ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute for
Dynamic Systems and Control
IDSC
Institut für Dynamische Systeme
und Regelungstechnik

Victor Jimenez Rodriguez

Improved robustness of deep learning models through posterior agreement-based model selection

Master Thesis

Institute for Machine Learning
Swiss Federal Institute of Technology (ETH) Zurich

Supervision

Dr. Joao Borges de Sa Carvalho, Dr. Alessandro Torcinovich
Prof. Dr. Joachim M. Buhmann

September 2024

IDSC-XX-YY-ZZ

Preface

The research presented in this thesis was conducted within the Information and Science Engineering Group at the Institute for Machine Learning (ETH Zurich), during the period October 2023 - September 2024, under the supervision of Prof. Dr. Joachim M. Buhmann. The thesis was co-supervised at Universitat Politecnica de Catalunya by Prof. Dr. Alexandre Parera i Lluna.

Contents

Abstract **v**

Notation **vii**

1 Introduction **1**
1.1 The robustness challenge . . . 1
1.1.1 Adversarial setting . . . 2
1.1.2 Out-of-distribution setting . . . 4
1.2 Related work . . . 6
1.3 Objectives . . . 7

2 Theoretical background **9**
2.1 The learning framework . . . 9
2.2 Learning with neural networks . . . 10
2.2.1 Backpropagation and gradient descent . . . 10
2.2.2 Loss landscape and parameter space . . . 11
2.3 Posterior agreement . . . 11
2.3.1 Posterior distribution . . . 12
2.3.2 Generalization error . . . 13
2.3.3 Maximum posterior agreement . . . 14

3 Experimental setup **17**
3.1 Problem formulation . . . 17
3.1.1 The classification problem . . . 17
3.1.2 Robustness to covariate shift . . . 18
3.1.3 Adversarial setting . . . 20
3.1.4 Domain generalization setting . . . 22
3.2 Robustness enhancement . . . 22
3.3 Robustness assessment with posterior agreement . . . 23
3.3.1 Posterior in classification tasks . . . 23
3.3.2 The posterior agreement kernel . . . 25
3.3.3 Analytical example . . . 26
3.4 Posterior agreement beyond robustness . . . 26

4 Results and discussion **27**

5 Discussion **29**

A Supplementary material **31**
A.1 Proof of problem formulation . . . 31
A.2 Properties of the PA kernel . . . 33

B Again Something **37**

Abstract

Posterior Agreement (PA) has been proposed as a theoretically-grounded alternative for model robustness assessment in covariate shift settings. In this work, we provide further evidence in favor of this hypothesis and we explore the use of PA as a model selection criterion for deep learning models in supervised classification tasks.

Starting from the theoretical principles leading to PA, we derive a computationally-efficient approximation to its value in discrete hypothesis set problems, and we show that it is a valid alternative to cross-validation in the presence of distribution shifts. Additionally, we follow some threads from the original hypothesis and we extend the use of PA to some other specific settings in which a domain shift can be defined, such as subpopulation (unbalanced) settings, mislabelled datasets and data augmentation strategies.

REWRITE AT THE END

Notation

Symbols

EHC	Conditional equation	[−]
e	Willans coefficient	[−]
F, G	Parts of the system equation	[K/s]

Indicies

a	Ambient
air	Air

Acronyms and Abbreviations

NEDC	New European Driving Cycle
ETH	Eidgenössische Technische Hochschule

Chapter 1

Introduction

This chapter aims to set the stage for the detailed analysis and discussion that will follow, by providing a general overview of the problem of model robustness in machine learning and the current approaches to address it.

1.1 The robustness challenge

Robustness is defined as the ability of a model to maintain its predictive power on unseen observations that present some kind or transformation or variation. This work will provide experimental results for the principal sources of variability that are relevant in the context of image classification, namely sampling randomness, adversarial attacks, and out-of-distribution generalization.

Out of these three, only sampling randomness is commonly accounted for, in the sense that model selection and benchmarking are conducted using randomized subsets of unseen observations. In this way, the most generalizable features, and in turn the most generalizable models, are naturally selected. As it will be outlined in this chapter, this approach presents fundamental limitations that are rooted in the very nature of deep learning models and the data from which they learn.

First, the operative principles of neural networks make them vulnerable to small perturbations in the input space, often unnoticeable to humans, that lead to high-confidence incorrect predictions. This issue is commonly known as adversarial vulnerability, and an ongoing arms race incentivizes the design of new ways of perturbing models and new ways of defending them against such attacks. Strategies that foster robustness to adversarial attacks are possible, but come at a price of hindering conventional generalization to sampling randomness in the original data.

Second, the nature of the data used for training and selecting models is known to influence heavily the features that the model will learn to be the most predictive. Lack of representativity of certain aspects of the data and the presence of spurious correlations can lead to models that generalize well to sampling randomness within the same dataset but that fail to do so when those accidental relationships are not present, which is known as an out-of-distribution setting.

At the core of the robustness challenge lies the poor understanding of how models construct their inductive bias and the nature of the transformations between the space of weights and the space of functions that they are able to represent [20]. Features learned by the optimal standard classifier can be completely different from those learned by a robust classifier, regardless of the amount of data provided, which results in a fundamental limitation of standard performance in robust models [39, 51]. Besides, the feature space that deep learning models navigate is fundamentally different than that in which humans implicitly rely on, and we should therefore not expect models to be invariant to the same features humans are naturally invariant to [19].

(a) Original (b) Sampling (c) Adversarial (d) Domain

Figure 1.1: Illustrative example of three sources of variability mentioned. A pre-trained MobileNetV2 architecture is shown to be vulnerable to adversarial perturbations as the one represented in (c), and also to domain shifts as the one illustrated in (d), possibly because its inductive bias is influenced by the spurious correlation between cows and their natural background.

This thesis will encompass both phenomena under the same theoretical framework, and devise a common approach to the measurement of the distribution shift entailed by both adversarial and domain variability. Robustness will be characterized from the space of outcomes of the model, by means of a (posterior) probability distribution that will rank models and methods according to the agreement in their predictions when subject to different noise instantiations.

1.1.1 Adversarial setting

As it was already mentioned, certain perturbations of original test images, which can be almost imperceptible to the human eye, can lead to high-confidence incorrect predictions by deep neural networks, even when their standard performance metrics are high. Adversarial examples have been shown to transfer across architectures and training procedures, and even across subsets of data, often yielding the same incorrect prediction in all of these cases [37].

These intriguing phenomena were initially hypothesized to arise from a lack of smoothness over the input space, a property commonly assumed in other learners, deriving from their non-linear nature. Nevertheless, extensive research on the field has ellucidated that the root cause is instead the linearity of its learning units, which makes them vulnerable in certain directions of high-dimensional spaces where small effects can add up to significally change the outcome [15].

Following this intuition, several attacks have been proposed to evaluate the robustness of models to adversarial examples by finding vulnerable directions and adjusting the perturbation to have the desired misleading effect. Adversarial examples generated by these attacks can be used to train robust models via regularization, pushing generalization to those features present in the worst-case bounded perturbations and thus selecting models insensitivized to them.

Nevertheless, adversarial learning entails decision boundaries that are more complex than the ones derived via standard training (see Figure 1.2), intuitively demanding more data and more complex architectures, at the risk of overfitting to adversarial examples themselves [34]. These limitations express a fundamental trade-off that arises from an intrinsic difference between robust and non-robust features [39, 51].

Features selected by standard training are the most predictive towards generalization to sampling randomness within the same dataset, but they do not necessarily represent the features implicitly selected by humans and are not invariant to a human-based notion of similarity. Instead, features selected via adversarial training have been shown to represent this invariance, and thus align much better with human perception (see Figure 1.3) [19].

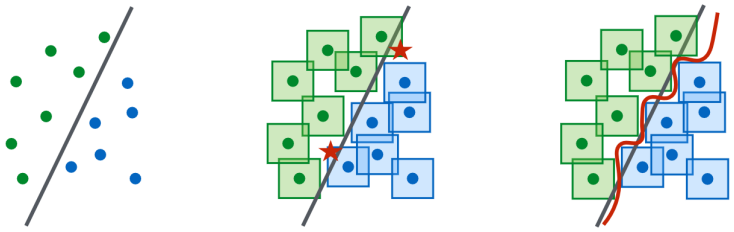
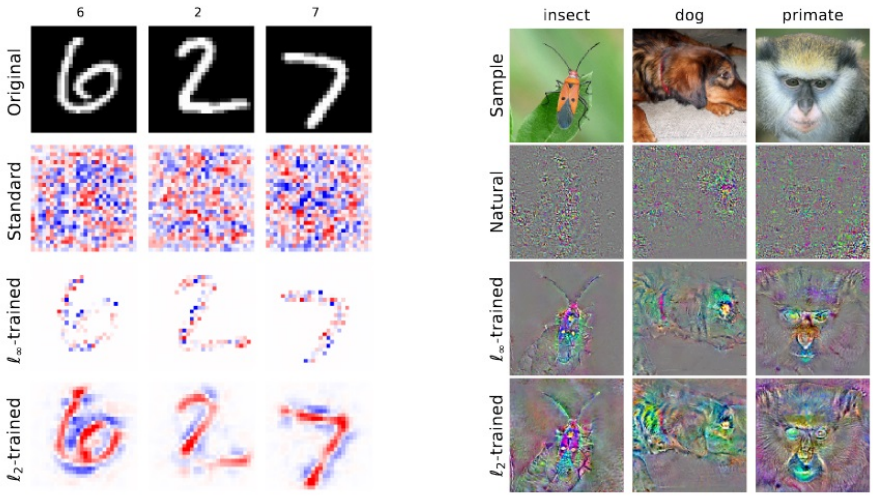


Figure 1.2: A conceptual illustration of standard vs. adversarial decision boundaries. (**left**) A set of linearly-separable points. (**middle**) Decision boundary learned via standard training. (**right**) Decision boundary learned via adversarial training. Both methods achieve zero training error, but only the robust model is able to generalize to ℓ_∞ perturbations. Source: [27]



(a) MNIST (b) Restricted ImageNet

Figure 1.3: Scaled loss gradient with respect to input images. Input pixels yielding the most predictive power are aligned with perceptually relavant features for the case of adversarial models, while appearing completely random in the case of standard models. Source: [39]

Furthermore, adversarial perturbations of robust models have been shown to display salient characteristics; that is, their features are perceived to belong to the class they are "misclassified" to, as it is illutrateed in Figure 1.4.

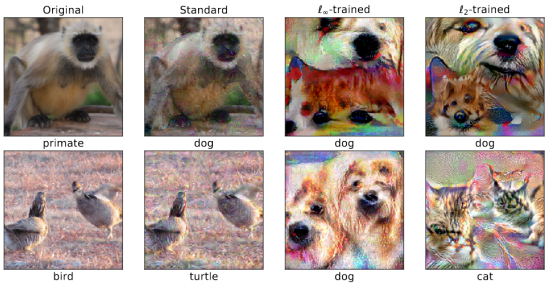


Figure 1.4: Adversarial examples for standard and PGD-trained models. Perturbed images produced for robust models effectively capture salient data characteristics and appear similar to examples of a different class. Source: [39]

Overall, these and other findings suggest that robustness in the adversarial setting is a fundamental property of data rather than models, and the phenomenon of transferability can be explained in these terms. Training strategies that are able to navigate the robustness-generalization trade-off will be the ones providing the best results, provided that the data distribution is representative of the true underlying features.

1.1.2 Out-of-distribution setting

Most learning algorithms work under the fundamental assumption that a causal relationship exists between input and output spaces and the target function to learn represents that causality and thus remains invariant regardless of the available data, implying that suitable approximations of this function can be obtained when data samples are independent and identically distributed in the input space [29, 33]. Nevertheless, this is not always the case, as often real-world data does not match the same statistical patterns as the data used for training, which defines a distribution shift that leads to poor generalization performance [53, 42, 26].

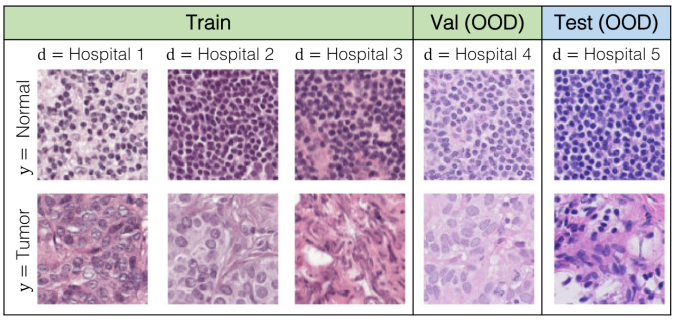


Figure 1.5: The `camelyon17` (WILDS) dataset comprises tissue patches from different hospitals. The goal is to accurately predict the presence of tumor tissue in patches taken from hospitals that are not in the training set. Source: [22]

A distribution shift can arise for various reasons, namely the unfeasibility of collecting diverse enough data, the lack of representativity of certain features, the changing or time-dependent nature of the data and also the implicit bias induced in the data collection process. This last point is particularly relevant, as it can serve as a generalization of all the previous cases and raise epistemological questions about the learning framework itself. For instance, Figure 1.6 refers to a cross-generalization analysis in which popular machine learning datasets were shown to be biased towards specific representation of features. Considering the fact that all data is sampled from the same source (i.e. internet), numerous human-induced biases are shown to determine the nature of representations, the most significant of all being negative bias, which arises when the negative subset[1] of the dataset is not representative of the input subspace excluding that particular class and results in a model that performs significantly worse in other datasets, even when trained with the same observations of that class.

Several approaches can be taken to address this issue, depending on the nature of the distribution shift and the access to its causal structure (see Figure 3 and Table 2 in [42]). Nevertheless, the common goal is to push the model towards domain-invariant representations that foster robustness in the face of distribution shifts, sometimes relaxing the causality condition to an assumption of invariance or stability of the distribution in the output space [42, 26].

[1] When certain observations in a dataset are labelled as belonging to a specific class, the remaining observations are implicitly assigned to not belong to that class, and therefore define a negative set in the model feature space.

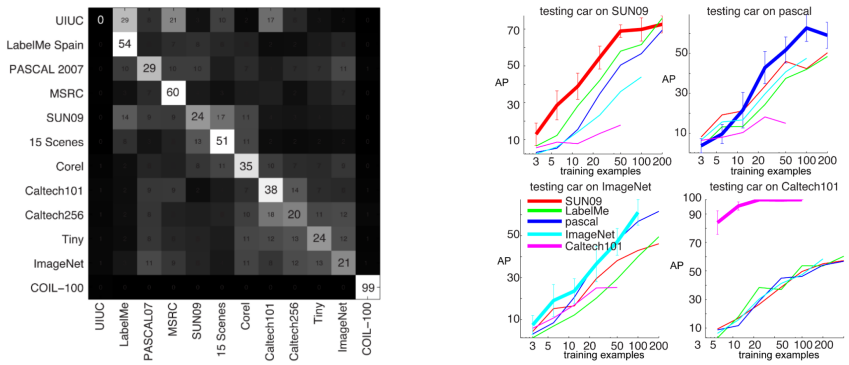


Figure 1.6: **(left)** Confusion matrix associated with a dataset identification task. There is a clearly pronounced diagonal, which indicates that each dataset posesses unique traits that make it distinguishable from the rest. **(right)** Cross-dataset generalization for "car" detection as function of training data. The vertical gap between two curves represents the decrease in performance resulting from training on a different dataset, and horizontal shift corresponds to the increase in amount of data needed to reach the same level of performance. Source: [38]

In general, every formulation considers a set of source domains encompassing data that is available for the training of the model, including any validation subsets used for model selection, regularization, or other hyperparameter tuning, and a set of target domains encompassing unseen data on which model performance will be evaluated. Within this framework, a straightforward approach to improving robustness is to directly sample target domains and adjust feature representations to be invariant between both, which is known as domain adaptation.

In this work we will focus instead on domain generalization, which refers to the case in which sampling from target domains is not feasible and feature invariance can be only enforced from the source [4]. In particular, two strategies will be considered, namely domain alignment and data augmentation/generation.

On the one hand, domain alignment stems from the output stability condition, and can be formulated as a regularization problem that pushes towards the minimization of the dissimilarity of feature representations originated from different source environments. The feature space in which the alignment is performed (e.g. kernel latent space [29], adversarial [31] or model-based [1]) and the similarity metric will determine the the particularities of the method [35, 25].

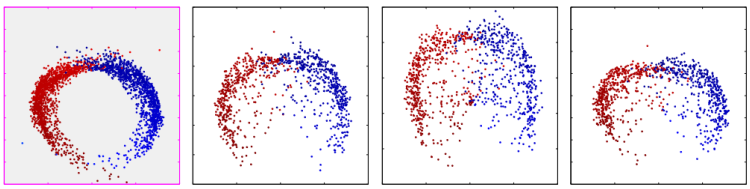


Figure 1.7: Projections of a binary synthetic dataset in the two principal DICA dimensions. The shaded box depicts the projection of training data, whereas the unshaded boxes show projections of unseen test datasets. Source: [29]

On the other hand, data augmentation/generation strategies do not need to assume output stability and instead achieve cross-domain generalization by generating new samples that diversify the original dataset with the hope of capturing the underlying causal structure of the problem. Augmented samples can either be randomizations of original observations (e.g. transformations such as rescaling or rotations) or new samples filling the distribution gaps between domains.

(a) Original (b) Mixup [52] (c) CutMix [50]

Figure 1.8: Mixup and Cutmix strategies can be used to interpolate between different labels and/or domains by generating intermediate observations. Source: [50]

Unlike in the adversarial setting, there is no common way of measuring the shift in distribution betweeen source domains, and current approaches are often constrained to specific datasets or training strategies. Robustness is instead quantified during (cross-)validation, either by reserving a subset of each domain, leaving one domain out, or by directly accessing target domains if they are available, which is known as the oracle approach. This last strategy is often used to provide an upper bound estimate of model robustness, as it usually provides over-confident performance estimates [53]. Numerous benchmark datasets, some of which will be considered in this work, are the current standard for robustness assessment even with the limitations they present [22].

1.2 Related work

In the adversarial front, early work [37] unveiled the nature of the susceptibility of deep learning models to adversarial examples and FGSM [15] was introduced as an intuitive approach for model regularization. Since then, several gradient-based methods have been proven to enhance adversarial robustness, such as PGD [27], C&W [9], FMN [32] and many others (see [24] for reference). All of them ultimately entail a strategy to find a vulnerable direction and adjust the perturbation (e.g. minimum-norm, maximum-confidence, etc.) based on the location of the decision boundary, either via soft constraints (i.e. regularization), boundary attacks or gradient projections [3].

In general, the primary distinction among adversarial attacks lies in their knowledge of the model's architecture and parameters. In that sense, white-box and black-box attacks can be distinguished, where the former have full access to the model and the latter only to the model's predictions. In black-box settings the loss gradient is unknown and other strategies such as score-based or decision-based attacks are used [24]. Regarding adversarial training (i.e. defenses), robustness can be achieved by a variety of methods, such as ensemble learning, defensive distillation, generative adversarial networks [47, 28], diffusion models [45, 18] and adaptive-boundary methods [12]. In this project, the Robustbench benchmark attacks [13] computed in the `CIFAR10` dataset will will be used as a standard for adversarial robustness evaluation.

In the domain generalization front, the existing rich taxonomy of methods can be classified into three main groups, namely data manipulation, representation learning and alternative learning strategies [42, 53, 26]. Data manipulation strategies refer to augmentation and generation, as for example randomization or adversarial augmentation [48, 52, 50]. Representation learning strategies are primarily divided into domain-invariant methods (e.g. IRM [1] or kernel-based [29, 2]) and feature disentanglement methods, which encompass causality-inspired approaches and general multi-component analysis. Other learning strategies include meta-learning [23, 41], ensemble learning or self-supervised learning.

Regarding robustness characterization, a wide range of metrics have been conceived (see [17] for reference), but accuracy-based criteria are still the most common. Alternatively, some theoretically-grounded approaches have been proposed, such as CLEVER [46], ACTS [43] or PA [7], which is the one we will explore in this work. In general, robustness is often reported and compared using robustness benchmark datasets. Some of the most relevant for image classification tasks are MNIST (and its multiple variations, such as DiagVib-6 [14]), PACS [49], VLCS [21] or WILDS [22].

1.3 Objectives

WRITE AT THE END

The main objective of this thesis is thus to assess the suitability of this framework in the context of deep learning model robustness in image classification tasks. For that, an operative version of posterior agreement will be derived, and an efficient implementation of its computation will be used as a metric to evaluate and select models based on the robustness of their response to different sources and levels of variability. The results of this work will be compared with the current state-of-the-art in robustness evaluation, namely robustbench [**?**] and WILDS [22] benckmarks in the adversarial and out-of-distribution settings, respectively, and an overall analysis of the use of the metric as an early-stopping criterion will be provided.

- Lead to the derivative work that will be presented in the thesis.=> benchmarking
- Lead to derivative but next level (more useful, current interest...) work => model selection
- Lead to non-derivative (i.e. probably unsuccessful) work => model selection beyond robustness
- Outline the structure of the thesis in terms of hypothesis.

Chapter 2

Theoretical background

2.1 The learning framework

Statistical learning theory encompasses the mathematical framework used to study generalization in machine learning [30, 20]. In this formalism, the goal is to learn a target function $f^* : \mathcal{X} \longmapsto \mathcal{Y}$ by means of an approximated function $f \in \mathcal{F}$ using a finite set of observations.

Definition (Supervised dataset)**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be the input and output spaces of the target function f^*, respectively. Let X be a random variable associated with a measure of probability in $\mathcal{X}$, and let $\underline{X} = (X_1, ..., X_N) \overset{iid}{\sim} X$ be a N-sized (simple) random sample of X [10]. A supervised dataset D is a realization $\boldsymbol{x} \sim \underline{X}$ paired with its output values under the target function mapping.*

$$D = \{(x_n, f^*(x_n))\}_{n\in[N]} = \{(x_n, y_n)\}_{n\in[N]}$$

$\mathcal{D}$ will represent the class of supervised datasets generated from $\underline{X}$.

The quality of the approximation can be measured with the expected risk $\mathcal{R}(f)$

$$\mathcal{R}(f) = \mathbb{E}_X[\mathcal{L}(f(x), f^\star(x))]$$

where $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ denotes a loss function. Glivenko-Cantelli theorem allow us to estimate the expected risk with its empirical (plug-in) analogous when N is large enough CITE1.

Definition (Empirical risk)**.** *Let D and $\mathcal{L}$ be the dataset and loss function of our problem, respectively. The empirical risk of $f \in \mathcal{F}$ computed on D is defined as*

$$\hat{\mathcal{R}}_D(f) = \frac{1}{N}\sum_{n=1}^{N} \mathcal{L}(f(x_n), f^\star(x_n))$$

Training, therefore, amounts to minimizing the empirical risk over the function class $\mathcal{F}$.

$$\text{ERM}_D = \hat{f}_D = \min_{f\in\mathcal{F}} \hat{\mathcal{R}}_D(f)$$

The best learning algorithm will be that with the lowest generalization error under sampling randomness, which is estimated as the empirical risk on a different dataset $D' \in \mathcal{D}$.

It can be shown that generalization error is ultimately linked to the complexity of the function class. The definition of complexity depends on the nature of the problem, but intuitively measures the cardinality of the subset of $\mathcal{F}$ that the algorithm is able to represent. A complex or high-capacity algorithm will be able to represent a larger subset of $\mathcal{F}$ and achieve a low empirical error, but will be also prone to overfitting to the specific learning realization thus yielding a higher generalization error [30].

As a general principle, the inductive bias of the algorithm (i.e. the set of constraints imposed on $\mathcal{F}$ during learning) should be aligned with that of our target function [20]. Given that more expressive classes are always preferred by optimization algorithms, the ERM objective function is tweaqued to include a regularization term penalizing complexity.

Definition (Regularized empirical risk)**.** *Let $\Omega : \mathcal{F} \to \mathbb{R}$ be a measure of complexity. The regularized empirical risk of a function f computed on D is defined as*

$$\hat{\mathcal{R}}_\Omega(f) = \hat{\mathcal{R}}(f) + \lambda\Omega(f)$$

where $\lambda \in \mathbb{R}$ controls the trade-off between empirical risk and generalization error.

2.2 Learning with neural networks

Neural networks are biologically-inspired machine learning models that consist of a set of nodes (neurons) organized in layers and connected by weighted edges (synapses). Figure 2.1 illustrates the transformation performed within a single node [36, 30, 40].

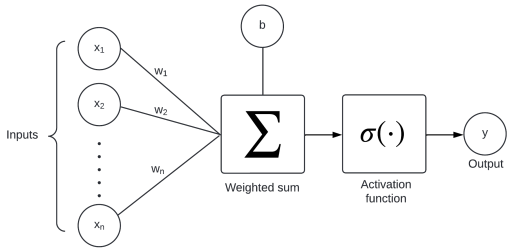


Figure 2.1: The output of a node is computed by applying a non-linear activation function $\sigma(\cdot)$ to the weighted sum of its inputs.

Let $\boldsymbol{x}_k \in \mathbb{R}^{d_k}$ be the input to the layer $k \leq L$, and let $\boldsymbol{W} \in \mathbb{R}^{d_k \times d_{k+1}}$ be the k-th weight matrix. The output of the layer can be expressed as

$$\boldsymbol{x}_{k+1} = \sigma_k(\boldsymbol{z}_{k+1}) = \sigma_k(\boldsymbol{W}_k^T \boldsymbol{x}_k + \boldsymbol{b}_k)$$

where σ_k is the non-linear activation function at layer k. We can therefore express the overall transformation of a neural network as the composition of its layers.

$$f_{\text{NN}}(\boldsymbol{x}) = \bigcirc_{k=0}^{L-1} \sigma_k(\boldsymbol{W}_k^T \boldsymbol{x} + \boldsymbol{b}_k) = f(\boldsymbol{x}; \gamma)$$

where γ represents the set of parameters of the network. In order to solve the learning problem, the optimization algorithm must navigate the non-convex loss landscape towards the minimum of the empirical risk. This is computationally achieved by means of gradient-descent-based optimizers, which compute the gradient over the parameters by means of the backpropagation algorithm.

2.2.1 Backpropagation and gradient descent

Let $w_{ji}^{(k)}$ be the weight from node j on layer $k-1$ to node i on layer k. Let $a_i^{(k-1)}$ be the output of node i on layer $k-1$ and let $z_j^{(k)} = \sum_{i=0}^{n_k - 1} w_{ji}^{(k)} a_i^{(k-1)} + b_k^{(k)}$ be the linear input of node j on layer k, so that $a_j^{(k)} = \sigma_j(z_j^{(k)})$ is the ouput from node j. We can compute the gradient of the loss $\mathcal{L}$ with respect to the weights by means of the chain rule as follows:

$$\frac{\partial \mathcal{L}}{\partial w_{ji}^{(k)}} = \frac{\partial \mathcal{L}}{\partial a_j^{(k)}} \frac{\partial a_j^{(k)}}{\partial z_j^{(k)}} \frac{\partial z_j^{(k)}}{\partial w_{ji}^{(k)}} = \frac{\partial \mathcal{L}}{\partial a_j^{(k)}} \frac{\partial \sigma_j^{(k)}}{\partial z_j^{(k)}} a_i^{(k-1)}$$

Given that the loss is computed as a function of the output of the network, all the edges from node i of layer $k-1$ influence the loss value at that node:

$$\frac{\partial \mathcal{L}}{\partial a_i^{(k-1)}} = \sum_{j=0}^{n_k-1} \frac{\partial \mathcal{L}}{\partial a_j^{(k)}} \frac{\partial a_j^{(k)}}{\partial z_j^{(k)}} \frac{\partial z_j^{(k)}}{\partial a_i^{(k-1)}} = \sum_{j=0}^{n_k-1} \frac{\partial \mathcal{L}}{\partial a_j^{(k)}} \frac{\partial \sigma_j^{(k)}}{\partial z_j^{(k)}} w_{ji}^{(k)}$$

All in all, we see that the same terms are required in different nodes to compute the gradient, making backpropagation algorithm very efficient. Equivalently, for the bias term:

$$\frac{\partial \mathcal{L}}{\partial b_j^{(k)}} = \frac{\partial \mathcal{L}}{\partial a_j^{(k)}} \frac{\partial a_j^{(k)}}{\partial z_j^{(k)}} \frac{\partial z_j^{(k)}}{\partial b_j^{(k)}} = \frac{\partial \mathcal{L}}{\partial a_j^{(k)}} \frac{\partial \sigma_j^{(k)}}{\partial z_j^{(k)}}$$

These derivatives are the components of the gradient vector that will be used to update the weights and biases of the network.

$$w_{ji}^{(k)} = w_{ji}^{(k)} - \eta \frac{\partial \mathcal{L}}{\partial w_{ji}^{(k)}}$$

$$b_j^{(k)} = b_j^{(k)} - \eta \frac{\partial \mathcal{L}}{\partial b_j^{(k)}}$$

where η is the learning rate. More efficient variations of gradient descent such as stochastic gradient descent or Adam are used in practice.

2.2.2 Loss landscape and parameter space

A neural network architecture NN can be expressed as a parametrization of the function space $\mathcal{F}$.

$$\begin{aligned} \mathrm{NN} : \Gamma \subseteq \mathbb{R}^S &\longmapsto \mathcal{F}_\Gamma \\ \gamma &\longmapsto f(\boldsymbol{x}; \gamma) = f_{\mathrm{NN}}(\boldsymbol{x}) \end{aligned}$$

where Γ is the parameter space associated with this particular architecture. The functional landscape $\mathcal{F}_\Gamma$ consists of all mappings $f(\gamma) : \mathcal{X} \longmapsto \mathcal{Y}$ that that can be realized by some parameter configuration $\gamma \in \Gamma$.

Universal approximation theorems state that arbitrarily wide or abitrarily deep architectures are able to represent virtually any function, but it is an open challenge to theoretically describe which complexity measure regulates generalization. A possible approach to this problem is to study the geometry of the loss landscape, especially in the vicinity of local minima. For instance, connected flat minima are often linked to better generalization capabilities, as they intuitively represent a robust region in the parametrization space and should be preferred over sharp minima [20].

In this work we will explore a different approach to the generalization problem, based on a definition of the generalization error that relies on the implicit randomness of the data sampling process.

2.3 Posterior agreement

As it was mentioned in the first lines of this chapter, the input of learning algorithms are datasets containing samples of the random variable X with support $\mathcal{X}$. The implicit randomness embedded in the sampling process extends to the outcome of algorithms, even when performing a deterministic set of operations [5]. An alternative intuition of generalization arises from this perspective, in the sense that a good algorithm should be expected to learn the same function when trained on different realizations of the same experiment; that is, when datasets are drawn from the same distribution but entail different instantiations of the noise associated with the sampling process.

A regularization principle is derived from this intuition and can be formalized as a generalization-complexity trade-off by defining generalization as the robustness or stability of the learned function to sampling noise. A suitable measure of complexity in this framework is the informativeness of the function, which represents its ability to learn the patterns in the data while filtering out the noise. The more expressive (i.e. complex) a function class is, the higher will be the estimated information content of the data. If the information content is overestimated, the approximated function will overfit to the noise and thus not generalize to different realizations of the experiment [11, 8, 6].

The robustness-informativeness regularization principle can be enforced from the set of outputs of the learned model. This section will formalize this principle and derive an operative model selection criterion.

Definition (Distribution of $\underline{\mathrm{X}}$)**.** *The simple random sample $\underline{X} \overset{iid}{\sim} X$ has a probability distribution described by the density function $f_{\underline{X}}$.*

$$f_{\underline{X}} = \prod_{n=1}^{N} f_X(x)$$

We will use $\mathbf{P}_{\underline{X}}$ to refer to the measure of probability encoded in this distribution.

Definition (Sample)**.** *Let X be a random variable associated with a measure of probability in $\mathcal{X}$. Let $\tau \in \mathbb{T}$ be a source of randomness allowed by such measurement. The set of transformations $\mathbb{T}$ is composed of the possible experimental conditions for the data sampling process from X. The dependency of measurement realizations $\boldsymbol{x} \sim \underline{X}$ on experimental conditions will be captured by index τ, and we will implicitely consider sample*

$$\boldsymbol{x} := \tau \circ \boldsymbol{x}$$

to be a realization of experiment $\underline{X} \overset{iid}{\sim} X$ under conditions τ. Given the stochastic nature of τ, we will also refer to it as noise instantiation.

2.3.1 Posterior distribution

Definition (Hypothesis class)**.** *Let $\mathcal{D}$ be the class of datasets generated from N-sized realizations of $\underline{X}$. A data science algorithm learns a function f implementing the following mapping:*

$$\begin{aligned} f : \mathcal{D} &\longmapsto \Theta \\ \boldsymbol{x} &\longmapsto (f(x_1), \ldots, f(x_N)) = \theta \end{aligned}$$

The hypothesis class Θ is the output space of hypothesis representing all possible outcomes of a function f learned on a dataset sampled from $\underline{X}$.

Intuitively, this framework interprets complexity from the perspective of the possible set of outcomes of the function, rather than the function class itself. It can be argued that both perspectives are equivalent, in the sense that any function class can be ultimately mapped to a specific hypothesis space Θ. Nevertheless, the underlying transformation is not homeomorphic in general, and more suitable generalization regularization constraints can be defined in Θ, especially when dealing with intractable function classes $\mathcal{F}_\Gamma$ represented by deep neural networks.

For instance, complexity in the hypothesis class can be associated to the nature of the randomness displayed by X. Ideally, too restrictive hypothesis classes that lack desirable hypothesis for some realization $\boldsymbol{x} \sim \underline{\mathrm{X}}$ should be avoided, and also those hypothesis classes containing unrealizable elements (i.e. hypothesis that are not outcome of any possible realization of the experiment). A richness condition can thus be postulated following this intuition.

Definition (Richness condition)**.** *We require a sufficiently rich set of experimental conditions* $\mathbb{T}$ *such that every hypothesis* $\theta \in \Theta$ *is the outcome of some realization* $\boldsymbol{x} \sim \underline{X}$*.*

$$\forall \theta, \exists \tau \in \mathbb{T} \text{ such that } f(\boldsymbol{x}) = \theta$$

Since we assume a mapping f and a data distribution $\mathbf{P}_{\underline{X}}$, we can describe the randomness of the hypothesis outcome conditioned on the the distribution of the data.

Definition (Posterior)**.** *Let* $\mathfrak{P}^f$ *be a probability distribution family under consideration. A probability distribution over the hypothesis class can be defined as a conditional distribution given an realization* $\boldsymbol{x} \sim \underline{X}$*. We will refer to this distribution as the posterior over* Θ *under* f*.*

$$\begin{aligned} \mathbf{P}^f : \mathcal{D} \times \Theta &\longmapsto \mathbb{R} \\ (\boldsymbol{x}, \theta) &\longmapsto \mathbf{P}^f(\theta \mid \boldsymbol{x}) \end{aligned}$$

The posterior $\mathbf{P}^f \in \mathfrak{P}^f$ *establishes the stochastic relation between data realizations and hypotheses.*

Using these definitions we can operate over Θ within the framework of probability theory. For instance, we can obtain the (prior) probability of a hypothesis to be selected by f as

$$\Pi^f(\theta) = \mathbb{E}_{\mathbb{T}}\, \mathbf{P}^f(\theta \mid \tau) = \mathbb{E}_{\underline{X}}\, \mathbf{P}^f(\theta \mid \boldsymbol{x})$$

from which we can derive a probabilistic version of the richness condition, where a limit case can be imposed with exactly one experiment per hypothesis, leading to a uniform prior:

$$\Pi^f(\theta) = |\Theta|^{-1}$$

Within this framework, selecting suitable hypothesis classes amounts to selecting posterior distributions that yield a higher probability to the desired subset of hypothesis. This is the leading principle that will guide the derivations that follow.

2.3.2 Generalization error

In order to define a robustness-based generalization error, we will proceed in an analogous way as we did in the previuous section. We will consider the datasets D' and D'' that arise from different sampling realizations $\boldsymbol{x}', \boldsymbol{x}'' \sim \underline{\mathrm{X}}$. Both realizations are independent and they differ in the implicit noise entailed by their measurement.

$$\mathbf{P}(\boldsymbol{x}', \boldsymbol{x}'') = \mathbf{P}(\boldsymbol{x}')\mathbf{P}(\boldsymbol{x}'')$$

Two posterior selection principles are derived from the robustness-informativeness trade-off:

P1 Posteriors should be expressive enough to cover the realizable subset of the hypothesis space.
P2 Equally likely inputs drawn from the same experiment should yield similar sets of hypothesis.

Definition (Description length)**.** *Let* $\mathcal{F}_\Gamma(\cdot)$ *be the function class containing all functions represented by a parametrization* Γ*. Let* $\mathbf{P}_\Gamma$ *be the universal distribution relative to* $\mathcal{F}_\Gamma$ *fulfilling the minimum description length principle. The description length of a function* $f_\gamma \in \mathcal{F}_\Gamma$ *is defined as the number of bits required to encode its parameters [16]. The code length of the argument of such distribution is*

$$DL_{f_\gamma}(\cdot) = -\log f_\gamma(\cdot)$$

The quality of the represented function f will be measured by the description length of its posterior, and thus a loss function can be defined as follows [5].

$$\ell(\theta, \boldsymbol{x}) = -\log \mathbf{P}^f(\theta \mid \boldsymbol{x})$$

Given that description length also accounts for the complexity of the hypothesis class and not only its generalization capabilities, we will normalize loss values by dividing by the description length of the prior.

$$-\log \Pi^f(\theta) = -\log \mathbb{E}_{\underline{X}} \mathbf{P}^f(\theta \mid \boldsymbol{x})$$

Definition (Generalization error)**.** *Let $\boldsymbol{x'}$ and $\boldsymbol{x''}$ be realizations of $\underline{X}$. Let Θ be the hypothesis class represented by f given $\underline{X}$. The generalization error is defined as the out-of-sample description length:*

$$\mathcal{G}_{\mathcal{X}} = \mathbb{E}_{\mathbf{P}^f(\boldsymbol{x'},\boldsymbol{x''})} \mathbb{E}_{\mathbf{P}^f(\theta|\boldsymbol{x'})} \left[-\log \frac{\mathbf{P}^f(\theta|\boldsymbol{x''})}{\Pi^f(\theta)} \right]$$

It amounts to the expected loss over the normalized posteriors on the validation data $\boldsymbol{x''}$ weighted over the posterior distribution on the training data $\boldsymbol{x'}$. Intuitively, a lower generalization error is achieved when good quality hypothesis on $\boldsymbol{x''}$ are likely to be drawn from $\boldsymbol{x'}$.

Lemma 2.3.1 (Posterior agreement)**.** *The generalization error $\mathcal{G}_{\mathcal{X}}$ is non-negative and has a lower bound $-\mathcal{J}$. We define $\mathcal{J}$ as the posterior agreement.*

Proof.

$$\begin{aligned}
\mathcal{G}_{\mathcal{X}} &\geq \mathbb{E}_{\mathbf{P}^f(\boldsymbol{x'},\boldsymbol{x''})} \left[-\log \left(\mathbb{E}_{\mathbf{P}^f(\theta|\boldsymbol{x'})} \frac{\mathbf{P}^f(\theta \mid \boldsymbol{x''})}{\Pi^f(\theta)} \right) \right] \\
&= \mathbb{E}_{\mathbf{P}^f(\boldsymbol{x'},\boldsymbol{x''})} \left[-\log \left(\sum_{\theta \in \Theta} \frac{\mathbf{P}^f(\theta \mid \boldsymbol{x'}) \mathbf{P}^f(\theta \mid \boldsymbol{x''})}{\Pi^f(\theta)} \right) \right] = -\mathcal{J} \\
&\geq -\log \left(\mathbb{E}_{\mathbf{P}^f(\boldsymbol{x'},\boldsymbol{x''})} \mathbb{E}_{\mathbf{P}^f(\theta|\boldsymbol{x'})} \mathbb{E}_{\mathbf{P}^f(\theta|\boldsymbol{x''})} \frac{\mathbf{P}^f(\theta \mid \boldsymbol{x''})}{\Pi^f(\theta)} \right) = 0.
\end{aligned}$$

□

where Jensen's inequality has been applied twice to the convex function $-\log(\cdot)$.

THINGS LEFT TO MENTION: - Symmetric
- Equivalent expression on KL and I. Interpretation of terms.

ALSO LEFT TO MENTION
- From hypothesis class selection to model selection. I think this relies on the interpretation of hypothesis. Regardless of whether my interpretation is right or wrong, we can always define the mapping $\Gamma \longmapsto \Theta$ as (not homeomorphic, but something similar, check carefeully), in a way that model selection is feasible (i.e. not only algorithm selection)

2.3.3 Maximum posterior agreement

This section has outlined the foundations of a generalization-rooted model selection criterion over the hypothesis space. The maximum posterior agreement criterion, which follows from Lemma 2.3.1, will be formalized as an optimization problem over the function class.

Definition (Kullback-Leibler divergence)**.** *Let $\mathbf{P}$ and $\mathbf{Q}$ be two probability distributions over the same support Θ. The Kullback-Leibler divergence of $Q(\theta)$ relative to $P(\theta)$ is defined as*

$$KL(\mathbf{P}(\theta) \parallel \mathbf{Q}(\theta)) = \mathbb{E}_{\mathbf{P}(\theta)} \left[\log \frac{\mathbf{P}(\theta)}{\mathbf{Q}(\theta)} \right]$$

Definition (Cross-entropy)**.** *Let $\mathbf{P}$ and $\mathbf{Q}$ be two probability distributions over the same support Θ. The cross-entropy of $Q(\theta)$ relative to $P(\theta)$ is defined as*

$$\mathcal{H}_{\mathbf{P},\mathbf{Q}} = -\mathbb{E}_{\mathbf{P}(\theta)} \log \mathbf{Q}(\theta)$$

Definition (Posterior agreement criterion)**.** *The posterior agreement model-selection criterion is defined as follows.*

$$\begin{aligned} &\sup_{\mathcal{F}} \mathcal{J} \\ &\text{s.t. } KL(\mathbf{\Pi}^f(\theta) \parallel |\Theta|^{-1}) \leq \xi \end{aligned}$$

where $\xi \in \mathbb{R}$ represents a small allowed deviation from uniformity in the prior.

Theorem 2.3.1. *The optimal $\mathbf{P}_*^f$ maximizing the posterior agreement criterion defines a lower bound in the generalization error $\mathcal{G}_{\mathcal{X}}$ under the richness condition.*

$$\inf_{\mathcal{F}} \mathcal{G}_{\mathcal{X}} \geq -\sup_{\mathcal{F}} \mathcal{J}$$

Proof. We consider the lagrangian formulation of the generalization error minimization problem and apply Lemma 2.3.1.

$$\begin{aligned} &\inf_{\mathcal{F}} \left\{ \mathcal{G}_{\mathcal{X}} + \alpha \mathrm{KL}(\mathbf{\Pi}^f(\theta)) \parallel |\Theta|^{-1} \right\} \\ =\ &\inf_{\mathcal{F}} \left\{ \mathcal{G}_{\mathcal{X}} + \alpha \mathbb{E}_{\mathbf{\Pi}^f(\theta)} \log \mathbf{\Pi}^f(\theta) + \alpha \mathbb{E}_{\mathbf{\Pi}^f(\theta)} \log |\Theta| \right\} \\ \geq\ &\alpha \log |\Theta| + \inf_{\mathcal{F}} \{ \alpha \mathcal{H}_{\mathbf{\Pi}^f} \} - \sup_{\mathcal{F}} \{ \mathcal{J} \} \\ \geq\ &-\sup_{\mathcal{F}} \mathcal{J} \end{aligned}$$

The last inequality follows from the fact that the entropy does not exceed the log-cardinality of the hypothesis class.

$$\mathcal{H}_{\mathbf{\Pi}^f}(\theta) \leq \log |\Theta|, \quad \forall \mathbf{\Pi}^f$$

□

Chapter 3

Experimental setup

This chapter delineates the covariate shift setting within the supervised classification framework and introduces an operative formulation of posterior agreement. This formulation represents the cornerstone of this work as it allows for robustness-based model selection in discrete hypothesis classes.

3.1 Problem formulation

3.1.1 The classification problem

Out of all the possible learning problems in which a distribution shift can be defined, this project will focus on the supervised classification of images. The function space to navigate is composed of parametrized classifiers.

Definition (Classifier)**.** *Let $\mathcal{X}$ and $\mathcal{Y} \subset N$ be the input and output spaces of the target function, respectively. Let $K \in \mathbb{N}$ be the cardinality of $\mathcal{Y}$. A K-class classifier can be defined as the composition of three functions:*

- *A feature extractor. This function maps the input space to a d-dimensional feature space.*

$$\begin{aligned} \Phi : \mathcal{X} &\longmapsto \mathbb{R}^d \\ x &\longmapsto \Phi(x) = z \end{aligned}$$

- *A discriminant function. This function assigns a score to each of the K classes given a feature vector.*

$$\begin{aligned} \boldsymbol{F} : \mathbb{R}^d &\longmapsto \mathbb{R}^K \\ z &\longmapsto (F_1(z), \dots, F_K(z)) = \boldsymbol{F}(z) \end{aligned}$$

- *A decision rule. This function assigns the class label from a vector of scores. We will set it to be the maximum a posteriori (MAP) rule.*

$$\begin{aligned} \eta : \mathbb{R}^K &\longmapsto \mathcal{Y} = \{1, \dots, K\} \\ \boldsymbol{F}(z) &\longmapsto \hat{y} = \arg\max_j F_j(z) \end{aligned}$$

A classifier is defined as the composition of these three functions.

$$c = \eta \circ \boldsymbol{F} \circ \Phi$$

The results presented in this work are limited to neural network classifiers. These are parametrized NN architectures in $\Gamma \subseteq \mathbb{R}^{|\Gamma|}$, such that:

$$c : \mathcal{X} \times \Gamma \longmapsto \mathcal{Y} = \{1, \dots, K\}$$
$$(x, \gamma) \longmapsto c(x; \gamma) = \hat{y}$$

thus $c(x; \gamma) = \eta \circ (\boldsymbol{F} \circ \Phi)(x; \gamma)$.

The concepts defined in the previous chapter allow us to formalize the learning problem in which our robustness experiments will be conducted. We will refer to this problem as a K-class classification.

Definition (K-class classification)**.** *Let D be a supervised dataset. Let $c(\cdot; \gamma)$ be a neural network classifier, parametrized in $\Gamma \subseteq \mathbb{R}^{|\Gamma|}$. Let RRM_D be the regularized risk minimization problem for c on D. Let $\mathcal{L}$ be the cross-entropy loss function for the classifier c.*

$$\mathcal{L}(x, y) = -\log F_y(\Phi(x); \gamma)$$

The K-class classification problem is the RRM_D with loss function $\mathcal{L}$ parametrized in Γ.

$$\gamma^* = \arg\min_{\gamma \in \Gamma} -\frac{1}{N} \sum_{n=1}^{N} \log F_{y_n}(x_n; \gamma) + \lambda \Omega(\gamma)$$

No further characterization of the regularization factor will be provided in this chapter, as specific learning models and methods will be introduced together with the results.

3.1.2 Robustness to covariate shift

The concept of robustness, as defined in the previous chapter, entails a measure of the stability of the learner to the randomness of the data sampling process, but also requires an adequate characterization of such randomness. In the context of the K-class classification problem, sampling randomness can be formalized as a shift in the distribution of the input space, also known as covariate shift.

Definition (Covariate shift)**.** *Let $\boldsymbol{x}'$ and $\boldsymbol{x}''$ be two N-sized samples of $\underline{X} \overset{iid}{\sim} X$. A covariate shift exists between $\boldsymbol{x}'$ and $\boldsymbol{x}''$ if their (empirical) distributions are significantly different*[1] *for N large enough:*

$$\mathbf{P}_{\boldsymbol{x}'} \not\sim \mathbf{P}_{\boldsymbol{x}''}$$

It must be noted that, since the target function is assumed to be invariant (see Section 1.1.2), the true distribution over the output space remains the same [33].

The presence of covariate shift as defined above already leads to a non-zero generalization error, given that $\boldsymbol{x}'$ and $\boldsymbol{x}''$ represent different noise instantiations and result in different learning outcomes. Nevertheless, this definition can be further expanded to encompass more practical sources of shift in the context of classification tasks.

Definition (Domain shift)**.** *Let X' and X'' be two random variables associated to different sampling experiments in $\mathcal{X}$ such that $\mathbf{P}_{X'} \not\sim \mathbf{P}_{X''}$. The randomness entailed by their respective measurement process is also different in general (i.e. $\mathbb{T}' \neq \mathbb{T}''$). In such case*

$$\boldsymbol{x}' \sim \underline{X}' \overset{iid}{\sim} X' \text{ and } \boldsymbol{x}'' \sim \underline{X}'' \overset{iid}{\sim} X''$$

lead to a covariate shift known as domain or out-of-distribution (OOD), given that the fundamental source of distribution shift is the difference in the probability measure over the support induced by each experiment. [33]

[1] The notion of difference relies on the nature of the data. Common measures include statistical distances such as the Kullback-Leibler divergence, Wasserstein distance, or even simpler metrics like the difference in means or variances. These methods help establish whether observed differences are statistically significant. [33]

In the OOD case, $\boldsymbol{x}'$ and $\boldsymbol{x}''$ are drawn from different random variables, each with a distinct probability landscape over the support, namely source and target domains, that result in implicit differences (sometimes unbalanced) in the distribution of some features. Therefore, empirical distributions $\mathbf{P}_{\boldsymbol{x}'}$ and $\mathbf{P}_{\boldsymbol{x}''}$ will be different in general, and thus a covariate shift will be induced leading to a non-zero generalization error.

The reader should note that this definition generalizes the concept of sampling randomness as defined in the previous chapter, as it explicitely allows for X' and X'' to be different random variables. Therefore, each realization of $\underline{\mathrm{X}}'$ and $\underline{\mathrm{X}}''$ will not only entail a different noise instantiation but might also favour a different region of $\mathcal{X}$.

Definition (Adversarial shift)**.** *Let $\boldsymbol{x}' \sim \underline{X}$ be a sample drawn from experiment $X = \tau \circ X$. Let $\boldsymbol{\Delta}$ be a perturbation over the sample space. In this case, $\boldsymbol{x}''$ is generated by applying the perturbation to $\boldsymbol{x}'$.*

$$\boldsymbol{x}'' = \boldsymbol{x}' + \boldsymbol{\Delta}$$

which induces a covariate shift known as adversarial, given that perturbation $\boldsymbol{\Delta}$ is crafted ad-hoc to hinder the output of the model.

In adversarial examples, sampling randomness is not the source of distribution shift, as both $\boldsymbol{x}'$ and $\boldsymbol{x}''$ arise from the same realization of the experiment.

In this work we will consider a wider concept of sampling randomness that does not only comprise the implicit noise instantiation of each realization $\boldsymbol{x} \sim \underline{\mathrm{X}}$ but also the explicit shift in the distribution of the input space generated by intentional or unintentional perturbations of the data generation process. This broader interpretation aligns practical covariate shift experiments with the robustness framework defined in previouos chapters.

Once the possible sources of randomness in the data generation process have been established and formalized, a general concept of robustness measure must be introduced accordingly, so that the suitability of posterior agreement as a robustness metric can be assessed.

Definition (Robustness metric)**.** *Let D' and D'' be datasets generated from realizations $\boldsymbol{x}'$ and $\boldsymbol{x}''$, respectively. A robustness metric is a function $\Omega : \mathcal{D}'' \times \mathcal{F} \longmapsto \mathbb{R}$ that quantifies the generalization capability of a learned $\hat{f}_{D'} \in \mathcal{F}$ to observations in D''.*

The baseline robustness metric in supervised classification tasks is accuracy, defined as the proportion of correct predictions achieved by a learned classifier $\hat{c}_{D'}$ over dataset D''.

$$\mathrm{ACC}_{D'}(D'') = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{\delta}_{y''_n} \left(\hat{c}_{D'}(x''_n) \right)$$

As it was argued before, we will interpret the concept of generalization from the perspective of the possible learning outcomes of a specific experiment. The ultimate goal of robustness measurement is thus the characterization of the "resolution" limit that can be achieved in the hypothesis space consistent with the intrinsic randomness entailed by each possible realization of the experiment.

The resolution limit does not depend on the model but on the nature of the randomness of the data generation process. Therefore, a robustness metric should evaluate how stable are hypothesis to different realizations of the same experiment, regardless of the complexity of the model. The more complex the model is, the higher will be the resolution of its associated hypothesis space, but the more prone will be to overfit to the noise and thus yield unstable hypothesis. A regularization or model selection procedure derived from the robustness metric should then find the sweet spot between resolution and stability.

Properties (Robustness metric)**.** *A suitable robustness metric should possess the following two properties. [7]*

P1 *(Non-increasing) The metric should be non-increasing with respect to the response of the model under increasing levels of shift.*

P2 *(Model-independent) The metric should differentiate models only by their generalization capabilities against covariate shift. For instance, the metric should be independent of the task performance of the model.*

The first property is commonly satisfied, but the second one entails a specific interpretation of stability that is not straightforward to quantify. Let us consider the following example.

Example 3.1.1. *Let $\mathcal{D}$ be a class of balanced binary supervised datasets; that is, containing exactly the same number of observations of each label. Let's consider the following three classifiers evaluating observations in $D \in \mathcal{D}$.*

C1 *A random classifier, returning a random prediction to each observation in the dataset. Overall performance would tend to 50% accuracy as dataset size increases.*

C2 *A constant classifier, returning exactly the same prediction for each observation in the dataset. Overall performance is 50% accuracy, as the dataset is exactly balanced.*

C3 *A perfect classifier, returning the correct prediction to each observation in the dataset. Overall performance is 100% accuracy.*

*In terms of performance, **C1** and **C2** are equivalent when dataset size is big enough, and **C3** would be selected as the best. Nevertheless, a robustness metric compliant with **P2** would evaluate **C1** to be non-robust, while **C2** and **C3** would be considered equivalent (and achieve maximum robustness), since their set of hypothesis remains the same for every dataset in $\mathcal{D}$.*

It is now straightforward to see that accuracy or any task-dependent metric does not comply with **P2**. This work will provide a **P2**-compliant robustness metric derived from the concept of posterior agreement.

Before that, the statement of the problem must be completed with an extended characterization of adversarial and out-of-distribution shifts from a practical perspective; that is, the specific characterization of the shift magnitude that will be considered in the experiments.

3.1.3 Adversarial setting

The magnitude of adversarial shifts will be quantified by an aggregated measure of the perturbation applied to the each observation in the dataset.

Definition (Perturbation)**.** *Let $\boldsymbol{x}'$ be a realization of $\underline{X} \overset{iid}{\sim} X$ with support $\mathcal{X} \subset \mathbb{R}^d$. Let $x \in \boldsymbol{x}'$ be an observation of the sample. Let $\mathbf{B}_p^\epsilon(x)$ be the ℓ_p-norm ball of radius ϵ centered at x. A perturbation Δ is defined as*

$$\Delta \in \mathbb{R}^d \text{ s.t. } x + \Delta \in \mathbf{B}_p^\epsilon(x)$$

where $\epsilon \in [0, 1]$ keeps it hard-box constrained due to the normalization of the input space. A perturbation set $\boldsymbol{\Delta}$ will be ϵ_p-constrained if each of its components satisfies the previous definition. In such case,

$$\boldsymbol{x}'' = \boldsymbol{x}' + \boldsymbol{\Delta}$$

defines an adversarial shift of magnitude ϵ_p.

As it was previously outlined, the existence of adversarial examples in NNs was initially associated with their heavily non-linear nature and a lack of smoothness over the hypothesis space [37]. Nevertheless, it is instead the linearity of their units and the high dimensionality of inner representations that make them vulnerable to perturbations in certain directions [15].

Let $w \in \mathbb{R}^d$ be the weight vector of a NN unit. The difference in activation responses between perturbed and original observations

$$w^\top (x'' - x') = w^\top \Delta$$

will be maximum when $\Delta \propto \text{sign}(w)$; that is, when the perturbation is aligned with the weights. Following the same intuition, we can define the most adversarial direction of perturbation as the one maximizing the resulting loss.

Attack (FGSM)**.** *Perturbations are generated by alignment with the gradient of the loss with respect to the original observation.*

$$\Delta = \epsilon_p \, \text{sign}(\nabla_{x'} \mathcal{L}(x', y; \gamma))$$

This is known as the fast gradient sign method attack [15].

An effective regularizer for adversarial training can be built by including the FGSM term on the objective that makes the model robust to ϵ_p-constrained perturbations [15]. A multi-step version can be immediately derived that sistematically perturbs observations in the most adversarial direction at each optimization step.

Attack (PGD)**.** *Perturbations are generated by iteratively applying the FGSM perturbation to each step and projecting the result back to the ϵ_p-constrained ball.*

$$x^{s+1} = \Pi_{\mathbf{B}_p^\epsilon(x)} \left(x^s + \epsilon_p \, \text{sign}(\nabla_{x'} \mathcal{L}(x', y; \gamma)) \right)$$

where Π is the projection operator. This is known as projected gradient descent attack [27].

It can be shown that a PGD regularizer for adversarial training navigates the loss landscape to minimize the model loss under the maximum adversarial perturbation.

$$\gamma^* = \arg\min_{\gamma \in \Gamma} \left\{ \mathbb{E} \left[\max_{\Delta} \mathcal{L}(f(x + \Delta), y; \gamma) \right] \right\}$$

The inherent complexity of this optimization problem requires making certain assumptions in oder to solve it. For instance, it is commonly assumed that the loss landscape contains numerous local minima, but with very similar values. Then, the distribution of loss values attained with different starting points is well concentrated and has no outliers, which fosters robustness.

Our experimental setup will also consider a minimum-norm adversarial training method, that works by iteratively finding the sample misclassified with maximum confidence within $\mathbf{B}_p^\epsilon(x)$, while adapting its radius to minimize the distance between the perturbed sample and the decision boundary.

Attack (FMN)**.** *Perturbations are generated as follows.*

$$\begin{aligned} \Delta^\star = \arg\min_{\Delta} \; & ||\Delta||_p \\ & s.t. \;\; F_y(x; \gamma) - \max_{j \neq y} F_j(x; \gamma) < 0, \\ & x + \Delta \in \mathbf{B}_p^\epsilon(x) \end{aligned}$$

This is known as the fast minimum-norm (FMN) attack [32].

3.1.4 Domain generalization setting

As described in the introductory chapter, domain generalization refers to a specific setting in which several instantiations of the data are shifted in the OOD sense, and only a subset of them are available. We can formalize the problem as follows.

Definition (Domain generalization)**.** *Let $\mathcal{S} = \{X_1^S, \cdots, X_S^S\}$ and $\mathcal{T} = \{X_1^T, \cdots, X_T^T\}$ be two sets of random variables associated with specific probability measures over the input space $\mathcal{X}$. The probability measure induced by each random variable implicitly selects a region of the support $\mathcal{X}$, so in this context we will metonymically refer to them as domains. Set $\mathcal{S}$ encompasses source domains, and $\mathcal{T}$ target domains (see Section 1.1.2) [26, 42].*

According to Definition 3.1.2, datasets sampled from each domain entail a OOD shift that will lead to non-zero generalization error. The domain generalization problem consists of selecting the model with the lowest generalization error between source target domains without having access to the target domains at all.

Unlike the adversarial case, there is no standard way of quantifying the magnitude of the shift besides reporting model performance in benchmark datasets. In this work we will consider both dataset-specific measures of variation and a general-purpose metric to evaluate the structure similarity between images. ... FINISH WITH NEW APPROACH, MAYBE THE DISTANCE IN THE FEATURE SPACE, ETC.

Definition (ASS)**.** *The structural similarity index (SSIM) separates the task of similarity measurement into three comparisons, namely luminance, contrast and structure [44].*

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

where μ_x, μ_y and σ_x, σ_y represent the means and variances of input images, respectively, and c_1 and c_2 are constants used to stabilize the division with weak denominator. We define the average structural similarity (ASS) as its average over a specific dataset [17].

Taking into account the magnitude of the existing covariate shift among source domains, the performance of the selected model will be reported for each of the target domains. In particular, average accuracy and worst-case accuracy will be provided [53].

3.2 Robustness enhancement

This project will evaluate the suitability of posterior agreement as a robustness metric in the adversarial and out-of-distribution settings. In accordance with **P2** (see Properties 3.1.2), we must eventually assess whether our metric is able to differentiate between robust and non-robust models. For that reason, we will consider ERM (see Definition 2.1) as our baseline vanilla model and compare its generalization performance to covariate shift with two models representing two different robustness enhancement strategies.

As a first approach, DNN architectures will be trained by means of IRM [1], a regularization method (see Definition 2.1) driven by feature alignment. In particular, IRM follows a domain-invariant representation learning strategy emerging from the hypothesis of invariance of the causal structure of the input-output relation. The existence of a data representation encoding that causality in the feature space is assumed, and thus the invariance of such representation under different source domains is enforced [26].

Definition (IRM)**.** *Let $\mathcal{R}^d$ be the risk of a classifier c (see Definition 3.1.1) over domain $d \in \mathcal{S}$. The IRM problem minimizes risk over all domains while enforcing the feature extractor to yield domain-invariant representations [1].*

$$c^* = \min_{c=\eta\circ\boldsymbol{F}\circ\Phi} \sum_{d\in\mathcal{S}} \mathcal{R}^d(c)$$
$$s.t.\ (\eta\circ\boldsymbol{F}) = \arg\min_{\bar{c}} \mathcal{R}^d(\bar{c})\ \forall d\in\mathcal{S}$$

A surrogate version of the problem simplifies its implementation:

$$c^* = \min_{c} \sum_{d\in\mathcal{S}} \mathcal{R}^d(c) + \lambda||\nabla_{w|w=1}\mathcal{R}^d(w\cdot c)||^2$$

where w is a dummy classifier added to the problem to relax the invariance constraint and enforce instead that the optimal feature representation induces an optimal classifier that is the same in all domains (see [1] for details). The balance between the ERM term and the invariance predictor is controlled by the regularization hyperparameter $\lambda\in[0,\infty)$.

As a second approach, we will consider a data generation strategy that populates the gaps among source domain distributions with new observations obtained via interpolation. Learning invariant features via selective augmentation (LISA) is accomplished by interpolating original samples that either belong to the same class but a different source domain (LISA-D), or belong to the same domain but have different labels (LISA-L). The former helps the model learn domain-invariant features, while the latter fosters the learning of class-invariant features. Two interpolation strategies will be considered, namely Mixup [52] and CutMix [50].

Definition (LISA)**.** *Let D_1 and D_2 be datasets associated with two different source domains. A convex interpolation with weight $\lambda\sim$ Beta(α,β) generates a new sample that lies in the line segment connecting the two original samples.*
(LISA-D) Let $(x_1,y_1)\in D_1$ and $(x_2,y_2)\in D_2$, with $y_1=y_2$.
(LISA-L) Let $(x_1,y_1),(x_2,y_2)\in D_1$, with $y_1\neq y_2$.

$$x_{LISA} = \lambda x_1 + (1-\lambda)x_2$$
$$y_{LISA} = \lambda y_1 + (1-\lambda)y_2$$

where a random value $s\in$ Bernoulli(p) will determine the strategy to be applied, being $p\in[0,1]$ the probability of LISA-L [48].

3.3 Robustness assessment with posterior agreement

As it was argued in Section 3.1.2, accuracy and by extension the custom law of quantifying robustness by reporting model performance in benchmark datasets does not offer any theoretical mechanism for the true characterization of robustness and the nature of the shift that the model is being made robust to.

In this section we will derive a practical version of posterior agreement (PA) that can be used in supervised classification tasks to assess the generalization performance to different kinds of shifts. Even before reaching the final expression, a fundamental distinction between PA and accuracy can be made, namely the fact that posterior agreement is computed with the output of the discriminant function, which encodes the confidence in the classification, whereas accuracy only considers the output prediction label. Confidence information increases the discriminative power of the metric, for instance when comparing models with similar predictive capabilities, but also involves a probability distribution over the output space that allows for a broader theoretical interpretation.

3.3.1 Posterior in classification tasks

Definition 2.3.1 established the posterior as the probability distribution over the hypothesis space encoding the stochastic nature of model outputs. The hypothesis class Θ of a K-class classification

problem is the set of all possible vectors of labels associating each of the N samples to one of the K classes, which is

$$\Theta = \{1, \dots, K\}^N$$

with cardinality $|\Theta| = K^N$.

Theorem 3.3.1 (Classification posterior)**.** *Let Θ be the classification hypothesis class associated with the K-class classification problem with approximating function c. The posterior distribution class $\mathfrak{P}^c$ is the Gibbs distribution family with inverse temperature parameter β.*

$$\mathbf{P}^c(\theta|\boldsymbol{x}) = \frac{\exp\left(\beta R(\theta, \boldsymbol{x}; \gamma)\right)}{\sum_{\theta \in \Theta} \exp\left(\beta R(\theta, \boldsymbol{x}; \gamma)\right)}$$

Proof. The proof is based on the maximum entropy principle (MEP), which states that given some prior testable information to be encoded by a probability distribution, the distribution that best encodes that information is the one minimizing additional assumptions besides the testable information; that is, the one maximizing information entropy within the testable space. Testable information amounts to certain constraints on the MEP optimization problem over the non-negative, Lebesgue-integrable function class $\mathcal{P}$.

$$\begin{aligned}
&\max_{\mathbf{P}^c(\theta|\boldsymbol{x}) \in \mathcal{P}} \mathcal{H}_{\mathbf{P}^c}(\theta \mid \boldsymbol{x}) \\
&\quad \text{s.t.} \sum_{\theta \in \Theta} \mathbf{P}^c(\theta \mid \boldsymbol{x}) = 1 \\
&\quad \mathbb{E}_{\mathbf{P}^c(\theta|\boldsymbol{x})}[R(\theta, \boldsymbol{x})] = \mu \;\; \forall \theta \in \Theta \\
&\quad [\mathbf{P}^c(\theta_i \mid \boldsymbol{x}) - \mathbf{P}^c(\theta_j \mid \boldsymbol{x})][R(\theta_i, \boldsymbol{x}) - R(\theta_j, \boldsymbol{x})] \geq 0 \;\; \forall \theta_i, \theta_j \in \Theta
\end{aligned}$$

where $\mu \in \mathbb{R}$ is a hyperparameter ensuring that the expected confidence is finite and the last constraint imposes a monotonic relationship between the confidence and the posterior. The lagrangian formulation of the problem with equality constraints is:

$$\mathcal{L}(\mathbf{P}^c, \alpha, \beta) = \mathcal{H}_{\mathbf{P}^c}(\theta \mid \boldsymbol{x}) + \alpha \left(1 - \sum_{\theta \in \Theta} \mathbf{P}^c(\theta \mid \boldsymbol{x})\right) + \beta \left(\mathbb{E}_{\mathbf{P}^c(\theta|\boldsymbol{x})}[R(\theta, \boldsymbol{x})] - \mu\right)$$

Its derivative with respect to $\mathbf{P}^c(\theta \mid \boldsymbol{x})$ is:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{P}^c(\theta \mid \boldsymbol{x})} = -1 - \log \mathbf{P}^c(\theta \mid \boldsymbol{x}) - \alpha + \beta R(\theta, \boldsymbol{x})$$

which has a unique solution

$$\mathbf{P}^c(\theta \mid \boldsymbol{x}) = \frac{\exp\left(\beta R(\theta, \boldsymbol{x})\right)}{\exp\left(1 + \alpha\right)}$$

Setting $\exp(1 + \alpha) = \sum_{\theta \in \Theta} \exp\left(\beta R(\theta, \boldsymbol{x})\right)$ and $\beta \geq 0$ we ensure normalization and thte fulfillment of the monotonic relationship constraint. □

The confidence R in supervised classification tasks is given by the negative loss resulting from the MAP principle (see Definition 3.1.1).

Definition (Classification confidence)**.** *Let D be a dataset associated with a realization $\boldsymbol{x} \sim \underline{X}$. Let $F_j(\cdot; \gamma)$ be the j-th component of the score vector returned by the discriminant of the classifier. The cost function driving posterior selection will be the negative confidence in the prediction.*

$$R(\theta, \boldsymbol{x}; \gamma) = -\sum_{n}^{N} F_{\theta_i}(x_i; \gamma)$$

where θ_i is the class label associated with the i-th sample in the dataset.

3.3.2 The posterior agreement kernel

Lemma 3.3.1 (Exchangeability)**.** *Let $N, K \in \mathbb{N}$ and let $\{\mathcal{E}_{ij} \mid i \leq N, j \leq K\}$ be an indexed set of values. Then,*

$$\sum_{c \in \mathcal{C}} \prod_{i=1}^{N} \mathcal{E}_{i,c(i)} = \prod_{i=1}^{N} \sum_{j=1}^{K} \mathcal{E}_{ij}$$

Proof. See Appendix A.1. □

Theorem 3.3.2 (Posterior factorization)**.** *The posterior distribution for a classification problem can be factorized as follows:*

$$\mathbf{P}^c(\theta \mid \boldsymbol{x}) = \prod_{i}^{N} \mathbf{P}_i^c(\theta_i \mid \boldsymbol{x}) = \prod_{i=1}^{N} \frac{\exp\left(\beta F_{\theta_i}(x_i)\right)}{\sum_{j=1}^{K} \exp\left(\beta F_j(x_i)\right)}$$

Proof. See Appendix A.1. □

Theorem 3.3.3 (PA kernel for classification)**.** *Let $\boldsymbol{x}'$ and $\boldsymbol{x}''$ be N-sized realizations of $\underline{X}$. Let Θ be the hypothesis class represented by classifier c under $\mathcal{X}$. With no prior information about Θ, the posterior agreement kernel for supervised K-class classification tasks has the following expression.*

$$\mathrm{PA}\left(\boldsymbol{x}', \boldsymbol{x}''; \beta\right) = \frac{1}{N} \sum_{i=1}^{N} \log \left\{ |\Theta| \sum_{j=1}^{K} \mathbf{P}_i^c\left(j \mid \boldsymbol{x}'\right) \mathbf{P}_i^c\left(j \mid \boldsymbol{x}''\right) \right\}$$

where $\mathbf{P}_i^c(j \mid \boldsymbol{x})$ can be shown to be:

$$\mathbf{P}_i^c\left(j \mid \boldsymbol{x}\right) = \frac{\exp\left(\beta F_j(x_i)\right)}{\sum_{k=1}^{K} \exp\left(\beta F_k(x_i)\right)}$$

Proof. The posterior agreement $\mathcal{J}$ has the following expression, derived in Lemma 2.3.1:

$$\mathcal{J} = \mathbb{E}_{X', X''} \left[\log \left(\mathbb{E}_{\mathbf{P}^c(\theta \mid \boldsymbol{x}')} \frac{\mathbf{P}^c(\theta \mid \boldsymbol{x}'')}{\mathbf{\Pi}^c(\theta)} \right) \right]$$

As defined previously, Θ is a discrete, finite set of possible classification vectors of the N observations, and the sampling distribution $\mathbf{P}_X$ is assumed to be uniform. Therefore, the expectation operators amount to:

$$\mathbb{E}_{X', X''} = \frac{1}{N} \sum_{i=1}^{N} \cdot$$

$$\mathbb{E}_{\mathbf{P}^c(\theta \mid \boldsymbol{x}')} = \sum_{\theta \in \Theta} \mathbf{P}^c(\theta \mid \boldsymbol{x}') \cdot$$

A non-informative prior is assumed, thus enforcing the richness condition:

$$\mathbf{\Pi}^c(\theta) = |\Theta|^{-1}$$

$\mathbf{P}^c(\theta|\boldsymbol{x})$ can be factorized on the terms expressed in Theorem 3.3.2.

$$\mathbf{P}^c(\theta \mid \boldsymbol{x}) = \prod_{i}^{N} \mathbf{P}_i^c(\theta_i \mid \boldsymbol{x}) = \prod_{i=1}^{N} \frac{\exp\left(\beta F_{\theta_i}(x_i)\right)}{\sum_{k=1}^{K} \exp\left(\beta F_k(x_i)\right)}$$

Operating analogously for $\boldsymbol{x}'$ and $\boldsymbol{x}''$, the expression for the PA kernel is obtained.

$$
\begin{aligned}
\mathrm{PA}\left(\boldsymbol{x}', \boldsymbol{x}''; \beta\right) =& \frac{1}{N} \sum_{i=1}^{N} \left[\log \left(\sum_{\theta \in \Theta} \mathbf{P}^c(\theta \mid \boldsymbol{x}') \frac{\mathbf{P}^c(\theta \mid \boldsymbol{x}'')}{|\Theta|^{-1}} \right) \right] \\
=& \frac{1}{N} \sum_{i=1}^{N} \left[\log \left(|\Theta| \sum_{\theta \in \Theta} \mathbf{P}^c(\theta \mid \boldsymbol{x}') \mathbf{P}^c(\theta \mid \boldsymbol{x}'') \right) \right] \\
=& \frac{1}{N} \sum_{i=1}^{N} \left[\log \left(|\Theta| \sum_{\theta \in \Theta} \prod_{i=1}^{N} \frac{\exp\left(\beta F_{\theta_i}(x_i')\right)}{\sum_{k=1}^{K} \exp\left(\beta F_k(x_i')\right)} \frac{\exp\left(\beta F_{\theta_i}(x_i'')\right)}{\sum_{k=1}^{K} \exp\left(\beta F_k(x_i'')\right)} \right) \right]
\end{aligned}
$$

Finally, applying Lemma 3.3.1 to the product inside the logarithm, we reach the final expression. □

Theorem 3.3.4 (Properties of the PA kernel)**.** $\mathrm{PA}\left(\boldsymbol{x}', \boldsymbol{x}''; \beta\right)$ *has the following properties.*

P1 *(Non-negativity)* $\mathrm{PA}\left(\boldsymbol{x}', \boldsymbol{x}''; \beta\right) \geq 0 \;\; \forall \boldsymbol{x} \sim \underline{X}$ *and* $\beta \in \mathbb{R}^+$.

P2 *(Symmetry)* $\mathrm{PA}\left(\boldsymbol{x}', \boldsymbol{x}''; \beta\right) = \mathrm{PA}\left(\boldsymbol{x}'', \boldsymbol{x}'; \beta\right)$. *This property is important from the robustness perspective, given that noise instantiations are not indexed and no reference noiseless experiment can be performed.*

P3 *(Concavity)* $\mathrm{PA}\left(\boldsymbol{x}', \boldsymbol{x}''; \beta\right)$ *is a concave function of* $\beta \in \mathbb{R}^+ \;\; \forall \boldsymbol{x} \sim \underline{X}$. *This means that the kernel optimization problem will have a unique solution.*

Proof. See Appendix A.2. □

3.3.3 Analytical example

- Include the analytical example derivation, adapt notation. Probably also leave some things in the appendix. For example, in the appendix you can leave the expectations over the normal distribution.

3.4 Posterior agreement beyond robustness

- Alternative formulation
- OJO: For cross-validation, I can consider the final feature vector associated with the image (i.e. before the classification layer) to be the measurement. Models trained with different subsets of data will have a different noise instantiation of the same measurement. therefore the alternative formulation is not necessary for cross-validation.
- Explain why important, and formalize the data augmentation strategy (presentation)

Chapter 4

Results and discussion

Blah, blah ...
DRAFT

- It's difficult to assess P2 (independence of robustness metric on the task performance of the model) because more complex models will be better able to represent invariant features (if proper regularization is performed) than less complex models, and complexity (again, if properly regularized) drives also task performance. For that reason, we will reproduce the results of the Example (classifiers) with posterior agreement.

- Justify that the hypothesis class in the OOD case is implicitly considered to be the union of the hypothesis classes of each of the random variables. Maybe this belongs to section 3.1.2

Chapter 5

Discussion

Blah, blah ...

Appendix A

Supplementary material

We will define some notation shortcuts for the following proofs.

A.1 Proof of problem formulation

Lemma A.1.1. *Let $N, K \in \mathbb{N}$ and let $\{\mathcal{E}_{ij} \mid i \leq N, j \leq K\}$ be an indexed set of values. Then,*

$$\sum_{c \in \mathcal{C}} \prod_{i=1}^{N} \mathcal{E}_{i,c(i)} = \prod_{i=1}^{N} \sum_{j=1}^{K} \mathcal{E}_{ij}$$

Proof. By induction on N. For the $N = 1$ base case, observe that $\mathcal{C}$ has only K elements, as there are only K functions mapping $\{1\}$ to $\{1, \ldots, K\}$. Then

$$\sum_{c \in \mathcal{C}} \prod_{i \leq N} \mathcal{E}_{i,c(i)} = \sum_{c \in \mathcal{C}} \mathcal{E}_{1,c(1)} = \sum_{j \leq K} \mathcal{E}_{1,j} = \prod_{i \leq N} \sum_{j \leq K} \mathcal{E}_{i,j}.$$

Assume now that the result holds for some N. We demonstrate then that it also holds for $N + 1$. Observe that there is a bijection between $\mathcal{C}$ and $\{1, \ldots, K\}^N$. Therefore, we identify every function $c \in \mathcal{C}$ with the tuple $(c(1), \ldots, c(N))$. Conversely, we identify every tuple $(c_1, \ldots, c_N) \in \{1, \ldots, K\}^N$, with the function c that maps i to c_i.

$$
\begin{aligned}
&\sum_{c\in\mathcal{C}} \prod_{i\leq N+1} \mathcal{E}_{i,c(i)} = \\
&= \sum_{(c_1,\ldots,c_{N+1})\in\{1,\ldots,K\}^{N+1}} \prod_{i\leq N+1} \mathcal{E}_{i,c_i} \\
&= \sum_{\substack{(c_1,\ldots,c_N)\in\{1,\ldots,K\}^{N} \\ c_{N+1}\leq K}} \prod_{i\leq N+1} \mathcal{E}_{i,c_i} \\
&= \sum_{(c_1,\ldots,c_N)\in\{1,\ldots,K\}^{N}} \sum_{c_{N+1}\leq K} \prod_{i\leq N+1} \mathcal{E}_{i,c_i} \\
&= \sum_{(c_1,\ldots,c_N)\in\{1,\ldots,K\}^{N}} \sum_{c_{N+1}\leq K} \left(\mathcal{E}_{N+1,c(N+1)} \prod_{i\leq N} \mathcal{E}_{i,c_i} \right) \\
&= \left(\sum_{c_{N+1}\leq K} \mathcal{E}_{N+1,c(N+1)} \right) \sum_{(c_1,\ldots,c_N)\in\{1,\ldots,K\}^{N}} \prod_{i\leq N} \mathcal{E}_{i,c_i} \\
&= \left(\sum_{c_{N+1}\leq K} \mathcal{E}_{N+1,c(N+1)} \right) \prod_{i\leq N} \sum_{j\leq K} \mathcal{E}_{i,j} \\
&= \left(\sum_{j\leq K} \mathcal{E}_{N+1,j} \right) \prod_{i\leq N} \sum_{j\leq K} \mathcal{E}_{i,j} \\
&= \prod_{i\leq N+1} \sum_{j\leq K} \mathcal{E}_{i,j}.
\end{aligned}
$$

□

Theorem A.1.1 (Posterior factorization)**.** *The posterior distribution for a classification problem can be factorized as follows:*

$$
\mathbf{P}^c(\theta \mid \boldsymbol{x}) = \prod_{i}^{N} \mathbf{P}^c(\theta_i \mid \boldsymbol{x}) = \prod_{i=1}^{N} \frac{\exp\left(\beta F_{\theta_i}(x_i)\right)}{\sum_{j=1}^{K} \exp\left(\beta F_j(x_i)\right)}
$$

Proof. The posterior distribution solution to the MAP problem is the following:

$$
\mathbf{P}^c(\theta \mid \boldsymbol{x}) \frac{\exp\left(\beta \sum_{i=1}^{N} F_{\theta_i}(x_i)\right)}{\sum_{\theta\in\Theta} \exp\left(\beta \sum_{i=1}^{N} F_{\theta_i}(x_i)\right)} = \frac{\prod_{i=1}^{N} \exp\left(\beta F_{\theta_i}(x_i)\right)}{\sum_{\theta\in\Theta} \prod_{i=1}^{N} \exp\left(\beta F_{\theta_i}(x_i)\right)}
$$

Using Lemma 3.3.1 we can rewrite the denominator as:

$$
\sum_{\theta\in\Theta} \prod_{i=1}^{N} \exp\left(\beta F_{\theta_i}(x_i)\right) = \prod_{i=1}^{N} \sum_{\theta\in\Theta} \exp\left(\beta F_{\theta_i}(x_i)\right)
$$

Therefore, the posterior distribution can be written as:

$$
\mathbf{P}^c(\theta \mid \boldsymbol{x}) = \prod_{i=1}^{N} \mathbf{P}^c(\theta_i \mid \boldsymbol{x}) = \prod_{i=1}^{N} \frac{\exp\left(\beta F_{\theta_i}(x_i)\right)}{\sum_{\theta\in\Theta} \exp\left(\beta R(\theta, \boldsymbol{x})\right)}
$$

□

A.2 Properties of the PA kernel

Theorem A.2.1 (Symmetry of the PA kernel)**.** *The posterior agreement kernel is symmetric with respect to the definition of X' and X''.*

$$PA(\boldsymbol{x}', \boldsymbol{x}'') = PA(\boldsymbol{x}'', \boldsymbol{x}')$$

Theorem A.2.2 (Non-negativity of the PA kernel)**.** *The posterior agreement kernel is non-negative.*

$$PA(\boldsymbol{x}', \boldsymbol{x}'') \geq 0$$

Theorem A.2.3 (Concavity of the PA kernel)**.** *The posterior agreement kernel is concave in $\mathbb{R}^+$, and therefore has a unique maximum.*

Proof. The posterior agreement kernel has been shown to have the following form:

$$\mathrm{PA}\left(\boldsymbol{x}', \boldsymbol{x}''\right) \propto \sum_{n=1}^{N} \log \left[\sum_{j=1}^{K} \mathbf{P}_n^c(\theta \mid x_n') \mathbf{P}_n^c(\theta \mid x_n'') \right]$$

where the posteriors $\mathbf{P}_n^c(\theta \mid x_n)$ are Gibbs distributions for each observation.

$$\mathbf{P}_n^c(\theta \mid x_n') = \frac{e^{\beta F_j(x_n)}}{\sum_{k=1}^{K} e^{\beta F_k(x_n)}}$$

We will require three important results from optimization theory:

T1 The minimum of $G(\beta) = -\,\mathrm{PA}(X', X'')$ over the convex set $\mathbb{R}^+$ is unique $\iff$ $G(\beta)$ is convex.

T2 G is absolutely convex $\iff$ $\frac{d^2}{d\beta^2} G(\beta) > 0$.

T3 The sum of convex functions is also convex.

To streamline the derivation, the following notation will be used:

$$F_j(x_n') = F_j'$$

$$e^{\beta F_j(x_n')} = e^{\beta F_j'} = e_j'$$

The observation index n will be omitted as it does not affect the convexity derivation (see **T3**). With that notation in mind, we can define $G(\beta)$ properly:

$$G(\beta) = -k(\boldsymbol{x}', \boldsymbol{x}'') = \sum_{n=1}^{N} -\log \left[\sum_{j=1}^{K} e_j' e_j'' \right] + \sum_{n=1}^{N} \log \left[\sum_{k=1}^{K} e_k' \sum_{p=1}^{K} e_p'' \right]$$

We will focus on the first term: $G_1^n(\beta) = G_1(\beta) = \log \left[\sum_{j=1}^{K} e_j' e_j'' \right]$.

$$\frac{d}{d\beta} G_1(\beta) = \frac{\sum_{j=1}^{K} (F_j' + F_j'') e_j' e_j''}{\sum_{j=1}^{K} e_j' e_j''}$$

The derivative $\frac{d}{d\beta} e_j' e_k''$ will be used recurrently in this section:

$$\frac{d}{d\beta} e_j' e_k'' = F_j' e_j' e_k'' + e_j' F_k'' e_k'' = (F_j' + F_k'') e_j' e_k''$$

The second derivative is straightforward:

$$\frac{d^2}{d\beta^2}G_1(\beta) = \frac{\sum_{j=1}^{K}(F'_j + F''_j)^2 e'_j e''_j}{\sum_{j=1}^{K} e'_j e''_j} - \frac{\left(\sum_{j=1}^{K}(F'_j + F''_j) e'_j e''_j\right)^2}{\left(\sum_{j=1}^{K} e'_j e''_j\right)^2}$$

We impose the convexity condition and see whether it can be contradicted.

$$\frac{d^2}{d\beta^2}G_1(\beta) > 0 \iff \left(\sum_{j=1}^{K} e'_j e''_j\right)\left(\sum_{j=1}^{K}(F'_j + F''_j)^2 e'_j e''_j\right) - \left(\sum_{j=1}^{K}(F'_j + F''_j) e'_j e''_j\right)^2 > 0$$

Using the distributive property of the product over the sum, we can reindex our expression:

$$\sum_{k=1}^{K}\sum_{j=1}^{K}(F'_j + F''_j)^2 e'_j e''_j e'_k e''_k - \sum_{k=1}^{K}\sum_{j=1}^{K}(F'_j + F''_j)(F'_k + F''_k) e'_j e''_j e'_k e''_k > 0 \iff$$

$$\iff \sum_{k=1}^{K}\sum_{j=1}^{K}[(F'_j + F''_j) - (F'_k + F''_k)](F'_j + F''_j) e'_j e''_j e'_k e''_k > 0$$

As we can see, $\Delta_{(jj),(kk)}$ corresponds to the difference in the cost attributed to reference class j and the cost attributed to class k, accumulated over $\boldsymbol{x}', \boldsymbol{x}''$. We can intuitively devise some symmetry in these terms, and we formalize it as follows:

$$E_{jk} = e'_j e''_j e'_k e''_k = E_{kj}$$

$$\Delta_{(jj),(kk)} = (F'_j + F''_j) - (F'_k + F''_k) = (F'_j - F'_k) + (F''_j - F''_k) = -\Delta_{(kk),(jj)}$$

Even if $\Delta_{(jj),(jj)} = 0$, we will still include this term to facilitate with the indexing. Overall, the sum can be expressed as:

$$\sum_{k=1}^{K}\sum_{j=1}^{K}[(F'_j+F''_j)-(F'_k+F''_k)](F'_j+F''_j) e'_j e''_j e'_k e''_k = \sum_{k=1}^{K}\sum_{j=1}^{K}(F'_j+F''_j) E_{jk}\Delta_{(jj),(kk)} = \sum_{k=1}^{K}\sum_{j=1}^{K} S_{(jj),(kk)}$$

Then, the pairwise sum of symmetric combinations of indexes k and j yields

$$\begin{aligned} S_{(jj),(kk)} + S_{(kk),(jj)} &= (F'_j + F''_j) E_{jk}\Delta_{(jj),(kk)} + (F'_k + F''_k) E_{kj}\Delta_{(kk),(jj)} \\ &= E_{jk}\Delta_{(jj),(kk)}[(F'_j + F''_j) - (F'_k + F''_k)] = E_{jk}\Delta^2_{(jj),(kk)} > 0 \end{aligned}$$

Given that the indexing sets in our nested sum are the same, it's straightforward to see that all the terms will be strictly positive, and the overall sum will be zero only if $e_j = 0\ \forall j = \{1, ...K\}$, which is not possible in a classification setting since $\beta \in \mathbb{R}^+$. We end up with the following expression:

$$\frac{d^2}{d\beta^2}G_1(\beta) = \sum_{k=1}^{K}\sum_{j<k}^{K} E_{jk}\Delta^2_{(jj),(kk)} > 0$$

Now we proceed analogously with the second term:

$$G_2^n(\beta) = G_2(\beta) = \log\left[\sum_{j=1}^{K} e'_j \sum_{k=1}^{K} e''_k\right] = \log\left[\sum_{k=1}^{K}\sum_{j=1}^{K} e'_j e''_k\right]$$

$$\frac{d}{d\beta}G_2(\beta) = \frac{\sum_{k=1}^{K}\sum_{j=1}^{K}(F'_j + F''_k) e'_j e''_k}{\sum_{k=1}^{K}\sum_{j=1}^{K} e'_j e''_k}$$

$$\frac{d^2}{d^2\beta}G_2(\beta) = \frac{\sum_{k=1}^{K}\sum_{j=1}^{K}(F'_j + F''_k)^2 e'_j e''_k}{\sum_{k=1}^{K}\sum_{j=1}^{K} e'_j e''_k} - \frac{\left(\sum_{k=1}^{K}\sum_{j=1}^{K}(F'_j + F''_k) e'_j e''_k\right)^2}{\left(\sum_{k=1}^{K}\sum_{j=1}^{K} e'_j e''_k\right)^2} > 0 \iff$$

$$\iff \left(\sum_{k=1}^{K}\sum_{j=1}^{K} e'_j e''_k\right)\left(\sum_{k=1}^{K}\sum_{j=1}^{K}(F'_j + F''_k)^2 e'_j e''_k\right) - \left(\sum_{k=1}^{K}\sum_{j=1}^{K}(F'_j + F''_k) e'_j e''_k\right)^2 > 0 \iff$$

$$\iff \sum_{k=1}^{K}\sum_{q=1}^{K}\sum_{j=1}^{K}\sum_{i=1}^{K}(F'_j + F''_k)^2 e'_j e''_k e'_i e''_q - (F'_j + F''_k) e'_j e''_k (F'_i + F''_q) e'_i e''_q > 0 \iff$$

$$\iff \sum_{k=1}^{K}\sum_{q=1}^{K}\sum_{j=1}^{K}\sum_{i=1}^{K}(F'_j + F''_k) e'_j e''_k e'_i e''_q [(F'_j + F''_k) - (F'_i + F''_q)] > 0$$

We can define as well:

$$\frac{d^2}{d^2\beta}G_2(\beta) = \sum_{k=1}^{K}\sum_{q=1}^{K}\sum_{j=1}^{K}\sum_{i=1}^{K} S_{(jk),(iq)} = \sum_{k=1}^{K}\sum_{q=1}^{K}\sum_{j=1}^{K}\sum_{i=1}^{K}(F'_j + F''_k) E_{(jk),(iq)} \Delta_{(jk),(iq)}$$

$$E_{(jk),(iq)} = e'_j e''_k e'_i e''_q = E_{(ik),(jq)} = E_{(jq),(ik)} = E_{(iq),(jk)}$$

$$\Delta_{(jk),(iq)} = (F'_j - F'_i) + (F''_k - F''_q) = -\Delta_{(iq),(jk)}$$

The symmetry arises when adding two elements that have mirror indexes in both $\boldsymbol{x}'$ and $\boldsymbol{x}''$.

$$\begin{aligned} S_{(jk),(iq)} + S_{(iq),(jk)} &= (F'_j + F''_k) E_{(jk),(iq)} \Delta_{(jk),(iq)} + (F'_i + F''_q) E_{(iq),(jk)} \Delta_{(iq),(jk)} \\ &= E_{(jk),(iq)} \Delta_{(jk),(iq)} [(F'_j + F''_k) - (F'_i + F''_q)] = E_{(jk),(iq)} \Delta^2_{(jk),(iq)} > 0 \end{aligned}$$

Given that symmetries are independent for $\boldsymbol{x}'$ and $\boldsymbol{x}''$, we end up with a similar expression:

$$\frac{d^2}{d\beta^2}G_2(\beta) = \sum_{k=1}^{K}\sum_{q<k}^{K}\sum_{j=1}^{K}\sum_{i<j}^{K} E_{(jk),(iq)} \Delta^2_{(jk),(iq)} > 0$$

Even if a further simplified version can be obtained, this one will allow us to complete the proof. We can now define the function $G(\beta)$ as the sum of the two terms:

$$\frac{d^2}{d\beta^2}G(\beta) = \sum_{n=1}^{N}\left[\sum_{k=1}^{K}\sum_{q<k}^{K}\sum_{j=1}^{K}\sum_{i<j}^{K} E_{(jk),(iq)} \Delta^2_{(jk),(iq)} - \sum_{k=1}^{K}\sum_{q<k}^{K} E_{(qq),(kk)} \Delta^2_{(qq),(kk)}\right]$$

where we can clearly see that the particular case $\{k = j, q = i\}$ cancels the negative terms:

$$\begin{aligned} \frac{d^2}{d\beta^2}F^n(\beta) &= \sum_{k=1}^{K}\sum_{q<k}^{K}\sum_{j=\{1:K\}\setminus\{k\}}\sum_{i=\{1:K|i<j\}\setminus\{k\}} E_{(jk),(iq)} \Delta^2_{(jk),(iq)} \\ &+ \sum_{k=1}^{K}\sum_{q<k}^{K} E_{(qq),(kk)} \Delta^2_{(qq),(kk)} - \sum_{k=1}^{K}\sum_{q<k}^{K} E_{(qq),(kk)} \Delta^2_{(qq),(kk)} = \\ &= \sum_{k=1}^{K}\sum_{q<k}^{K}\sum_{j=\{1:K\}\setminus\{k\}}\sum_{i=\{1:K|i<j\}\setminus\{k\}} E_{(jk),(iq)} \Delta^2_{(jk),(iq)} > 0 \end{aligned}$$

Which proves that $G(\beta)$ is absolutely convex in $\mathbb{R}^+$:

$$\frac{d^2}{d\beta^2}G(\beta) = \sum_{n=1}^{N} \frac{d^2}{d\beta^2}G^n(\beta) = \sum_{n=1}^{N} \left[\sum_{k=1}^{K} \sum_{q<k}^{K} \sum_{j=\{1:K\}\setminus\{k\}} \sum_{i=\{1:K|i<j\}\setminus\{k\}} E_{(jk),(iq)}\Delta^2_{(jk),(iq)} \right] > 0$$

We must note that on the limit $\beta \to \infty$ the curvature is not defined, so it will be always a good practice to start the numerical procedure at a value $\beta_0 = 0^+$:

$$\lim_{\beta\to 0^+} \frac{d^2}{d\beta^2}G(\beta) > 0$$

□

Appendix B

Again Something

Blah, blah ...

Bibliography

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant Risk Minimization.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN.

[3] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent Advances in Adversarial Training for Adversarial Robustness.

[4] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from Several Related Classification Tasks to a New Unlabeled Sample.

[5] Joachim M. Buhmann. Data Science Algorithms and the Rate-Distortion Tradeoff.

[6] Joachim M. Buhmann. Information theoretic model validation for clustering.

[7] Joachim M. Buhmann. Posterior Agreement for Model Robustness Assessment in Covariate Shift Scenarios.

[8] Joachim M Buhmann, Morteza Haghir Chehreghani, Mario Frank, and Andreas P Streich. Information Theoretic Model Selection for Pattern Analysis.

[9] Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks.

[10] George Casella and Roger L. Berger. *Statistical Inference*. Wadsworth Group Duxbury, second edition.

[11] Morteza Haghir Chehreghani, Alberto Giovanni Busetto, and Joachim M Buhmann. Information Theoretic Model Validation for Spectral Clustering.

[12] Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified Adversarial Robustness via Randomized Smoothing.

[13] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. RobustBench: A standardized adversarial robustness benchmark.

[14] Elias Eulig, Piyapat Saranrittichai, Chaithanya Kumar Mummadi, Kilian Rambach, William Beluch, Xiahan Shi, and Volker Fischer. DiagViB-6: A Diagnostic Benchmark Suite for Vision Models in the Presence of Shortcut and Generalization Opportunities.

[15] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples.

[16] Peter Grünwald and Teemu Roos. Minimum Description Length Revisited. 11(01):1930001.

[17] Jun Guo, Wei Bao, Jiakai Wang, Yuqing Ma, Xinghai Gao, Gang Xiao, Aishan Liu, Jian Dong, Xianglong Liu, and Wenjun Wu. A comprehensive evaluation framework for deep model robustness. 137:109308.

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models.

[19] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial Examples Are Not Bugs, They Are Features.

[20] Ortiz Jimenez. The inductive bias of deep learning: Connecting weights and functions.

[21] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A. Efros, and Antonio Torralba. Undoing the Damage of Dataset Bias. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, volume 7572, pages 158–171. Springer Berlin Heidelberg.

[22] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A Benchmark of in-the-Wild Distribution Shifts.

[23] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to Generalize: Meta-Learning for Domain Generalization. 32(1).

[24] Yao Li, Minhao Cheng, Cho-Jui Hsieh, and Thomas C. M. Lee. A Review of Adversarial Attack and Defense for Classification Methods. 76(4):329–345.

[25] Jian Liang, Ran He, and Tieniu Tan. A Comprehensive Survey on Test-Time Adaptation under Distribution Shifts.

[26] Jiashuo Liu, Zheyan Shen, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards Out-Of-Distribution Generalization: A Survey.

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks.

[28] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning.

[29] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain Generalization via Invariant Feature Representation.

[30] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Second edition.

[31] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-Adversarial Domain Adaptation.

[32] Maura Pintor, Fabio Roli, Wieland Brendel, and Battista Biggio. Fast Minimum-norm Adversarial Attacks through Adaptive Norm Constraints.

[33] Joaquin Quiñonero-Candela, editor. *Dataset Shift in Machine Learning*. Neural Information Processing Series. MIT Press.

[34] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Mądry. Adversarially Robust Generalization Requires More Data.

[35] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein Distance Guided Representation Learning for Domain Adaptation.

[36] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition.

[37] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks.

[38] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE.

[39] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness May Be at Odds with Accuracy.

[40] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep Learning for Computer Vision: A Brief Review. 2018:1–13.

[41] Chengyu Wang, Minghui Qiu, Jun Huang, and Xiaofeng He. Meta Fine-Tuning Neural Language Models for Multi-Domain Text Mining.

[42] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. Generalizing to Unseen Domains: A Survey on Domain Generalization.

[43] Yang Wang, Bo Dong, Ke Xu, Haiyin Piao, Yufei Ding, Baocai Yin, and Xin Yang. A Geometrical Approach to Evaluate the Adversarial Robustness of Deep Neural Networks. 19:1–17.

[44] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. 13(4):600–612.

[45] Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better Diffusion Models Further Improve Adversarial Training.

[46] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach.

[47] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating Adversarial Examples with Adversarial Networks.

[48] Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving Out-of-Distribution Robustness via Selective Augmentation.

[49] Samuel Yu, Peter Wu, Paul Pu Liang, Ruslan Salakhutdinov, and Louis-Philippe Morency. PACS: A Dataset for Physical Audiovisual CommonSense Reasoning.

[50] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features.

[51] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically Principled Trade-off between Robustness and Accuracy.

[52] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. Mixup: Beyond Empirical Risk Minimization.

[53] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain Generalization: A Survey. pages 1–20.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute for Dynamic Systems and Control
Prof. Dr. R. D'Andrea, Prof. Dr. L. Guzzella

Title of work:

Improved robustness of deep learning models through posterior agreement-based model selection

Thesis type and date:

Master Thesis, September 2024

Supervision:

Dr. Joao Borges de Sa Carvalho, Dr. Alessandro Torcinovich
Prof. Dr. Joachim M. Buhmann

Student:

Name:	Victor Jimenez Rodriguez
E-mail:	vjimenez@student.ethz.ch
Legi-Nr.:	97-906-739
Semester:	5

Statement regarding plagiarism:

By signing this statement, I affirm that I have read and signed the Declaration of Originality, independently produced this paper, and adhered to the general practice of source citation in this subject-area.

Declaration of Originality:

`http://www.ethz.ch/faculty/exams/plagiarism/confirmation_en.pdf`

Zurich, 9. 5. 2024: _____