i, j \in \{1, ..., n\}, \tag{5.21}$$

where λ is sampled according to a Beta(α, α) distribution. We used $\alpha = 1.0$ to train the models. Observe a slight abuse of notation here, where y is actually an one-hot encoding of the labels $y = [\mathbb{1}_{y=1}, \ldots, \mathbb{1}_{y=C}]^\top$.

5.A.3.7 RegMixUp

Pinto et al. (2022) use the cross entropy of the mixup data as in Eq. (5.21) with λ sampled according to a Beta$(10, 10)$ distribution as a regularizer of the classic cross entropy loss for training a network. The objective is balanced with a hyperparameter γ, usually set to 0.5.

5.A.3.8 OpenMix

Zhu et al. (2023) explicitly add an extra class for outlier samples and uses mixup as a regularizer for the cross entropy loss, but mixing between inlier training samples and outlier samples collected from the wild. It yields the objective

$$\mathcal{L} = \mathbb{E}_{\mathcal{D}_{\text{inlier}}}[\ell(f(\mathbf{x}), y)] + \gamma \mathbb{E}_{\mathcal{D}_{\text{outlier}}}[\ell(f(\tilde{\mathbf{x}}), \tilde{y})], \tag{5.22}$$

where $\gamma \in \mathbb{R}^+$ is a hyperparameter, $\tilde{\mathbf{x}} = \lambda \mathbf{x}_{\text{inlier}} + (1-\lambda)\mathbf{x}_{\text{outlier}}$, and $\tilde{y} = \lambda y_{\text{inlier}} + (1-\lambda)(C+1)$ with a slight abuse of notation. The parameter λ is sampled according to a $\text{Beta}(10, 10)$ distribution.

5.A.4 Additional comments on the ablation study for hyper-parameter selection

We conducted ablation studies on all relevant parameters: T, ϵ, and λ (cf. Section 5.5.1). It is crucial to emphasize that T is intrinsic to the network architecture and, therefore, must not be considered a hyper-parameter for REL-U. Additionally, the introduction of additive noise ϵ serves the purpose of ensuring a fair comparison with Doctor/ODIN, where the noise was utilized to enhance detection performance. Nevertheless, as indicated by the results in the ablation study illustrated in Fig. 5.3, $\epsilon = 0$ seems to be close to optimal most of the time, thereby positioning REL-U as an effective algorithm that relies only on the soft-probability output, therefore comparable to Granese et al. (2021); Liang et al. (2018b) in their version with no perturbation, and Hendrycks and Gimpel (2017). Furthermore, REL-U exhibits a considerable degree of insensitivity to various values of λ, as evident from Fig. 5.3. This suggests that a potential selection for λ could have been $\lambda = N_+/(N_+ + N_-)$, aiming to balance the ratio between the number of positive (N_+) and negative (N_-) examples. In such a scenario, there are no hyper-parameters at all.

5.A.5 Additional Results on Misclassification Detection

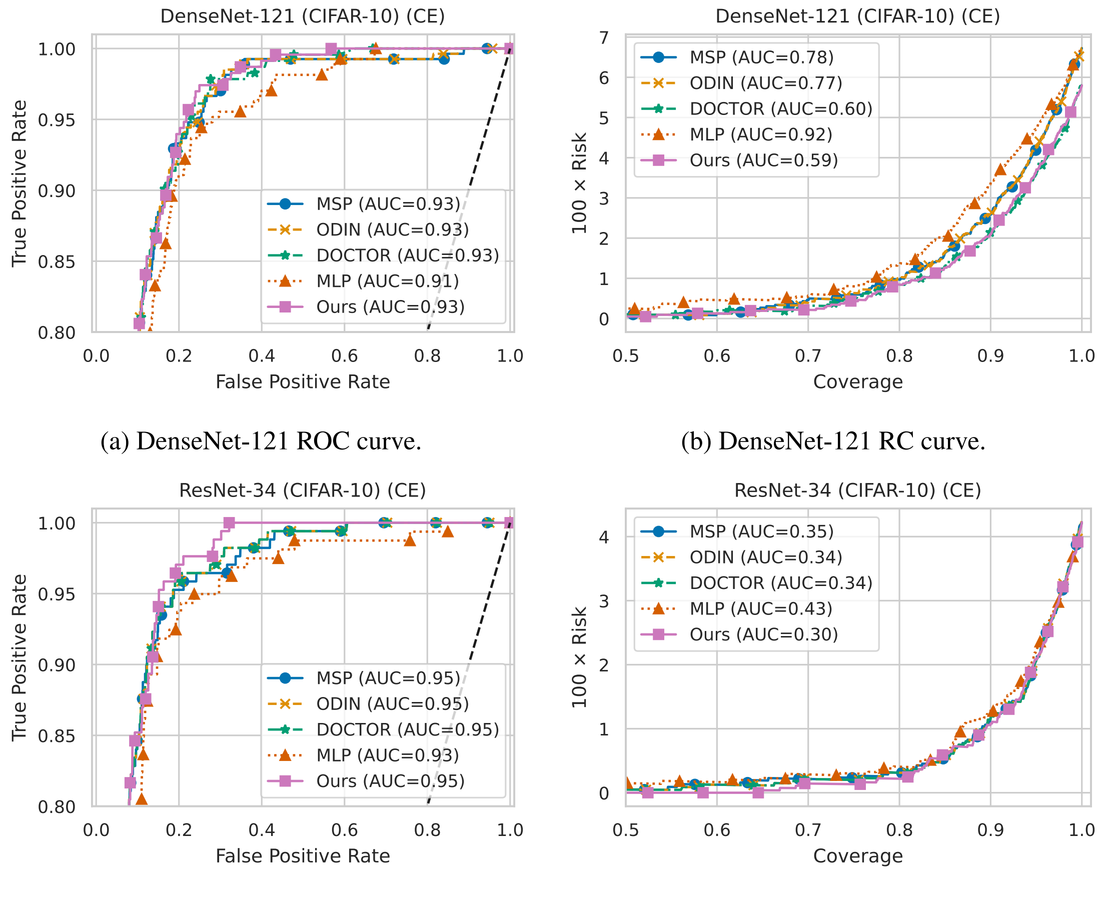
Bayesian methods. In this paragraph, we compare our method to additional uncertainty estimation methods, such as Deep Ensembles (Lakshminarayanan et al., 2017), MCDropout (Gal and Ghahramani, 2016), and a MLP directly trained on the validation data used to tune the relative uncertainty matrix. The results are available in Table 5.4.

Table 5.4: Misclassification detection results across two different architectures trained on CIFAR-10 and CIFAR-100 with CrossEntropy loss. We report the detection performance in terms of average FPR at 95% TPR (lower is better) in percentage with one standard deviation over ten different seeds in parenthesis.

Model	Dataset	MCDropout	Deep Ensembles	MLP	REL-U
DenseNet-121	CIFAR-10	30.3 (3.8)	25.5 (0.8)	37.3 (5.8)	**18.3** (0.2)
DenseNet-121	CIFAR-100	47.6 (1.2)	45.9 (0.7)	78.4 (1.4)	**41.5** (0.2)
ResNet-34	CIFAR-10	25.8 (4.9)	14.8 (1.4)	33.6 (2.7)	**14.1** (0.1)
ResNet-34	CIFAR-100	42.3 (1.0)	37.4 (1.9)	63.3 (1.0)	**32.7** (0.3)

ROC and Risk-Coverage curves. We also display the ROC and the risk-coverage curves for our main benchmark on models trained on CIFAR-10 with cross entropy loss. We observe that the performance of REL-U is comparable to other methods in terms of AUROC while outperforming them in high-TPR regions and reducing the risk of classification errors when abstention is desired (coverage) as observed in Fig. 5.7.

Performance of conformal prediction. We take into account the application of conformal predictors applied to the problem of misclassification. In particular, we consider the excellent work in Angelopoulos and Bates (2021), but most importantly Angelopoulos et al. (2021), which, in turn,



(a) DenseNet-121 ROC curve.

(b) DenseNet-121 RC curve.

(c) ResNet-34 ROC curve.

(d) ResNet-34 RC curve.

Figure 5.7: Equivalent performance of the detectors in terms of ROC demonstrating lower FPR for our method for high TPR regime. The risk and coverage (RC) curves also looks similar between methods, with a small advantage to our method in terms of AURC.

builds upon Romano et al. (2020). Conformal predictors, in stark contrast with standard prediction models, learn a “prediction set function”, i.e. they return a set of labels, which should contain the correct value with high probability, for a given data distribution. In particular, Angelopoulos et al. (2021) proposed a revision of Romano et al. (2020), with the main objective of preserving the guarantees of conformal prediction, while, at the same time, minimizing the prediction set cardinality on a sample basis: samples that are “harder” to classify can produce larger sets than samples that are easier to correctly classify. The models are “conformalized” (cf. Angelopoulos et al. (2021)) using the same validation samples, that are also available to the other methods. We reject the decision if the second largest probability within the prediction set exceeds a given threshold, as then, the prediction set would contain more than one label, indicating a possible misclassification event. The experiments are run on 2 models, 2 datasets and 3 training techniques for a total of 12 additional numerical results reported in Table 5.5. For the model trained with cross entropy in Table 5.5, the area under the ROC curve, averaged over 10 seeds, is $0.92\ (0.7)$ for the DenseNet-121 conformalized model on CIFAR-10, and $0.93\ (0.7)$ for the ResNet-34 conformalized model on CIFAR-10, showing comparable results w.r.t. the results in Figs. 5.7a and 5.7b.

Table 5.5: Misclassification detection results across two different architectures trained on CIFAR-10 and CIFAR-100 with five different training losses. We report the average accuracy of these models and the detection performance in terms of average FPR at 95% TPR (lower is better) in percent with one standard deviation over ten different seeds in parenthesis. The values for the conformalized models are reported in the right-most column.

Model	Training	Accuracy	MSP	ODIN	Doctor	Rel-U	Conf.
DenseNet-121 (CIFAR-10)	CrossEntropy	94.0	32.7 (4.7)	24.5 (0.7)	21.5 (0.2)	**18.3** (0.2)	31.6 (3.3)
	Mixup	95.1	54.1 (13.4)	38.8 (1.2)	**24.5** (1.9)	37.6 (0.9)	57.6 (6.9)
	RegMixUp	95.9	41.3 (8.0)	30.4 (0.4)	23.3 (0.4)	**22.0** (0.2)	30.3 (5.1)
DenseNet-121 (CIFAR-100)	CrossEntropy	73.8	45.1 (2.0)	41.7 (0.4)	**41.5** (0.2)	**41.5** (0.2)	46.5 (1.3)
	Mixup	77.5	48.7 (2.3)	41.4 (1.4)	**37.7** (0.6)	**37.7** (0.6)	47.0 (1.3)
	RegMixUp	78.4	49.7 (2.0)	45.5 (1.1)	43.3 (0.4)	**40.0** (0.2)	46.0 (1.3)
ResNet-34 (CIFAR-10)	CrossEntropy	95.4	25.8 (4.8)	19.4 (1.0)	14.3 (0.2)	**14.1** (0.1)	26.8 (4.6)
	Mixup	96.1	60.1 (10.7)	38.2 (2.0)	26.8 (0.6)	**19.0** (0.3)	58.1 (5.6)
	RegMixUp	97.1	34.0 (5.2)	26.7 (0.1)	21.8 (0.2)	**18.2** (0.2)	41.9 (7.0)
ResNet-34 (CIFAR-100)	CrossEntropy	79.0	42.9 (2.5)	38.3 (0.2)	34.9 (0.5)	**32.7** (0.3)	38.7 (1.5)
	Mixup	78.1	53.5 (6.3)	43.5 (1.6)	**37.5** (0.4)	**37.5** (0.3)	43.3 (0.9)
	RegMixUp	80.8	50.5 (2.8)	45.6 (0.9)	40.9 (0.8)	**37.7** (0.4)	47.7 (1.5)

CHAPTER 6

Conclusion and Perspectives

In conclusion, this thesis delves into the important field of the safety and trustworthiness of machine learning algorithms and their profound impact on the evolution of artificial intelligence. As AI applications extend their reach into diverse domains, including safety-critical areas like autonomous driving and healthcare, the need for reliability and trust in these systems takes center stage. Failures and unforeseen outcomes of AI systems cast shadows of doubt, and addressing these issues is paramount to restoring trust in automated systems.

Our primary mission has been to bolster the detection capabilities of machine learning models, empowering them to identify situations that deviate from the norm. To achieve this, we have developed detection methods for identifying out-of-distribution (OOD) samples and quantifying uncertainty in inlier predictions, which are critical aspects of AI safety. These contributions hold significant implications for enhancing the reliability and safety of AI systems, particularly in scenarios characterized by changing data distributions. While we do not anticipate adverse outcomes from our work, it is important to exercise caution when employing detection methods in critical domains, especially given the evolving threat landscape, where adversaries may actively exploit vulnerabilities to bypass safety measures.

To summarize, in the first chapter, we formalized the problems of OOD and misclassification detection, laying the groundwork for our research. In Chapter 2, We introduced a novel methodology based on the Fisher-Rao geodesic distance between distributions, unifying the formulation for the logits and features of the network. Furthermore, we tackled integrating information from pseudo-outliers by integrating a reference distribution. In Chapter 3, we relaxed any needs for supervision or hyperparameter tuning by designing a methodology that measures the similarity between the neural trajectory of a sample w.r.t. the training data distribution, effectively distinguishing OOD samples in a completely unsupervised manner. In Chapter 4, we showed how to effectively combine any detectors to obtain a more powerful and robust detector with a consolidated decision region grounded on solid statistical principles. Finally, in Chapter 5, we addressed misclassification detection and uncertainty estimation, deriving a data-driven approach to better separate positive and negative samples with a practical and convenient closed-form solution.

Moving forward, several opportunities for future work have emerged from this research. First and foremost, a focus on enhancing the interpretability and explainability of OOD detection methods is imperative. Ensuring that AI systems provide clear explanations for their decisions is essential, especially in applications with legal or ethical implications. Additionally, investigating ways to facilitate better collaboration between humans and AI systems is crucial, particularly in scenarios where AI encounters situations it is not equipped to handle. Designing systems that can effectively communicate their limitations to human operators can enhance trust and user experience. Collaboration with researchers from various disciplines, including psychology, philosophy, and ethics, can provide valuable insights into human-AI interaction and the broader societal implications of AI safety. Furthermore, extending the methods to accommodate multi-modal data, where information is derived from various sources such as text, images, and sensor data, is a promising avenue. Multi-modal AI systems are increasingly relevant in real-world applications and require robust safety measures.

In closing, by addressing these perspectives and challenges, the current work can contribute to the development of more trustworthy and resilient AI systems with a positive impact on various aspects of society and industry.

APPENDIX A

Appendix

A.1 Proof of Proposition 1

We recall the definition of the total variation distance when applied to distributions P, Q on a set $\mathcal{X} \subseteq \mathbb{R}^d$ and the Scheffé's identity (Scheffe, 1947, Lemma 2.1):

$$\|P - Q\|_{\text{TV}} \triangleq \sup_{\mathcal{A} \in \mathcal{B}^d} |P(\mathcal{A}) - Q(\mathcal{A})| = \frac{1}{2} \int |p_X(\boldsymbol{x}) - q_X(\boldsymbol{x})| d\mu(\boldsymbol{x}) \tag{A.1}$$

with respect to a base measure μ, where $\mathcal{B}^d$ denotes the class of all Borel sets on $\mathbb{R}^d$.

Proof. First of all, we prove the equality for $\gamma = 1$. Let us denote with $\mathcal{A}^\star \equiv \mathcal{A}(1, \mathcal{D}_n)$ and $\mathcal{A}^{\star c} \equiv \mathcal{A}^c(1, \mathcal{D}_n)$ the optimal decision regions. Let $\epsilon_0(\mathcal{A}^\star, \mathcal{D}_n)$ and $\epsilon_1(\mathcal{A}^{\star c}, \mathcal{D}_n)$ the Type-I and Type-II errors. Then,

$$\begin{aligned}
\epsilon_0(\mathcal{A}^\star, \mathcal{D}_n) + \epsilon_1(\mathcal{A}^{\star c}, \mathcal{D}_n) &= \int_{\mathcal{A}^\star} P_{U|Z}(\boldsymbol{du}|z=0; \mathcal{D}_n) + \int_{\mathcal{A}^{\star c}} P_{U|Z}(\boldsymbol{du}|z=1; \mathcal{D}_n) \\
&= \int_{\mathcal{A}^\star} \min\left\{P_{U|Z}(\boldsymbol{du}|z=0; \mathcal{D}_n), P_{U|Z}(\boldsymbol{du}|z=1; \mathcal{D}_n)\right\} \\
&+ \int_{\mathcal{A}^{\star c}} \min\left\{P_{U|Z}(\boldsymbol{du}|z=0; \mathcal{D}_n), P_{U|Z}(\boldsymbol{du}|z=1; \mathcal{D}_n)\right\} \\
&= \int_{\mathcal{U}} \min\left\{P_{U|Z}(\boldsymbol{du}|z=0; \mathcal{D}_n), P_{U|Z}(\boldsymbol{du}|z=1; \mathcal{D}_n)\right\} \\
&= 1 - \left\|P_{U|Z}(\cdot|z=1; \mathcal{D}_n) - P_{U|Z}(\cdot|z=0; \mathcal{D}_n)\right\|_{\text{TV}},
\end{aligned} \tag{A.2}$$

where the last identity follows by applying Scheffé's theorem (Scheffe, 1947, Lemma 2.1).

From the last identity in equation A.2 and any decision region $\mathcal{A} \subseteq \mathcal{U}$, we have

$$\begin{aligned}
1 - \big\| & P_{U|Z}(\cdot|z=1;\mathcal{D}_n) - P_{U|Z}(\cdot|z=0;\mathcal{D}_n)\big\|_{\text{TV}}, \\
&= \int_{\mathcal{U}} \min\big\{P_{U|Z}(\boldsymbol{du}|z=0;\mathcal{D}_n), P_{U|Z}(\boldsymbol{du}|z=1;\mathcal{D}_n)\big\} \\
&= \int_{\mathcal{A}} \min\big\{P_{U|Z}(\boldsymbol{du}|z=0;\mathcal{D}_n), P_{U|Z}(\boldsymbol{du}|z=1;\mathcal{D}_n)\big\} \\
&+ \int_{\mathcal{A}^c} \min\big\{P_{U|Z}(\boldsymbol{du}|z=0;\mathcal{D}_n), P_{U|Z}(\boldsymbol{du}|z=1;\mathcal{D}_n)\big\} \\
&\leq \int_{\mathcal{A}} P_{U|Z}(\boldsymbol{du}|z=0;\mathcal{D}_n) + \int_{\mathcal{A}^c} P_{U|Z}(\boldsymbol{du}|z=1;\mathcal{D}_n) \\
&= \epsilon_0(\mathcal{A},\mathcal{D}_n) + \epsilon_1(\mathcal{A}^c,\mathcal{D}_n).
\end{aligned}$$

It remains to show the last statement related to the Bayesian error of the test. Assume that $P_Z(1) = P_Z(0) = 1/2$. By using the last identity in equation A.2, we have

$$\begin{aligned}
\frac{1}{2}\Big[1 - \big\| & P_{U|Z}(\cdot|z=1;\mathcal{D}_n) - P_{U|Z}(\cdot|z=0;\mathcal{D}_n)\big\|_{\text{TV}}\Big] \\
&= \frac{1}{2}\int_{\mathcal{U}} \min\big\{P_{U|Z}(\boldsymbol{du}|z=0;\mathcal{D}_n), P_{U|Z}(\boldsymbol{du}|z=1;\mathcal{D}_n)\big\} \\
&= \int_{\mathcal{U}} \min\big\{P_{UZ}(\boldsymbol{du}, Z=0;\mathcal{D}_n), P_{UZ}(\boldsymbol{du}, Z=1;\mathcal{D}_n)\big\} \\
&= \mathbb{E}_U\left[\min\big\{P_{Z|U}(Z=0|\boldsymbol{U};\mathcal{D}_n), P_{Z|U}(Z=1|\boldsymbol{U};\mathcal{D}_n)\big\}\right] \\
&= \frac{1}{2}\big[\epsilon_0(\mathcal{A}^\star,\mathcal{D}_n) + \epsilon_1(\mathcal{A}^{\star c},\mathcal{D}_n)\big] \\
&= \inf_{\psi} \boldsymbol{P}_{\mathcal{D}_n}\left\{\psi(\boldsymbol{U}) \neq Z\right\},
\end{aligned}$$

where the last identity follow by the definition of the decision regions. □

A.2 Datasets Details

- **SVHN** (Netzer et al., 2011) dataset collects street house numbers for digit classification. It contains 73,257 training and 26,032 test RGB images of printed digits (from 0 to 9). Usually only the first 10,000 examples of the test set is used for evaluating the methods.

- **Tiny-ImageNet** (Le and Yang, 2015) dataset is a subset of the large-scale natural image dataset ImageNet (Deng et al., 2009). It contains 200 different classes and 10,000 examples.

- **LSUN** (Yu et al., 2015) dataset, which has 10,000 examples, is used for the large-scale scene classification of different scene categories (e.g., bedroom, bridge, kitchen, etc.).

- **iSUN** (Xu et al., 2015) dataset consists of selected natural scene images from the SUN (Xiao et al., 2010) dataset. The test set has 8925 images. This dataset is often used as a source of OOD for validation purposes as an independent dataset from the test OOD data.

- **Textures.** The Describable Textures Dataset (DTD) (Cimpoi et al., 2014) is a collection of textural pattern images observed in nature. It contains 47 categories totaling 5640 images.

- **Chars74K** dataset (de Campos et al., 2009) contains 74,000 samples of 62 classes of characters found in natural images, handwritten text, and synthesized from computer fonts. We take as OOD data only the *EnglishImg* dataset split, which contains 7705 characters from natural scenes.

- **Places365** (Zhou et al., 2017) contains images of 365 natural scenes categories. We used the small images validation split as OOD data in our experiments. It contains 36,500 RGB images.

- **Gaussian** dataset usually contains 10,000 synthetic RGB images generated from 2D Gaussian noise, where each RGB pixel is sampled from a Gaussian distribution with mean 0.5 and variance 1.0. The pixel values are clipped to the $[0, 1]$ interval. They should be easily detectable against natural images.

- **Uniform** dataset usually contains 10,000 synthetic RGB images generated from Uniform noise in the $[0, 1]$ interval. They should also be easily detectable against natural images.

For the large-scale benchmark, in addition to Textures and Places365, the following datasets are considering with the curated splits introduced by (Bitterwolf et al., 2023):

- **Species** (Hendrycks et al., 2022) is sourced from iNaturalist (Horn et al., 2017) and consists of 700,000 images from 1,316 species which were selected for not being in ImageNet-21K. They sort the species into 10 superclasses.

- **OpenImage-O** (Wang et al., 2022) consists of 17,632 images from the OpenImage-v3 (Krasin et al., 2017) which were manually annotated as being OOD w.r.t ImageNet-1K.

- **iNaturalist** (Huang and Li, 2021) split is composed of 10,000 test samples with concepts from 110 plant classes different from ImageNet-1K ones. The original dataset (Horn et al., 2017) cantains 859,000 images from more than 5,000 species of plants and animals.

- **Sun** (Huang and Li, 2021) is dataset with a split of 10,000 randomly sampled test examples belonging to 50 nature-related categories of the 397 categories and 130,519 images from the Xiao et al. (2010) scene dataset.

- **Semantic Shift Benchmark** proposed by Vaze et al. (2022) aims to capture the notion of semantic novelty by exploit the hierarchical, tree-like semantic structure of the ImageNet-21K (Ridnik et al., 2021) database that are disjoint. For each pair of classes between ImageNet-1K and ImageNet-21K, they define the semantic distance between two classes as the total path distance between their nodes in the semantic tree.

Bibliography

M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2021.05.008.

N. A. Ahuja, I. J. Ndiour, T. Kalyanpur, and O. Tickoo. Probabilistic modeling of deep features for out-of-distribution and adversarial detection. *Bayesian Deep Learning Workshop, NeurIPS*, 2019.

A. N. Angelopoulos and S. Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *CoRR*, abs/2107.07511, 2021.

A. N. Angelopoulos, S. Bates, M. I. Jordan, and J. Malik. Uncertainty sets for image classifiers using conformal prediction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

C. Atkinson and A. F. S. Mitchell. Rao's distance measure. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 43(3):345–365, 1981. ISSN 0581572X.

A. Birnbaum. Combining independent tests of significance. *Journal of the American Statistical Association*, 49(267):559–574, 1954. ISSN 01621459.

J. Bitterwolf, M. Mueller, and M. Hein. In or out? fixing imagenet out-of-distribution detection evaluation. In *International Conference on Machine Learning*, 2023.

B. Bolstad, R. Irizarry, M. Åstrand, and T. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19:185–193, 01 2003. doi: 10.1093/bioinformatics/19.2.185.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021.

M. B. Brown. 400: A method for combining non-independent, one-sided tests of significance. *Biometrics*, 31(4):987–992, 1975. ISSN 0006341X, 15410420.

N. Burkart and M. F. Huber. A survey on the explainability of supervised machine learning. *J. Artif. Int. Res.*, 70:245–317, may 2021. ISSN 1076-9757. doi: 10.1613/jair.1.12228.

J. Cen, D. Luan, S. Zhang, Y. Pei, Y. Zhang, D. Zhao, S. Shen, and Q. Chen. The devil is in the wrongly-classified samples: Towards unified open-set recognition. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay. Adversarial attacks and defences: A survey. *CoRR*, abs/1810.00069, 2018. URL `http://arxiv.org/abs/1810.00069`.

S. Choi and S.-Y. Chung. Novelty detection via blurring. In *International Conference on Learning Representations*, 2020.

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Trans. Inf. Theory*, 16(1): 41–46, 1970. doi: 10.1109/TIT.1970.1054406.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3606–3613, 2014. doi: 10.1109/CVPR.2014.461.

J. H. Clark. The geometry engine: A vlsi geometry system for graphics. *SIGGRAPH Comput. Graph.*, 16(3):127–133, jul 1982. ISSN 0097-8930. doi: 10.1145/965145.801272.

O. Cobb and A. V. Looveren. Context-aware drift detection. In *International Conference on Machine Learning*, 2022.

P. Colombo, E. Dadalto, G. Staerman, N. Noiry, and P. Piantanida. Beyond mahalanobis distance for textual ood detection. In *Advances in Neural Information Processing Systems*, 2022.

W. Conover and R. Iman. Rank transformations as a bridge between parametric and nonparametric statistics: Rejoinder. *American Statistician - AMER STATIST*, 35:124–129, 08 1981. doi: 10.1080/00031305.1981.10479327.

C. Corbière, N. Thome, A. Bar-Hen, M. Cord, and P. Pérez. Addressing failure prediction by learning model confidence. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2898–2909, 2019.

I. Csiszár. Eine informationstheoretische ungleichung und ihre anwendung auf den beweis der ergodizität von markoffschen ketten. *Magyer Tud. Akad. Mat. Kutato Int. Koezl.*, 8:85–108, 1964.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314, Dec. 1989. ISSN 0932-4194. doi: 10.1007/BF02551274.

E. Dadalto. Detectors: a python library for generalized out-of-distribution detection, 5 2023. URL `https://github.com/edadaltocg/detectors`.

E. Dadalto, F. Alberge, P. Duhamel, and P. Piantanida. Igeood: An information geometry approach to out-of-distribution detection. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

E. Dadalto, F. Alberge, P. Duhamel, and P. Piantanida. Igeood: An information geometry approach to out-of-distribution detection. In *International Conference on Learning Representations*, 2022.

E. Dadalto, F. Alberge, P. Duhamel, and P. Piantanida. Combine and conquer: A meta-analysis on data shift and out-of-distribution detection, 2023a.

E. Dadalto, P. Colombo, G. Staerman, N. Noiry, and P. Piantanida. Neural trajectories for out-of-distribution detection, 2023b.

E. Dadalto, M. Romanelli, F. Granese, and P. Piantanida. Trusting the untrustworthy: A cautionary tale on the pitfalls of training-based rejection option, 2024a.

E. Dadalto, M. Romanelli, G. Pichler, and P. Piantanida. A data-driven measure of relative uncertainty for misclassification detection. In *International Conference on Learning Representations*, 2024b.

M. Darrin, G. Staerman, E. Dadalto, J. C. Cheung, P. Piantanida, and P. Colombo. Unsupervised layer-wise score aggregation for textual ood detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

T. E. de Campos, B. R. Babu, and M. Varma. Character recognition in natural images. In *Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal*, February 2009.

M. de Carvalho. Mean, what do you mean? *The American Statistician*, 70(3):270–274, 2016. doi: 10.1080/00031305.2016.1148632.

R. Dechter. Learning while searching in constraint-satisfaction-problems. In *Proceedings of the Fifth AAAI National Conference on Artificial Intelligence*, AAAI'86, page 178–183. AAAI Press, 1986.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *2009 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

T. DeVries and G. W. Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

A. Djurisic, N. Bozanic, A. Ashok, and R. Liu. Extremely simple activation shaping for out-of-distribution detection. In *The Eleventh International Conference on Learning Representations*, 2023a.

A. Djurisic, N. Bozanic, A. Ashok, and R. Liu. Extremely simple activation shaping for out-of-distribution detection. In *The Eleventh International Conference on Learning Representations*, 2023b.

X. Dong, J. Guo, A. Li, W.-T. M. Ting, C. Liu, and H. T. Kung. Neural mean discrepancy for efficient out-of-distribution detection. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19195–19205, 2021.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

X. Du, G. Gozum, Y. Ming, and Y. Li. SIREN: Shaping representations for detecting out-of-distribution objects. In *Advances in Neural Information Processing Systems*, 2022a.

X. Du, Z. Wang, M. Cai, and S. Li. Towards unknown-aware learning with virtual outlier synthesis. In *International Conference on Learning Representations*, 2022b.

E. S. Edgington. An additive method for combining probability values from independent experiments. *The Journal of Psychology*, 80(2):351–363, 1972. doi: 10.1080/00223980.1972.9924813.

B.-S. Einbinder, Y. Romano, M. Sesia, and Y. Zhou. Training uncertainty-aware classifiers with conformalized deep learning. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

Z. Fang, Y. Li, J. Lu, J. Dong, B. Han, and F. Liu. Is out-of-distribution detection learnable? In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

R. Fisher. *Statistical methods for research workers*. Edinburgh Oliver & Boyd, 1925.

R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London, A*, 222:309–368, 1922.

E. Fix and J. L. Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review / Revue Internationale de Statistique*, 57(3):238–247, 1989. ISSN 03067734, 17515823.

S. Fort, J. Ren, and B. Lakshminarayanan. Exploring the limits of out-of-distribution detection. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

K. Fukushima. Visual feature extraction by a multilayered network of analog threshold elements. *IEEE Transactions on Systems Science and Cybernetics*, 5(4):322–333, 1969. doi: 10.1109/TSSC.1969.300225.

Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1050–1059. JMLR.org, 2016.

S. Gallón, J.-M. Loubes, and E. Maza. Statistical properties of the quantile normalization method for density curve alignment. *Mathematical Biosciences*, 242(2):129–142, 2013. ISSN 0025-5564. doi: https://doi.org/10.1016/j.mbs.2012.12.007.

J. a. Gama, I. Žliobaitundefined, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4), mar 2014. ISSN 0360-0300. doi: 10.1145/2523813.

Y. Geifman and R. El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4878–4887, 2017.

C. Geng, S.-J. Huang, and S. Chen. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3614–3631, oct 2021. doi: 10.1109/tpami.2020.2981604.

I. Gibbs and E. Candes. Adaptive conformal inference under distribution shift. In *Advances in Neural Information Processing Systems*, volume 34, pages 1660–1672. Curran Associates, Inc., 2021.

C. Gini. Variabilità e mutabilità; contributo allo studio delle distribuzioni e delle relazioni statistiche. In *[Fasc. I.]. Tipogr. Di P. Cuppini 1912.*, 1912.

G. V. Glass. Primary, secondary, and meta-analysis of research. *Educational Researcher*, 5(10):3–8, 1976. doi: 10.3102/0013189X005010003.

I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

F. Granese, M. Romanelli, D. Gorla, C. Palamidessi, and P. Piantanida. DOCTOR: A simple method for detecting misclassification errors. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 5669–5681, 2021.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 2017.

M. Haroush, T. Frostig, R. Heller, and D. Soudry. A statistical framework for efficient out of distribution detection in deep neural networks, 2021.

J. Hartung. A note on combining dependent tests of significance. *Biometrical Journal*, 41(7): 849–855, 1999. doi: https://doi.org/10.1002/(SICI)1521-4036(199911)41:7<849::AID-BIMJ849> 3.0.CO;2-T.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

N. A. Heard and P. Rubin-Delanchy. Choosing between methods of combining p-values. *Biometrika*, 105:239–246, 2017.

M. Hein, M. Andriushchenko, and J. Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 41–50, 2019.

D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations, 2019.

D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

D. Hendrycks, M. Mazeika, and T. Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019.

D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021.

D. Hendrycks, S. Basart, M. Mazeika, M. Mostajabi, J. Steinhardt, and D. X. Song. Scaling out-of-distribution detection for real-world settings. In *International Conference on Machine Learning*, 2022.

G. V. Horn, O. M. Aodha, Y. Song, A. Shepard, H. Adam, P. Perona, and S. J. Belongie. The inaturalist challenge 2017 dataset. *2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Netw.*, 2(5):359–366, jul 1989. ISSN 0893-6080.

A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

Y.-C. Hsu, Y. Shen, H. Jin, and Z. Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10948–10957, 2020.

G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017. doi: 10.1109/CVPR.2017.243.

R. Huang and Y. Li. Mos: Towards scaling out-of-distribution detection for large semantic space. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8706–8715, 2021.

R. Huang, A. Geng, and Y. Li. On the importance of gradients for detecting distributional shifts in the wild. *ArXiv*, abs/2110.00218, 2021.

A. Ivakhnenko, V. Lapa, and P. U. L. I. S. O. E. ENGINEERING. *Cybernetic Predicting Devices*. JPRS 37, 803. Joint Publications Research Service [available from the Clearinghouse for Federal Scientific and Technical Information], 1965.

J. Katz-Samuels, J. B. Nakhleh, R. D. Nowak, and Y. Li. Training ood detectors in their natural habitats. In *International Conference on Machine Learning*, 2022.

D. Kaur, S. Uslu, K. J. Rittichier, and A. Durresi. Trustworthy artificial intelligence: A review. *ACM Comput. Surv.*, 55(2), jan 2022. ISSN 0360-0300. doi: 10.1145/3491209.

A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5580–5590, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

P. Kirichenko, P. Izmailov, and A. G. Wilson. Why normalizing flows fail to detect out-of-distribution data. In *Advances in Neural Information Processing Systems*, volume 33, pages 20578–20589. Curran Associates, Inc., 2020.

A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby. Big transfer (bit): General visual representation learning. In *ECCV*, 2020.

N. Y. Kotelevskii, A. Artemenkov, K. Fedyanin, F. Noskov, A. Fishkov, A. Shelmanov, A. Vazhentsev, A. Petiushko, and M. Panov. Nonparametric uncertainty quantification for single deterministic neural network. In *Advances in Neural Information Processing Systems*, 2022.

I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, and K. Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. 2017.

A. Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, Mar. 1951. doi: 10.1214/AOMS/1177729694.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

M. D. Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis; Machine Intelligence*, 44(07):3366–3385, jul 2022. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3057446.

Y. Le and X. Yang. Tiny imagenet visual recognition challenge. 2015.

Y. LeCun and C. Cortes. MNIST handwritten digit database. 2010.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551, 1989.

Y. Lecun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. *A tutorial on energy-based learning*. MIT Press, 2006.

K. Lee, H. Lee, K. Lee, and J. Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*, 2018a.

K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems 31*, pages 7167–7177. Curran Associates, Inc., 2018b.

E. L. Lehmann and J. P. Romano. *Testing statistical hypotheses*. Springer Texts in Statistics. Springer, New York, third edition, 2005. ISBN 0-387-98864-5.

S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018a.

S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018b.

Z. Lin, S. D. Roy, and Y. Li. Mood: Multi-level out-of-distribution detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

S. Linnainmaa. Taylor expansion of the accumulated rounding error. *BIT*, 16(2):146–160, jun 1976. ISSN 0006-3835. doi: 10.1007/BF01931367.

Z. Lipton, Y.-X. Wang, and A. Smola. Detecting and correcting for label shift with black box predictors. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3122–3130. PMLR, 10–15 Jul 2018.

B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin. When machine learning meets privacy: A survey and outlook. *ACM Comput. Surv.*, 54(2), mar 2021. ISSN 0360-0300. doi: 10.1145/3436755.

W. Liu, X. Wang, J. Owens, and Y. Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 2020.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

D. J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4:448–472, 1992.

P. C. Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936.

P. Massart. The Tight Constant in the Dvoretzky-Kiefer-Wolfowitz Inequality. *The Annals of Probability*, 18(3):1269 – 1283, 1990. doi: 10.1214/aop/1176990746.

F. J. Massey. The kolmogorov-smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951. ISSN 01621459.

N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), jul 2021. ISSN 0360-0300. doi: 10.1145/3457607.

M. Minderer, J. Djolonga, R. Romijnders, F. A. Hubis, X. Zhai, N. Houlsby, D. Tran, and M. Lucic. Revisiting the calibration of modern neural networks. In *Advances in Neural Information Processing Systems*, 2021.

Y. Ming, Y. Sun, O. Dia, and Y. Li. How to exploit hyperspherical embeddings for out-of-distribution detection? In *The Eleventh International Conference on Learning Representations*, 2023.

F. Mosteller and R. A. Fisher. Questions and answers. *The American Statistician*, 2(5):30–31, 1948. ISSN 00031305.

J. Mukhoti, J. van Amersfoort, P. H. S. Torr, and Y. Gal. Deep deterministic uncertainty for semantic segmentation. In *CoRR*, volume abs/2111.00079, 2021.

J. Mukhoti, A. Kirsch, J. van Amersfoort, P. H. Torr, and Y. Gal. Deep deterministic uncertainty: A new simple baseline. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24384–24394, June 2023.

E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019.

H. Narasimhan, A. K. Menon, W. Jitkrittum, and S. Kumar. Plugin estimators for selective classification with out-of-distribution detection, 2023.

Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

J. Neyman and E. S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231:289–337, 1933. ISSN 02643952.

A. M. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *CoRR*, abs/1412.1897, 2014.

D. Opitz and R. Maclin. Popular ensemble methods: An empirical study. *J. Artif. Int. Res.*, 11(1): 169–198, jul 1999. ISSN 1076-9757.

G. Pang, C. Shen, L. Cao, and A. V. D. Hengel. Deep learning for anomaly detection: A review. *ACM Comput. Surv.*, 54(2), mar 2021. ISSN 0360-0300. doi: 10.1145/3439950.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

M. Pimentel, D. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 06 2014. doi: 10.1016/j.sigpro.2013.12.026.

J. Pinele, J. E. Strapasson, and S. I. R. Costa. The fisher–rao distance between multivariate normal distributions: Special cases, bounds and applications. *Entropy*, 22(4), 2020. ISSN 1099-4300. doi: 10.3390/e22040404.

F. Pinto, H. Yang, S.-N. Lim, P. Torr, and P. K. Dokania. Regmixup: Mixup as a regularizer can surprisingly improve accuracy and out distribution robustness. In *Advances in Neural Information Processing Systems*, 2022.

J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009. ISBN 0262170051.

S. Rabanser, S. Günnemann, and Z. C. Lipton. Failing loudly: An empirical study of methods for detecting dataset shift, 2019.

C. R. Rao. Diversity and dissimilarity coefficients: a unified approach. *Theoretical population biology*, 21(1):24–43, 1982.

J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon, and B. Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

A. Rényi. On measures of entropy and information. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*. Berkeley, California, USA, 1961.

T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik. Imagenet-21k pretraining for the masses. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021.

H. Robbins and S. Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 – 407, 1951. doi: 10.1214/aoms/1177729586.

Y. Romano, M. Sesia, and E. J. Candès. Classification with valid and adaptive coverage. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958. ISSN 0033-295X. doi: 10.1037/h0042519.

C. S. Sastry and S. Oore. Detecting out-of-distribution examples with Gram matrices. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8491–8501. PMLR, 13–18 Jul 2020.

H. Scheffe. A Useful Convergence Theorem for Probability Distributions. *The Annals of Mathematical Statistics*, 18(3):434 – 438, 1947. doi: 10.1214/aoms/1177730390.

T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *Information Processing in Medical Imaging*, pages 146–157, Cham, 2017. Springer International Publishing. ISBN 978-3-319-59050-9.

C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3): 379–423, July 1948. ISSN 0005-8580. doi: 10.1002/J.1538-7305.1948.TB01338.X.

R. J. Simes. An improved bonferroni procedure for multiple tests of significance. *Biometrika*, 73(3): 751–754, 1986. ISSN 00063444.

J. Snoek, Y. Ovadia, E. Fertig, B. Lakshminarayanan, S. Nowozin, D. Sculley, J. V. Dillon, J. Ren, and Z. Nado. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13969–13980, 2019.

Y. Song, N. Sebe, and W. Wang. Rankfeat: Rank-1 feature removal for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, 2022.

G. Staerman, P. Mozharovskyi, and S. Clémençon. Affine-invariant integrated rank-weighted depth: Definition, properties and finite sample analysis. *arXiv preprint arXiv:2106.11068*, 2021.

S. A. Stouffer, E. A. Suchman, L. C. Devinney, S. A. Star, and J. Williams, Robin M. *The American Soldier: Adjustment During Army Life*. Studies in Social Psychology in World War II. Princeton University Press, 1949.

J. E. Strapasson, J. Pinele, and S. I. R. Costa. Clustering using the fisher-rao distance. In *2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 1–5, 2016. doi: 10.1109/SAM.2016.7569717.

Y. Sun and Y. Li. Dice: Leveraging sparsification for out-of-distribution detection. In *European Conference on Computer Vision*, 2022.

Y. Sun, C. Guo, and Y. Li. React: Out-of-distribution detection with rectified activations. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021.

Y. Sun, Y. Ming, X. Zhu, and Y. Li. Out-of-distribution detection with deep nearest neighbors. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20827–20840. PMLR, 17–23 Jul 2022.

E. Techapanurak, M. Suganuma, and T. Okatani. Hyperparameter-free out-of-distribution detection using cosine similarity. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020.

J. J. Thiagarajan, R. Anirudh, V. Narayanaswamy, and P. timo Bremer. Single model uncertainty estimation via stochastic data centering. In *Advances in Neural Information Processing Systems*, 2022.

L. H. C. Tippett. The methods of statistics. *London: Williams and Norgate, Ltd*, 1931.

C. Tsallis. Possible generalization of boltzmann-gibbs statistics. *Journal of statistical physics*, 52: 479–487, 1988.

J. W. Tukey. Mathematics and the picturing of data. *Proceedings of the International Congress of Mathematicians, Vancouver, 1975*, 2:523–531, 1975a.

J. W. Tukey. Mathematics and the picturing of data. In R. James, editor, *Proceedings of the International Congress of Mathematicians*, volume 2, pages 523–531. Canadian Mathematical Congress, 1975b.

A. M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950. ISSN 00264423, 14602113.

G. Valiant and P. Valiant. An automatic inequality prover and instance optimal identity testing. *SIAM J. Comput.*, 46(1):429–455, 2017. doi: 10.1137/151002526.

L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, nov 1984. ISSN 0001-0782. doi: 10.1145/1968.1972.

V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995. ISBN 0-387-94559-8.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

S. Vaze, K. Han, A. Vedaldi, and A. Zisserman. Open-set recognition: A good closed-set classifier is all you need. In *International Conference on Learning Representations*, 2022.

S. Vernekar, A. Gaurav, V. Abdelzad, T. Denouden, R. Salay, and K. Czarnecki. Out-of-distribution detection in classifiers via generation. In *Neural Information Processing Systems (NeurIPS 2019), Safety and Robustness in Decision Making Workshop*, 12/2019 2019.

V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer-Verlag, Berlin, Heidelberg, 2005. ISBN 0387001522.

H. Wang, Z. Li, L. Feng, and W. Zhang. Vim: Out-of-distribution with virtual-logit matching. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4911–4920, 2022.

H. Wei, R. Xie, H. Cheng, L. Feng, B. An, and Y. Li. Mitigating neural network overconfidence with logit normalization. In *International Conference on Machine Learning*, 2022.

B. Wilkinson. A statistical consideration in psychological research. *Psychol. Bull.*, 48(2):156–158, Mar. 1951.

G. Xia and C.-S. Bouganis. Augmenting softmax information for selective classification with out-of-distribution data. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 1995–2012, December 2022.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 08 2017.

J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3485–3492, 2010. doi: 10.1109/CVPR.2010.5539970.

Z. Xiao, Q. Yan, and Y. Amit. Likelihood regret: An out-of-distribution detection score for variational auto-encoder. In *Advances in Neural Information Processing Systems*, volume 33, pages 20685–20696. Curran Associates, Inc., 2020.

P. Xu, K. A. Ehinger, Y. Zhang, A. Finkelstein, S. R. Kulkarni, and J. Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking, 2015.

F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

J. Zhang, J. Yang, P. Wang, H. Wang, Y. Lin, H. Zhang, Y. Sun, X. Du, K. Zhou, W. Zhang, Y. Li, Z. Liu, Y. Chen, and H. H. Li. Openood v1.5: Enhanced benchmark for out-of-distribution detection. *ArXiv*, abs/2306.09301, 2023.

B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

W. Zhou, F. Liu, and M. Chen. Contrastive out-of-distribution detection for pretrained transformers. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1100–1111, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.84.

F. Zhu, Z. Cheng, X.-Y. Zhang, and C.-L. Liu. Rethinking confidence calibration for failure prediction. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXV*, page 518–536, Berlin, Heidelberg, 2022a. Springer-Verlag. ISBN 978-3-031-19805-2. doi: 10.1007/978-3-031-19806-9_30.

F. Zhu, Z. Cheng, X.-Y. Zhang, and C.-L. Liu. Openmix: Exploring outlier samples for misclassification detection. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Y. Zhu, Y. Chen, C. Xie, X. Li, R. Zhang, H. Xue', X. Tian, bolun zheng, and Y. Chen. Boosting out-of-distribution detection with typical features. In *Advances in Neural Information Processing Systems*, 2022b.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences de l'Environnement d'Île-de-France (SEIF)

Titre: Amélioration de la Fiabilité de l'Intelligence Artificielle par la Détection des Données Hors Distribution et des Erreurs de Classification

Mots clés: Sécurité de l'IA, Détection de Données Hors Distribution, Estimation de l'Incertitude

Résumé: Cette thèse explore l'intersection cruciale entre l'apprentissage automatique (IA) et la sécurité, visant à résoudre les défis liés au déploiement de systèmes intelligents dans des scénarios réels. Malgré des progrès significatifs en IA, des préoccupations liées à la confidentialité, à l'équité et à la fiabilité ont émergé, incitant à renforcer la fiabilité des systèmes d'IA. L'objectif central de la thèse est de permettre aux algorithmes d'IA d'identifier les écarts par rapport au comportement normal, contribuant ainsi à la sécurité globale des systèmes intelligents.

La thèse commence par établir les concepts fondamentaux de la détection des données hors distribution (OOD) et de la détection des erreurs de classification dans le chapitre 1, fournissant une littérature essentielle et expliquant les principes clés. L'introduction souligne l'importance de traiter les problèmes liés au comportement non intentionnel et nuisible en IA, en particulier lorsque les systèmes d'IA produisent des résultats inattendus en raison de divers facteurs tels que des divergences dans les distributions de données.

Dans le chapitre 2, la thèse introduit une nouvelle méthode de détection de données hors distribution basée sur la distance géodésique Fisher-Rao entre les distributions de probabilité. Cette approche unifie la formulation des scores de détection pour les logits du réseau et les espaces latents, contribuant à une robustesse et une fiabilité accrues dans l'identification des échantillons en dehors de la distribution d'entraînement.

Le chapitre 3 présente une méthode de détection des données hors distribution non supervisée qui analyse les trajectoires neuronales sans nécessiter de supervision ou d'ajustement d'hyperparamètres. Cette méthode vise à identifier les trajectoires d'échantillons atypiques à travers diverses couches, améliorant l'adaptabilité des modèles d'IA à des scénarios divers.

Le chapitre 4 se concentre sur la consolidation et l'amélioration de la détection hors distribution en combinant efficacement plusieurs détecteurs. La thèse propose une méthode universelle pour combiner des détecteurs existants, transformant le problème en un test d'hypothèse multivarié et tirant parti d'outils de méta-analyse. Cette approche améliore la détection des changements de données, en en faisant un outil précieux pour la surveillance en temps réel des performances des modèles dans des environnements dynamiques et évolutifs.

Dans le chapitre 5, la thèse aborde la détection des erreurs de classification et l'estimation de l'incertitude par une approche axée sur les données, introduisant une solution pratique en forme fermée. La méthode quantifie l'incertitude par rapport à un observateur, distinguant entre prédictions confiantes et incertaines même face à des données difficiles. Cela contribue à une compréhension plus nuancée de la confiance du modèle et aide à signaler les prédictions nécessitant une intervention humaine.

La thèse se termine en discutant des perspectives futures et des orientations pour améliorer la sécurité en IA et en apprentissage automatique, soulignant l'évolution continue des systèmes d'IA vers une plus grande transparence, robustesse et fiabilité. Le travail collectif présenté dans la thèse représente une avancée significative dans le renforcement de la sécurité en IA, contribuant au développement de modèles d'apprentissage automatique plus fiables et dignes de confiance, capables de fonctionner efficacement dans des scénarios réels divers et dynamiques.

Title: Improving Artificial Intelligence Reliability through Out-of-Distribution and Misclassification Detection

Keywords: AI Safety, Out-of-Distribution Detection, Uncertainty Estimation

Abstract: This thesis explores the intersection of machine learning (ML) and safety, aiming to address challenges associated with the deployment of intelligent systems in real-world scenarios. Despite significant progress in ML, concerns related to privacy, fairness, and trustworthiness have emerged, prompting the need for enhancing the reliability of AI systems. The central focus of the thesis is to enable ML algorithms to detect deviations from normal behavior, thereby contributing to the overall safety of intelligent systems.

The thesis begins by establishing the foundational concepts of out-of-distribution (OOD) detection and misclassification detection in Chapter 1, providing essential background literature and explaining key principles. The introduction emphasizes the importance of addressing issues related to unintended and harmful behavior in ML, particularly when AI systems produce unexpected outcomes due to various factors such as mismatches in data distributions.

In Chapter 2, the thesis introduces a novel OOD detection method based on the Fisher-Rao geodesic distance between probability distributions. This approach unifies the formulation of detection scores for both network logits and feature spaces, contributing to improved robustness and reliability in identifying samples outside the training distribution.

Chapter 3 presents an unsupervised OOD detection method that analyzes neural trajectories without requiring supervision or hyperparameter tuning. This method aims to identify atypical sample trajectories through various layers, enhancing the adaptability of ML models to diverse scenarios.

Chapter 4 focuses on consolidating and enhancing OOD detection by combining multiple detectors effectively. It presents a universal method for ensembling existing detectors, transforming the problem into a multi-variate hypothesis test and leveraging meta-analysis tools. This approach improves data shift detection, making it a valuable tool for real-time model performance monitoring in dynamic and evolving environments.

In Chapter 5, the thesis addresses misclassification detection and uncertainty estimation through a data-driven approach, introducing a practical closed-form solution. The method quantifies uncertainty relative to an observer, distinguishing between confident and uncertain predictions even in the face of challenging or unfamiliar data. This contributes to a more nuanced understanding of the model's confidence and helps flag predictions requiring human intervention.

The thesis concludes by discussing future perspectives and directions for improving safety in ML and AI, emphasizing the ongoing evolution of AI systems towards greater transparency, robustness, and trustworthiness. The collective work presented in the thesis represents a significant step forward in advancing AI safety, contributing to the development of more reliable and trustworthy machine learning models that can operate effectively in diverse and dynamic real-world scenarios.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France