THÈSE DE DOCTORAT DE SORBONNE UNIVERSITÉ
Spécialité **Informatique**
École Doctorale Informatique, Télécommunications et Électronique (Paris)

Diverse and Efficient Ensembling of Deep Networks

De la diversification et combinaison efficace des réseaux profonds

Présentée par
Alexandre Ramé

Dirigée par
Matthieu Cord

Pour obtenir le grade de
DOCTEUR de SORBONNE UNIVERSITÉ

Présentée et soutenue publiquement le 11 Octobre 2023

Devant le jury composé de :

Graham Taylor *Professor, University of Guelph. Research Director, Vector Institute.*	Rapporteur
Christian Wolf *Principal Scientist, Naver Labs.*	Rapporteur
Cordelia Schmid *Research Director, INRIA. Research Scientist, GoogleAI.*	Examinatrice
Léon Bottou *Principal Scientist, MetaAI.*	Examinateur
Thomas Wolf *Co-founder, HuggingFace.*	Examinateur
Patrick Gallinari *Professeur, Sorbonne Université. Senior Research Scientist, Criteo.*	Examinateur
Matthieu Cord *Professeur, Sorbonne Université. Principal Scientist, valeo.ai.*	Directeur de thèse

ABSTRACT

This thesis aims at enhancing the generalization abilities of deep neural networks, a critical step towards fair and reliable artificial intelligence. Specifically, we address the drop in performance when models are evaluated on test samples with a distribution shift with respect to the train samples.

To this end, we focus on ensembling strategies: indeed, combining multiple models is a standard, simple yet potent strategy to improve robustness. After an overview of the relevant literature, we provide a new explanation of ensembling's success under distribution shifts, especially when the members of the ensemble are diverse.

To foster such diversity within members, we investigate several strategies. The initial one, DICE, introduces an explicit regularization to eliminate redundant information across members. Subsequent diversity methods in this thesis are implicit, relying on diverse data augmentation (in MixMo), diverse hyperparameters (in DiWA), inter-training on auxiliary datasets (in ratatouille), and diverse objectives (in rewarded soups).

The second primary challenge addressed in this thesis is the enhancement of ensemble efficiency, and aims at lessening the computational burden of combining multiple models; indeed, when considering two members, the standard ensembling by averaging of predictions doubles the computational cost, thus impeding scalability. After exploring subnetwork ensembling (in MixMo), we introduce a significant contribution of this thesis; the observed ability to average the models in weights rather than in predictions. This finding was surprising due to the non-linearities in deep architectures. We empirically demonstrate that, when weights are fine-tuned from a shared pre-trained initialization, weight averaging succeeds by approximating ensembling without any inference overhead. The empirical gains are especially important on DomainBed, the reference benchmark evaluating out-of-distribution generalization. More broadly, weight averaging facilitates effortless parallelization, enhancing machine learning updatability and data privacy.

Finally, this thesis explores how ensembling can facilitate the alignment of models. This is critical to mitigate the societal ethical concerns from recent rapid scale-up in deep learning. To this end, we propose rewarded soups, a new strategy for multi-objective reinforcement learning, paving the way towards more transparent and reliable artifical intelligences, aligned with the world in all its diversity.

RÉSUMÉ

Cette thèse vise à améliorer les capacités de généralisation des réseaux de neurones profonds, un enjeu essentiel pour le développement de systèmes d'intelligence artificielle à la fois fiables et équitables. Le coeur du défi réside dans la gestion des potentiels changements de distributions entre les données d'entraînement et celles de test pour l'évaluation, pouvant réduire les performances.

Dans cette thèse, nous analysons principalement des stratégies consistant à combiner plusieurs réseaux de neurones. Cette simple méthode d'ensemble est classique mais particulièrement efficace pour améliorer la généralisation. Après avoir examiné la littérature existante, nous proposons une nouvelle explication de la réussite des méthodes d'ensemble hors-distribution, en particulier lorsque les différents membres de l'ensemble sont suffisamment divers pour compenser mutuellement leurs erreurs individuelles.

Pour encourager cette diversité entre les membres de l'ensemble, nous explorons plusieurs approches. La première, baptisée DICE, introduit explicitement une régularisation visant à éliminer de potentielles informations redondantes entre les membres de l'ensemble lors de l'apprentissage. Les autres méthodes de diversité utilisées dans cette thèse sont implicites, s'appuyant sur une augmentation diversifiée des données (dans MixMo), le choix d'hyperparamètres variés (dans DiWA), des entraînements intermédiaires sur des jeux de données auxiliaires (dans ratatouille), ou des récompenses différenciées en apprentissage par renforcement (dans rewarded soups).

Le second enjeu majeur de cette thèse concerne l'efficacité des méthodes d'ensemble. L'objectif est d'atténuer le coût computationnel inhérent à la combinaison de plusieurs réseaux ; en effet, considérant deux réseaux, la méthode standard qui consiste à moyenner leurs prédictions multiplie par deux le coût. Après avoir exploré une stratégie d'ensemble de sous-réseaux (dans MixMo), nous décrivons une contribution majeure de cette thèse : l'analyse d'une stratégie consistant à faire la moyenne des poids des réseaux de neurones plutôt que de leurs prédictions. Cette stratégie, surprenante en raison des non-linéarités des architectures profondes, fonctionne empiriquement lorsque les modèles sont appris à partir d'une même initialisation pré-entraînée. Cette moyenne des poids offre les bénéfices de l'ensemble sans aucun coût supplémentaire pour l'évaluation, en particulier sur DomainBed, le benchmark de référence pour évaluer la généralisation hors-distribution. Plus généralement, cette stratégie favorise la parallélisation des apprentissages et l'adaptabilité des modèles.

Pour finir, cette thèse explore comment les méthodes d'ensemble peuvent améliorer l'alignement des intelligences artificielles. Face à l'essor rapide des modèles de langues comme ChatGPT, cet alignement est crucial pour répondre aux nombreuses préoccupations éthiques actuelles. Pour prendre en compte la diversité des préférences humaines, nous proposons une stratégie de politiques multiples en apprentissage par renforcement, rendant plus transparent l'alignement sur le monde, dans toute sa diversité.

REMERCIEMENTS

Je souhaite exprimer ma plus profonde gratitude à toutes les personnes qui m'ont soutenu et guidé lors de ce parcours doctoral, professionnel et personnel.

Je commencerai par remercier sincèrement Matthieu, dont l'accompagnement exceptionnel a marqué ces trois années de recherche. Son optimisme constant, son réalisme avisé, son habile mélange de confiance et de questionnement scientifique, sa rigueur intellectuelle et sa bienveillance, ont contribué à faire de cette thèse une grande expérience humaine et professionnelle. Je tiens également à exprimer ma gratitude au jury de thèse : leur investissement dans la lecture de ce manuscrit et lors de ma soutenance est un grand honneur. Je suis touché par leur intérêt dans mon travail.

Je tiens également à remercier l'ensemble de l'équipe MLIA. Les réunions mais surtout les échanges informels, dans les couloirs ou lors des déjeuners, ont toujours été enrichissants et amicaux. Je pense particulièrement aux *Chordettes*, Corentin Dancette, les frères Couairon, Arthur Douillard, Mustafa Shukor, Asya Grechka, Rémy Sun, et à mes autres collaborateurs, notamment Matthieu Kirchmeyer et Jean Baptiste Gaya, avec qui j'ai adoré travailler. Notre dernier papier, *rewarded soups*, est une belle illustration de ce que nous avons accompli ensemble et qui n'aurait pas été possible seul, en une belle analogie de mes travaux de recherche sur l'ensembling, où la diversité des profils permet de dépasser les limites de chacun. Je pense que nous pouvons être fiers de notre laboratoire.

Je voudrais également remercier mes anciens collègues d'Heuritech, où j'ai découvert le monde du deep learning. Grâce à vous, j'ai pu arriver à ce travail de thèse avec pléthore d'idées. Charles, Tony, Didier, Paul, Sylvain, merci pour votre confiance. Un grand merci également à mes anciens collaborateurs de la R&D : Emilien, Arthur, Charles, Laura et Hedi. La *mafia d'Heuritech* a formidablement grandi. Je suis également extrêmement reconnaissant envers David Lopez-Paz pour le stage chez Meta, une expérience qui m'a totalement immergé dans un environnement de travail stimulant. Mes remerciements vont également à Kartik, qui a toujours su saisir l'essence de mes idées en un éclair.

Finalement, mes remerciements les plus chaleureux vont à mes proches. Tout d'abord à ma (désormais) épouse Julie. Son soutien indéfectible et son intelligence font de Julie une relectrice hors pair, mais aussi une interlocutrice extraordinaire ; certaines idées présentées dans cette thèse sont nées de nos discussions. D'ailleurs, n'hésitez pas à lui demander si vous avez besoin d'éclaircissements sur mon travail ! Je tiens également à souligner la chance que j'ai eue de grandir dans un cadre familial stimulant et bienveillant, favorisant l'épanouissement intellectuel. Pour cela et pour beaucoup plus, un grand merci à mes incroyables parents, Florence et Emmanuel, à mes sœurs, Valentine et Atalante, et mon frère Nicolas. À mes amis proches et témoins de mariage Théo, Thibaud, et Matthieu, à mes amis de Boulogne et de l'X, Rouille pour ses ronronnements, France Culture et Nicolas Martin pour ses podcasts, et à tous ceux et celles avec qui j'ai partagé des conversations enrichissantes, ou simplement passé des moments de qualité : merci !

CONTENTS

Chapter 1

INTRODUCTION

L'union fait la force

Homère, L'Iliade

1.1 Artificial intelligence: a historical perspective

The concept of **artificial intelligence (AI) machines** traces back to the previous century, beginning as a subject of science fiction, with Karel Čapek first coining the term "robot" in 1920 and Isaac Asimov exploring the concept in short stories from the 1940s. The notion then evolved into a tangible reality, from Turing's machine-breaking work during World War II, the seminal 1956 Dartmouth Workshop to the triumph of IBM's Deep Blue in 1997 [Cam+02]. Initially, the community was optimistic about achieving AI within a few years.

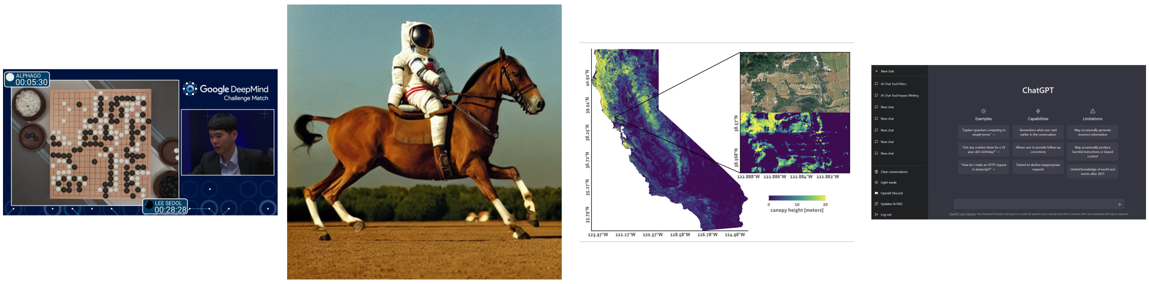
A successful AI revolution could usher a new era of enhanced intellectual capabilities, much like the industrial revolution enhanced human physical capabilities. The potential applications are wide-ranging, from education, healthcare, personal assistants, autonomous vehicles to AI-driven scientific research for addressing global challenges like climate change. Optimistic visionaries anticipate a world of material and cultural abundance, universal income, and freedom from some labor, enabling humans to focus on selected/creative/interesting tasks.

Despite early optimism, traditional expert systems encountered numerous setbacks, leading to periods known as "AI winters" filled with disillusionment and skepticism. In response, **machine learning (ML)** [Bis06] proposed a data-centric paradigm shift. The key element in ML is the statistical algorithm that learns automatically through experience and data. More specifically, models are learned for a target task using some training data, intending to achieve good accuracy (*i.e.*, generalize well) on new samples not seen during training. While early and simple ML models achieved some success in tasks like spam filtering, genome-wide association studies, movie recommendation [Tös+09], and medical diagnosis, they often fell short in large-scale real-world applications.

1.2 Deep learning

The prominence of AI in today's society is primarily caused by the success of deep learning (DL) [Goo+16], a specialized subfield of ML involving deep neural networks (DNNs). These models were biologically inspired by the human brain, yet actually are simply a collection of nested (neuron) functions, each of which is a linear transformation usually followed by a non-linear activation function. The term *deep* refers to the large number of layers stacked successively. DL became the dominant approach in computer vision (CV) after the success in 2012 of AlexNet [Kri+12] at the ILSVC competition [Rus+15] on ImageNet, a large dataset made of 1.28M training images and 1000 classes.

Then, DL powered breakthroughs to problems that seemed unsolvable by traditional ML, some of which are depicted in Figure 1.1. Stable Diffusion [Rom+22] can generate realistic photographs of the pope in a down puffer coat. ChatGPT [Ope23] could pass the bar exam and write 40 per cent of the code for a software engineer. Beyond these famous applications, DL is the core paradigm behind current high-precision machine translation [Vas+17], autonomous driving systems [Sun+20], accelerating physics simulations [Bre+20], predicting protein folding structures [Jum+21], estimating molecular toxicity [Adv14], optimizing data center cooling systems [Eva+16], and managing magnetic coils in nuclear fusion reactors [Deg+22], etc, to name a few.



(a) Alphago. (b) Stable Diffusion. (c) Canopy. (d) ChatGPT.

Figure 1.1. – A glimpse into the transformative and successful applications of DL. In Figure 1.1(a), AlphaGo [Sil+16] surpasses human performance in the strategic board game of Go. In Figure 1.1(b), Stable Diffusion [Rom+22] is prompted to generate "a photograph of an astronaut riding a horse", illustrating how AIs can imitate human creativity. In Figure 1.1(c), sub-meter resolution canopy height maps are generated from aerial and GEDI lidar data [Tol+23], offering climate scientists a powerful tool to understand deforestation. In Figure 1.1(d), the now famous ChatGPT [Ope23] can natively discuss with humans and augment their intelligence.

Scaling. The cornerstone of the DL revolution is scalability: results have consistently improved when increasing the number of parameters [Kap+20] or of training data [Hof+22], without hitting any apparent limitations. Such rapid progress was enabled by engineering advances in graphics processing units (GPUs) hardware adhering to Moore's law, the advent of cloud computing, more efficient algorithms, and the emergence of frameworks like PyTorch [Pas+19] or platforms like HuggingFace [Wol+20]. The field has benefited from substantial investments from major tech companies and from numerous collaborations fostered by the prevalent open-source philosophy. Over the past decade, the computational power used for training AI models has surged by a factor of one hundred million, while the cost has significantly diminished. We have transitioned from training models on relatively small datasets to feeding them with the entire internet's wealth of information. This scaling up predictably enhances results across a broad spectrum of tasks, but also induced unpredicted *emergent* abilities [Wei+22b] surfacing abruptly in (very) large models: examples include in-context learning, or multi-step reasoning [Wei+22c; Lam+22]. These emergent properties underscore the success of ChatGPT and further drive the scaling trend. They also raise questions about the potential capabilities and risks of LLMs, should we continue to scale them up.

Unified framework. Another paradigm that has significantly contributed to the success of DL is the *unification* of modalities and tasks under a single & simple framework. For example, the networks' architectures are now unified; though historically computer vision used convolutional [Fuk80; Kri+12] and natural language processing recurrent networks [Rum+85], most state-of-the-art are now with Transformers [Vas+17; Dos+21]. This architecture was first designed for machine translation. Moreover, *foundation models* [Bom+21] have emerged as the standard unified paradigm to learn DNN's weights: rather than trained from scratch on the target task, weights are now first pre-trained through self-supervision on vast corpus of data and then fine-tuned [Oqu+14; Yos+14] on the target task. The different trainings usually leverage the same tools and "tricks", such as the Adam [Kin+15] optimizer and general-purpose regularizations (dropout [Gal+16] or weight decay [Kro+91]). This unifying paradigm enables the transfer of knowledge and findings across DL setups.

1.3 Out-of-distribution generalization

The increasing importance of AI in our lives and society demands a close examination of potential failure cases. As large DNNs can actually *memorize* their training dataset [Zha+17] by latching onto specific patterns, they may struggle to generalize on new test samples. The *generalization gap* [Kaw+17] between train and test performance is a well-known phenomenon in ML usually named *overfitting*. Critically, this generalization gap worsens under distribution shifts [Arj+19; Gul+21], when the test distribution differs from the training distribution. These failures for out-of-distribution (OOD) generalization is a major limitation of DL, which can negatively impact human lives in several real-world contexts. We name a few below.

- Uncontrolled deployment: although training is centralized, the model is used in the wild for diverse scenarios where the input distribution can change unexpectedly.
- Constrained budget, causing training datasets of limited size without representing all possible domains. For example, it would be impractical to train an autonomous vehicle's object detection system on all possible weather conditions and in all cities.

- Dynamic scenarios, with a temporal shift between train and test. For example, when analyzing images from social media, where trends can change rapidly.

- Subpopulation shifts, as some (minority) groups may be less represented in the training dataset. This highlights a critical connection between OOD generalization and *fairness*. Indeed, errors induced by distribution shifts can lead to severe *ethical* issues: examples include facial recognition systems [Gro+19], loan approvals [Ang+21], or medical diagnoses [Lar+20]. Not only might DNNs replicate the biases found in the data—they confuse correlation and causation—but they also exhibit a simplicity bias [Sha+20]: DNNs tend to rely on the most straightforward features, potentially overlooking fairer explanations.

More broadly, the ability of a model to effectively handle OOD scenarios is a key indicator of its intelligence. Our cognitive abilities as humans allows us to create beyond our previous experiences, to solve new problems, and to find solutions when facing novel tasks in unseen situations. Therefore, OOD generalization appears as a necessary milestone in the journey towards more *general* systems.

The imperative of alignment. Assuming we continue to scale the architectures and improve DL performances in the future, we may then be dealing with potential new risks. Such powerful models could arm malicious users with unprecedented power, fuel widespread censorship or manipulation by totalitarian private entities or governments, or, more speculatively, seize global control [Hen+22; Hen23], leading to catastrophic outcomes. It appears necessary to ensure that the models remain aligned [Tay+16; Ken+21; Ngo+22] with core values. Recently, reinforcement learning from human feedback (RLHF) [Chr+17b] emerged as the leading alignment paradigm to fine-tune large language models (LLMs). Specifically, given a proxy reward approximating human preferences, a LLM optimized on this proxy reward is expected to make decisions for new inputs aligned with those preferences. Thus, the LLM needs to handle a wide range of OOD inputs, emphasizing that *alignment is fundamentally an OOD challenge*, as further detailed in Section 7.2.5.3. Given the importance of alignment for AI safety, this strengthens the need for developing DL models that can effectively deal with unanticipated inputs.

1.4 Ensembling and contributions

This thesis seeks to improve the generalization capabilities of deep neural networks. To this end, we turn to a traditional, straightforward, and highly practical strategy: ensembling. In essence, instead of relying on a single model, we combine multiple models. From a scaling perspective [Chi+20], rather than increasing the number of parameters inside one model, we scale the number of models.

The most common ensembling (ENS) strategy involves (i) training multiple models independently, (ii) passing the test input to each model, and (iii) averaging their predictions. Such *functional ensembling* is a very natural strategy, used for decades in ML [Nil65;

Han+90; Bre96; Die00], which still remains a standard solution for real-world applications and Kaggle competitions [Hin20].

The first key concept in ensembling is *diversity*. Indeed, ensembling succeeds if it reduces the variance of the predictions; this happens if its members are diverse and complementarity, meaning that they have different failure cases, allowing them to compensate for each other's errors. The other key concept in this thesis is *efficiency*. Indeed, the naive form of functional ensembling is costly as it requires multiple trainings and inferences, limiting its applicability in real-world scenarios.

The core challenge lies in the intricate trade-off between efficiency and diversity. In particular, traditional strategies to improve one often hurts the other. For example, let's consider a baseline *weight sharing* strategy [Lee+15] that shares weights across models; although this may improve efficiency, this would unfortunately homogenize the models and thus reduce diversity. As another example, consider members that predict randomly: these members may be highly diverse, but their ensemble would perform badly. This (naive) example highlights the importance of another criterion for accurate ensembling, *members' accuracies*, in tension with diversity and efficiency.

In other words, the principal challenge of our thesis can be summarized as follows:

How to best trade off between efficiency, diversity, and members' accuracies in ensembling?

The contributions of this thesis aim at proposing different solutions to this question, as presented in the subsequent chapters.

- Chapter 2: ENSEMBLING
  We first review the literature on ensembling, and propose a new error bound for ensembling strategies under distribution shifts. This explains why and when ensembling particularly excels, and highlights that diversity becomes even more critical under distribution shifts.
- Chapter 3: DICE [Ram+21a]
  We introduce a new regularization named DICE that increases diversity by removing irrelevant redundant information across members of the ensemble.
- Chapter 4: DIVERSE WEIGHT AVERAGING [Ram+22b]
  We present the main contribution of this thesis for efficient ensembling: instead of averaging the predictions, we propose averaging the weights of the models. We show that this weight averaging (WA) is possible (despite non-linearities in the architectures) when models are fine-tuned from a shared pre-trained initialization.
- Chapter 5: RATATOUILLE [Ram+23a]
  In this chapter, we demonstrate that the conditions to apply WA can be relaxed. The proposed ratatouille averages the weights of models fine-tuned from different initializations, inter-trained on different tasks; these inter-trainings enhance diversity across initializations and thus improve final performance for OOD generalization.
- Chapter 6: REWARDED SOUPS [Ram+23b]
  We explore how WA can help to manage the diversity of human opinions when aligning large language models with reinforcement learning from human feedback

(RLHF); rather than optimizing a single network for a given reward, we uncover a set of Pareto-optimal weights across the entire space of preferences.

In the Appendix, we detail other related works published during this thesis.

- Appendix D: MIXMO [Ram+21b]
  This chapter details our first attempt toward efficient ensembling, and proposes fitting multiple subnetworks within a single base model through a multi-input multi-output strategy named MixMo. Although successful, we move this work to the Appendix as it does not fit within the foundation model paradigm.
- Appendix E: FISHR [Ram+22a]
  This chapter considers a key limitation of ensembling strategies; their inability to tackle spurious correlations. The proposed Fishr regularization promotes invariance across training domains. We move this work to the Appendix as it does not involve ensembling.

In details, this thesis is based on the following papers, sorted in chronological order:

- Alexandre Ramé and Matthieu Cord. “DICE: Diversity in Deep Ensembles via Conditional Redundancy Adversarial Estimation”. In: *ICLR*. 2021.
- Alexandre Ramé, Remy Sun, and Matthieu Cord. “MixMo: Mixing Multiple Inputs for Multiple Outputs via Deep Subnetworks”. In: *ICCV*. 2021. The code is open-sourced: https://github.com/alexrame/mixmo-pytorch.
- Alexandre Ramé, Corentin Dancette, and Matthieu Cord. “Fishr: Invariant Gradient Variances for Out-of-Distribution Generalization”. In: *ICML*. 2022. The code is open-sourced: https://github.com/alexrame/fishr.
- Alexandre Ramé, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. “Diverse Weight Averaging for Out-of-Distribution Generalization”. In: *NeurIPS*. 2022. The code is open-sourced: https://github.com/alexrame/diwa.
- Alexandre Ramé, Kartik Ahuja, Jianyu Zhang, Matthieu Cord, Léon Bottou, and David Lopez-Paz. “Model Ratatouille: Recycling Diverse Models for Out-of-Distribution Generalization”. In: *ICML*. 2023. The code is open-sourced: https://github.com/facebookresearch/ModelRatatouille.
- Alexandre Ramé, Guillaume Couairon, Mustafa Shukor, Corentin Dancette, Jean-Baptiste Gaya, Laure Soulier, and Matthieu Cord. “Rewarded soups: towards Pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards”. In: *arXiv preprint* (2023). The code is open-sourced: https://github.com/alexrame/rewardedsoups.

Additionally, I contributed to other projects listed below; though they consider ensembling strategies and refine our understanding on important questions, they are not further investigated in this thesis for the sake of brevity.

- Rémy Sun, Alexandre Ramé, Clément Masson, Nicolas Thome, and Matthieu Cord. “Towards efficient feature sharing in MIMO architectures”. In: *CVPR Workshop*. 2022.

- Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. "DyTox: Transformers for Continual Learning with DYnamic TOken eXpansion". In: *CVPR*. 2022. The code is open-sourced: https://github.com/arthurdouillard/dytox.
- Alexandre Ramé, Jianyu Zhang, Léon Bottou, and David Lopez-Paz. "Pre-train, fine-tune, interpolate: a three-stage strategy for domain generalization". In: *NeurIPS Workshop*. 2022.
- Mustafa Shukor, Corentin Dancette, Alexandre Ramé, and Matthieu Cord. "UnIVAL: Unified Model for Image, Video, Audio and Language". In: *arXiv preprint* (2023).

Except Fishr, all those works study ensembling, where diversity and efficiency are systematically key contributions. This is highlighted in Table 1.1.

Table 1.1. – **Summary of the PhD publications**.

Name	Conference	Chapter	Diversity strategy	Efficiency strategy
DICE [Ram+21a]	ICLR 2021	Chapter 3	Diversity regularization	Weight sharing
MixMo [Ram+21b]	ICCV 2021	Appendix D	Data augmentation	Subnetworks
MixShare [Sun+22]	CVPR Workshop 2022	✗	Data augmentation + unmixing	Subnetworks
DyTox [Dou+22]	CVPR 2022	✗	Different target tasks	Weight sharing
Fishr [Ram+22a]	ICML 2022	Appendix E	✗	✗
DiWA [Ram+22b]	NeurIPS 2022	Chapters 2 and 4	Hyperparameters and data orders	Weight averaging
Interpolate [Ram+22c]	NeurIPS Workshop 2022	✗	Diverse auxiliary tasks	Weight averaging
Ratatouille [Ram+23a]	ICML 2023	Chapter 5	Diverse auxiliary tasks	Weight averaging
Rewarded soups [Ram+23b]	Under submission	Chapter 6	Diverse rewards	Weight averaging
UniVAL [Shu+23]	Under submission	✗	Diverse tasks and modalities	Weight averaging

As a final note, I also contributed to the organization of the PRINCE out-of-distribution generalization challenge: Eustache Diemert, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Alexandre Ramé, and Ugo Tanielian. "PRINCE: PRomoting INvariance for Classification of browsing journeys across Environments". In: *ECML PKDD* (2022). The results of the competition are available: https://codalab.lisn.upsaclay.fr/competitions/3353.

Chapter 2

THE UNREASONABLE EFFECTIVENESS OF ENSEMBLING FOR OUT-OF-DISTRIBUTION GENERALIZATION

Introduction

First, we shed light on the related literature in machine and deep learning: we recall the out-of-distribution (OOD) generalization challenge, the bias-variance decomposition of the error, and how ensembling reduces variance.

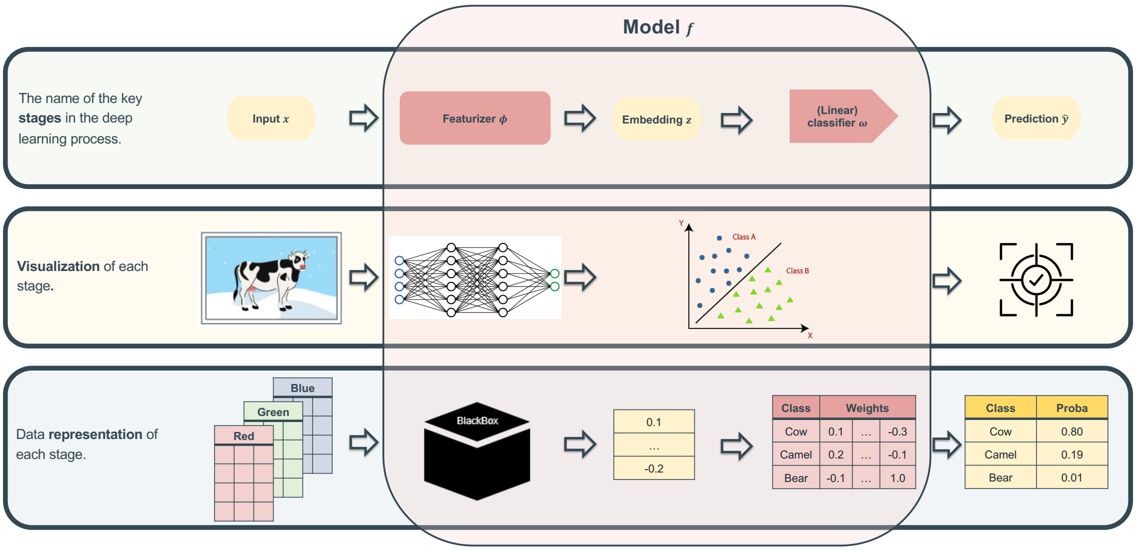
Then, we demonstrate that variance is intrinsically related to shifts in input distributions (*i.e.,* diversity shifts), while bias is inherently related to shifts in output conditional distributions (*i.e.,* correlation shifts). These theoretical contributions refine our understanding of OOD generalization, and in particular explain why the functional ensembling of diverse members excels under diversity shifts. These insights were first published in the theoretical section from: Alexandre Ramé, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. "Diverse Weight Averaging for Out-of-Distribution Generalization". In: *NeurIPS*. 2022.

Overall, we motivate our goal of *diverse and efficient ensembling*, and discuss how these two research directions have been previously addressed in the literature, the underlying challenges, trade-offs, and limitations of existing strategies.

2.1 Context: the fundamentals of deep learning

We first briefly explain how deep neural networks are trained in deep learning (DL) in a *supervised* setting, which is the key focus of this thesis.

Remark 2.1. *This thesis will primarily focus on classification among a predefined number of classes in computer vision (CV), a complex task due to the unstructured nature of images. However, thanks to the unification of architectures and training paradigms across modalities and tasks, all our findings could potentially be extended to various other setups. This will be done in our last Chapter 6, where we consider ensembling of models trained with reinforcement learning [Rus+16] in a wide range of setups: for example in text-to-text tasks, but also in multimodal tasks such as image-to-text captioning [Ren+17], visual question answering [Ben+17], and text-to-image generation with diffusion models [Rom+22].*


Model f
The name of the key **stages** in the deep learning process.
Input x
Featurizer ϕ
Embedding z
(Linear) classifier ω
Prediction $\hat{y}$
Visualization of each stage.
Y
Class A
Class B
X
Data **representation** of each stage.
Blue
Green
Red
BlackBox

0.1
…
-0.2

Class	Weights		
Cow	0.1	…	-0.3
Camel	0.2	…	-0.1
Bear	-0.1	…	1.0

Class	Proba
Cow	0.80
Camel	0.19
Bear	0.01

Figure 2.1. – **The fundamental concepts in deep learning.** The first row represents the names and notations used along this thesis, the second row illustrates the different concepts, while the third row depicts the corresponding mathematical objects. Specifically, an image x represented as a RGB matrix is given as input to a black-box featurizer Φ, that extracts an embedding vector z, representing in a structured way the different information in the image. This embedding is then fed to a (usually linear) classifier w, whose goal is to separate the different classes $\mathcal{Y}$; specifically, w predicts a probability distribution $\hat{y}$ that should be close to the true label y.

2.1.1 How to build a deep neural network

The key fundamental elements in a DNN are depicted in Figure 2.1. We analyze an input $x \in \mathcal{X}$, which in the context of CV typically refers to an image represented as a matrix of red green blue (RGB) pixels. The DNN f maps x to a prediction $\hat{y} = f(x, \theta)$ in the label space $\mathcal{Y}$ of size K. Specifically, $f(\cdot, \theta) : \mathcal{X} \to \mathcal{Y}$ is usually decomposed into a featurizer Φ parameterized by the weights ϕ, onto which we plug a dense linear classifier w parameterized by the weights ω; hence, $\theta = (\omega, \phi)$. The featurizer transforms a non-structured input into a *feature embedding* (denoted z) in a space where the K classes should ideally be linearly separated. These embeddings can then be processed arithmetically. For instance, the embeddings of two images containing the same object will usually exhibit a high cosine similarity (this would not hold if the similarity were computed in pixels). The classifier aims at detecting class boundaries in this embedding space, to allocate to each potential class a probability, all of which sum up to 1. The class with the highest predicted probability is chosen as the class prediction.

A significant portion of the DL literature is devoted to the design of the optimal *architecture* as a series of linear and non-linear transformations. The main difference between ML and DL is the number of layers in this featurizer. Some fundamental units of DNN architectures include:

- convolutional layers [Fuk80], the primary building block of convolutional neural network (CNN) architectures,
- self-attention layers [Vas+17], the primary building block of Transformer architectures [Vas+17; Dos+21],
- ReLU function [Aga18], a non-linear activation function applied element-wise which only retains positive values,
- normalization layers such as batch normalization (BN) [Iof+15],
- residual connections [He+16], adding direct connections between blocks to mitigate the *vanishing gradient* problem [Hoc+01].

Crucially, this thesis does *not* delve into these design choices; we aim to deliver architecture-agnostic insights applicable to any DNN, where understanding or knowledge of the various underlying architectural choices are not required. Consequently, we treat the deep featurizer as a non-linear black box.

2.1.2 How to train a deep neural network

Previous section has described the core components of a DNN $f(\cdot, \theta)$. Yet the key question remains: how to learn the appropriate weights θ?

2.1.2.1 Empirical risk minimization

Supervised DL is based on the statistical data-centric learning theory from [Vap92; Vap99]. Given pairs of (input x, label y), we want to learn a model that from x can predict $\hat{y}$ close to y. To measure the distance between a prediction $\hat{y} = f(x, \theta)$ and a class label y, we consider a loss function $\ell : \mathcal{Y}^2 \to \mathbb{R}_+$. In classification, the loss is usually the categorical cross-entropy:

$$\ell(\hat{y}, y) = -\sum_{c=1}^{K} y_c \log \hat{y}_c, \tag{2.1}$$

but we will also consider the mean-squared error (MSE) loss to simplify our proofs.

More specifically, we rely on a training (source) domain, denoted S with distribution p_S. In theory, we would seek θ minimizing the source error on the full training domain S: $\mathcal{E}_S(\theta) = \mathbb{E}_{(x,y)\sim p_S}[\ell(f(x, \theta), y)]$. In practice, we usually have only an empirical dataset $\mathcal{D}_S = \{(x_1, y_1), ..., (x_{n_S}, y_{n_S})\}$ of n_S *i.i.d.* samples from the underlying training distribution $p_S(X, Y)$. Then, we seek θ minimizing the source *empirical* error:

$$\hat{\mathcal{E}}_S(\theta) = \sum_{(x,y)\in\mathcal{D}_S} [\ell(f(x, \theta), y)]. \tag{2.2}$$

This strategy is called empirical risk minimization (ERM).

Algorithm 2.1 Procedure to optimize a neural network with ERM through SGD.

```
input: a model $f$ with trainable weights $\theta$
input: a dataset $\mathcal{D}_S$
input: a loss function $\ell$
input: a learning rate $\eta$ and a batch size $b$

while stopping criterion not satisfied do
    $(\boldsymbol{x}, \boldsymbol{y}) \leftarrow$ sample mini-batch of size $b$ from $\mathcal{D}_S$
    Forward pass: $\hat{\boldsymbol{y}} \leftarrow f(\boldsymbol{x}, \theta)$
    Compute loss: $\hat{\mathcal{E}}_S \leftarrow \ell(\hat{\boldsymbol{y}}, \boldsymbol{y})$
    Compute the gradients: $\delta \leftarrow \nabla_\theta \hat{\mathcal{E}}_S$
    Update all parameters: $\theta \leftarrow \theta - \eta\delta$
end while
```

2.1.2.2 Stochastic gradient descent

To optimize this ERM objective, the now standard strategy is stochastic gradient descent (SGD). As summarized in Algorithm 2.1, we iteratively update the weights θ in the direction opposite to the gradient of the loss w.r.t. the weights θ. Specifically, we sample a batch with b samples from $\mathcal{D}_S$, feed it into the model $f(\cdot, \theta)$, and compare the predictions $\hat{y}$ to the ground-truth labels y with the loss ℓ. Gradient computation is achieved by backpropagation [LeC+99] through the various layers of the architecture. The learning rate η controls the gradient step size.

Assuming proper convergence of SGD, and thanks to the overparameterization of the model and the universal approximation theorem [Cyb89], we expect this training loss to be relatively small; f will usually fit the training data almost perfectly.

2.1.2.3 Improved training procedures

While the architecture design and scale can significantly impact the model's performance, the training procedure is also crucial [Wig+21] and can go beyond SGD. For example, the most popular optimization algorithm is actually Adam [Kin+15]. Moreover, data augmentation is also essential, especially in CV where mixed sample data augmentation (MSDA) (such as Mixup [Zha+18a] or CutMix [Yun+19]) are widely used.

Notations. As stated earlier in the context of architectural design, the choice of training configuration will *not* be the primary focus of this thesis. Thus, we simply denote the training configuration as c, encompassing all training choices and sources of randomness in the learning (*e.g.*, initialization, hyperparameters, training stochasticity, data augmentation, epochs, *etc.*) excluding the fixed dataset $\mathcal{D}_S$. Then, we denote $l_S = \{\mathcal{D}_S, c\}$ a learning procedure on domain S. When necessary, we write $\theta(l_S)$ to refer to the weights obtained after optimization of $\hat{\mathcal{E}}_S(\theta)$ w.r.t. θ on $\mathcal{D}_S$ with configuration c.

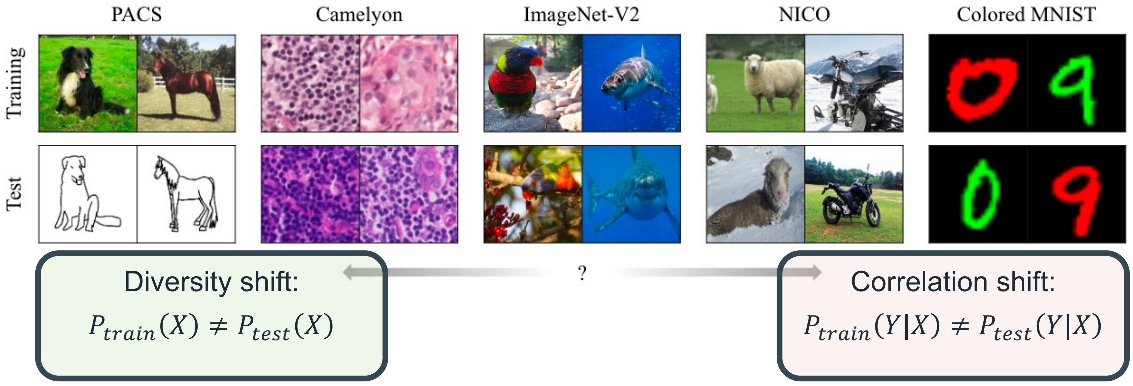


Figure 2.2. – **Visualization of the two types of distribution shifts**. Thanks to the bias-variance decomposition of the error Equation (BV), we will show that they have drastically different consequences on performances: while diversity shift increases the variance term, correlation shift increases the bias term. Image from [Ye+22].

2.1.3 Generalization

ID generalization. The generalization abilities of ML algorithms are evaluated on a test set, *i.e.*, a set of samples that were not seen during training. Thus, denoting T the (target) test domain with distribution p_T, we want θ to have small test target generalization error: $\mathcal{E}_T(\theta) = \mathbb{E}_{(x,y)\sim p_T}[\ell(f(x,\theta), y)]$. To this end, the learned model $f(\cdot,\theta)$ should ideally capture a robust mechanism, not just memorize all the training data [Zha+17]. The difference between the loss in train and in test is often referred to as the *generalization gap* [Kaw+17]. In the in-distribution (ID) setup, the train and test distributions are the same, *i.e.*, $p_S(X,Y) = p_T(X,Y)$.

OOD generalization. In this thesis, we focus on a more complex setup, named out-of-distribution (OOD) generalization, where $p_T(X,Y) \neq p_S(X,Y)$, *i.e.*, the test (target) distribution is different from the train (source) distribution. As highlighted in Section 1.3, OOD generalization under distribution shifts is critical to ensure applicability in real-world applications, where train and test hardly ever follow the same distributions. To better understand this key challenge, [Ye+22] decomposed distribution shifts into two types, visualized in Figure 2.2 and detailed below:

- *diversity shifts* (a.k.a. covariate shift) when $p_S(X) \neq p_T(X)$, *i.e.*, when the marginal input distributions differ. In this setup, the train and test distributions comprise data from related but distinct domains, for instance on PACS [Li+17] where we have pictures and drawings of the same objects.
- *correlation shifts* (a.k.a. concept shift) when $p_S(Y|X) \neq p_T(Y|X)$, *i.e.*, when the posterior covariate distributions differ. In this setup, the correlation between the input and the label depends on the domain: for instance on ColoredMNIST [Arj+19] where the color is spuriously correlated with the label.

Benchmarks. We primarily evaluate our approaches on standard image classification datasets. The first two papers [Ram+21a; Ram+21b] of this thesis used CIFAR-{10,100} [Kri+09]. Then, two years ago, the DomainBed [Gul+21] benchmark was introduced; it aimed at fairly evaluating the different OOD approaches and became the standard in the community. It includes five real-world multi-domain CV classification *datasets*, under which diversity shifts dominate.

- PACS [Li+17] includes domains {Art, Cartoon, Photo, Sketch}, with 9,991 examples and 7 classes.
- VLCS [Fan+13] includes photographic domains {Caltech101, LabelMe, SUN09, VOC2007}, with 10,729 examples and 5 classes.
- OfficeHome [Ven+17] includes domains {Art, Clipart, Product, Real}, with 15,588 examples and 65 classes.
- TerraIncognita [Bee+18] contains photographs of wild animals taken by camera traps at locations {L100, L38, L43, L46}, with 24,788 examples and 10 classes.
- DomainNet [Pen+19] has six domains {Clipart, Infograph, Painting, Quickdraw, Real, Sketch}, with 586,575 examples and 345 classes.

Moreover, DomainBed also includes the following synthetic dataset, under which correlation shifts dominate.

- Colored MNIST [Arj+19] is a variant of the MNIST handwritten digit classification dataset [LeC+10] with domain $\{90\%, 80\%, 10\%\}$: the correlation strengths between color and label vary across domains. It contains 70,000 examples and 2 classes.

Critically, all these datasets are multi-domain, *i.e.,* they contain several domains and each domain follows its own distribution. Each domain is successively considered as the test domain while others are for training and validation; this enables fair evaluation of generalization abilities to new domains. Moreover, DomainBed imposes a standard training setup and a strict evaluation protocol, detailed below.

- Each domain is split into 80% (used as training and testing) and 20% (used as validation for hyperparameter selection) splits.
- The network is a ResNet-50 [He+16] pre-trained on ImageNet, with frozen batch normalization layers and a dropout layer just before the classifier.
- The optimizer is Adam [Kin+15]. All runs are trained for 5k steps, except on DomainNet for 15k steps [Arp+21; Cha+21a].
- The hyperparameters follow either the mild or extreme distributions from Table 2.1.
- The experiments are repeated 3 times: the reported numbers will be the means and the standard errors.

The bitter lesson was that, when DomainBed was published, none of the existing methods performed significantly and consistently better than the standard ERM. Fortunately, the approaches proposed in this thesis will change this situation.

Table 2.1. – Hyperparameters, their default values and distributions for random search.

Hyperparameter	Default value	Random distribution Extreme	 Mild
Learning rate	$5 \cdot 10^{-5}$	$10^{\mathcal{U}(-5,-3.5)}$	$[1,3,5] \cdot 10^{-5}$
Batch size	32	$2^{\mathcal{U}(3,5.5)}$	32
ResNet dropout	0	$[0, 0.1, 0.5]$	$[0, 0.1, 0.5]$
Weight decay	0	$10^{\mathcal{U}(-6,-2)}$	$[10^{-6}, 10^{-4}]$

2.1.4 Bias-variance decomposition

To better understand and quantify the different sources of generalization error of our DNN, we follow the bias-variance decomposition of the expected error from [Koh+96].

Notations and assumptions. In the rest of this chapter, ℓ is the mean-squared error (MSE) for simplicity: yet, our results may be extended to other losses such as the cross-entropy following [Dom00; Woo+23]. We consider a model with weights $\theta(l_S)$ trained with the learning procedure l_S. We assume that there is no noise in the data; then the labeling function $f_S : \mathcal{X} \to \mathcal{Y}$ on S is defined on the source input support $\mathcal{X}_S = \{x \in \mathcal{X} / p_S(x) > 0\}$ by $\forall (x, y) \sim p_S, f_S(x) = y$. Similarly, $f_T : \mathcal{X} \to \mathcal{Y}$ the labeling functions on T is defined on the target input support $\mathcal{X}_T = \{x \in \mathcal{X} / p_T(x) > 0\}$ by $\forall (x, y) \sim p_T, f_T(x) = y$.

Bias and variance. [Koh+96] decomposed the expected error w.r.t. the learning procedure l_s into two terms:

$$\mathbb{E}_{l_S} \mathcal{E}_T \left(\theta\left(l_S\right)\right) = \mathbb{E}_{(x,y) \sim p_T} \left[\text{bias}^2\left(x, y\right) + \text{var}\left(x\right)\right], \tag{BV}$$

where, with $\bar{f}_S\left(x\right) = \mathbb{E}_{l_S}\left[f\left(x, \theta\left(l_S\right)\right)\right]$ the expected prediction:

- the bias $\text{bias}(x, y) = y - \bar{f}_S\left(x\right)$ measures how far off in general the predictions are from the ground-truth label,
- the variance $\text{var}(x) = \mathbb{E}_{l_S}\left[\left(f(x, \theta(l_S)) - \bar{f}_S\left(x\right)\right)^2\right]$ measures how much the predictions vary between different models.

In the traditional statistical learning theory, achieving good generalization usually requires finding a fine balance between these two terms. Indeed, it's generally assumed that as we add more parameters, the network becomes more flexible, thus the bias is reduced yet the variance increases. In details:

- small ML models with a limited number of parameters tend to have high bias but low variance. Such models are too simplistic to capture the intricate patterns in the data, resulting in a problem called *underfitting*.
- large DNNs with many parameters tend to have low bias but high variance. Such models are complex and flexible, thus capable of fitting their training data, but are

prone to a problem called *overfitting*, meaning they adapt too closely to the training data and would perform poorly on unseen data.

However, this traditional belief has been contested in the context of DL, this questioning being illustrated by the double descent phenomenon [Nak+19; DAs+20]. When increasing the model size, variance initially gets larger (consistently with traditional expectations); yet, at a certain point, when the architecture becomes sufficiently large, this trend reverses and variance gets smaller. In a similar spirit, the forthcoming Section 2.2.1 characterizes the variance in the limit case of infinitely large networks; we show that variance does not grow infinitely large, and depends mostly of the input shifts between train and test distributions.

2.2 A new bias-variance theoretical understanding of distribution shifts

We introduce the two main theoretical contributions of this thesis, which explain how the two kind of distribution shifts alter performance. First, in Section 2.2.1, we prove that variance dominates under diversity shift; this relies on the fact that variance essentially becomes a property of the train-test input distribution shifts for sufficiently large networks. Second, in Section 2.2.2, we show that bias dominates under correlation shift. These findings refine our understanding of the bias-variance trade-off and suggest different strategies to enhance generalization for each kind of shift.

2.2.1 Variance and diversity shift

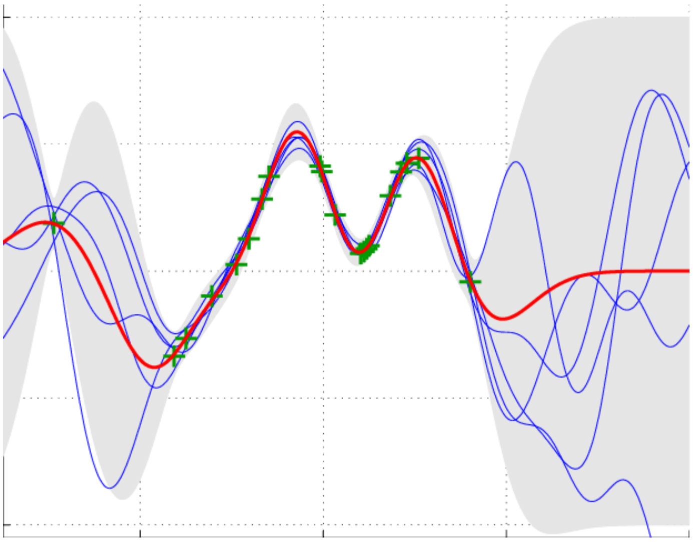


Figure 2.3. – **Mean and variance of the predictions for Gaussian processes**. Image from [Pér+13]. Intuitively, variance (grey area) across different predictions (blue lines) grows away from training samples (green crosses).

We relate variance to diversity shift [Ye+22] when the network gets infinitely large. We fix the source dataset $\mathcal{D}_S$ (with input support $X_{\mathcal{D}_S}$), the target dataset $\mathcal{D}_T$ (with input support $X_{\mathcal{D}_T}$) and the network's initialization. We get a closed-form expression for the variance of f over all other sources of randomness under Assumptions 2.1 and 2.2.

Assumption 2.1 (Infinite width)**.** *f is in the kernel regime [Dan17; Lee+17; Jac+18].*

This Assumption 2.1 follows the theoretical evidence [Dan17; Lee+17] that, when f is a sufficiently wide network, f behaves as a Gaussian process (GP). The corresponding kernel K is the neural tangent kernel (NTK) [Jac+18] characterized only by the initialization of the weights θ. This approximation is useful because GPs are more interpretable (see Figure 2.3) and more easily analyzed; in particular, their variances have a closed-form expression, as further discussed in Appendix C.1.1.1. To simplify this expression of variance, we make the following Assumption 2.2, further discussed and relaxed in Appendix C.1.1.2.

Assumption 2.2 (Constant norm and low intra-sample similarity on $\mathcal{D}_S$)**.** *$\exists(\lambda_S, \epsilon)$ with $0 \leq \epsilon \ll \lambda_S$ such that $\forall x_S \in X_{\mathcal{D}_S}, K(x_S, x_S) = \lambda_S$ and $\forall x'_S \neq x_S \in X_{\mathcal{D}_S}, |K(x_S, x'_S)| \leq \epsilon$.*

This Assumption 2.2 states that training samples have the same norm (following standard practice [Lee+17; Ah-10; Gho+21; Ren05]) and weakly interact [He+20; Sel+22]. We are now in a position to relate variance and diversity shift when $\epsilon \to 0$.

Proposition 2.1 (Variance and diversity shift. Proof in Appendix C.1.1)**.** *Given f trained on $\mathcal{D}_S$ (of size n_S) with NTK K, under Assumptions 2.1 and 2.2, the variance on $\mathcal{D}_T$ is:*

$$\mathbb{E}_{x_T \in X_{\mathcal{D}_T}}[\mathrm{var}(x_T)] = \frac{n_S}{2\lambda_S} MMD^2(X_{\mathcal{D}_S}, X_{\mathcal{D}_T}) + \lambda_T - \frac{n_S}{2\lambda_S}\beta_T + \mathcal{O}(\epsilon), \tag{2.3}$$

where MMD is the empirical maximum mean discrepancy in the RKHS of $K^2(x,y) = (K(x,y))^2$. Moreover, $\lambda_T = \mathbb{E}_{x_T \in X_{\mathcal{D}_T}} K(x_T, x_T)$ and respectively $\beta_T = \mathbb{E}_{(x_T, x'_T) \in X^2_{\mathcal{D}_T}, x_T \neq x'_T} K^2(x_T, x'_T)$ are the empirical mean similarities measured between identical (w.r.t. K) and respectively different (w.r.t. K^2) samples averaged over $X_{\mathcal{D}_T}$.

The statistical learning theory predicts that as the network becomes infinitely large, the variance also becomes infinitely large. Proposition 2.1 counters this traditional understanding by providing a new characterization of the variance when the network is sufficiently large to be in the kernel regime (as per Assumption 2.1). Then, the variance is actually mainly governed by the distance between the source and target inputs, as measured by the maximum mean discrepancy (MMD), rather than the model's complexity.

Critically for better understanding of OOD generalization, this MMD empirically estimates shifts in input marginals, *i.e.*, between $p_S(X)$ and $p_T(X)$. Then the first term in this expression of variance is actually very similar to the diversity shift formula in [Ye+22]: the MMD simply replaces the L_1 divergence, another equivalent distance used in [Ye+22]. The other terms, λ_T and β_T, both involve internal dependencies on the target dataset $\mathcal{D}_T$: they are constants w.r.t. $X_{\mathcal{D}_T}$. In conclusion, at fixed $\mathcal{D}_T$ and under our assumptions, Equation (2.3) shows that variance on $\mathcal{D}_T$ decreases when $X_{\mathcal{D}_S}$ and $X_{\mathcal{D}_T}$ are closer (for the MMD distance defined by the kernel K^2) and increases when they deviate. Intuitively, the further $X_{\mathcal{D}_S}$ is from $X_{\mathcal{D}_T}$, the less the model's predictions on $X_{\mathcal{D}_T}$ are constrained after fitting $\mathcal{D}_S$.

Remark 2.2. *This theoretical analysis explains an observation named underspecification [DAm+20]: models can behave differently in OOD despite similar test ID accuracy. This is simply because variance is larger in OOD than in ID.*

2.2.2 Bias and correlation shift (and support mismatch)

We now relate OOD bias to correlation shift [Ye+22] under Assumption 2.3.

Assumption 2.3 (Small ID bias). *$\exists \epsilon > 0$ small s.t. $\forall x \in \mathcal{X}_S, |f_S(x) - \bar{f}_S(x)| \leq \epsilon$.*

This Assumption 2.3 follows the traditional statistical theory; DNNs can fit their training distribution and thus have a small ID bias. This is realistic when they are sufficiently large, trained on a large training dataset representative of the source domain S with an appropriate training configuration c. Theses assumptions are relaxed in Appendix C.1.2.1.

Proposition 2.2 (Bias and correlation shift. Proof in Appendix C.1.2)**.** *With a bounded difference between the labeling functions $f_T - f_S$ on $\mathcal{X}_T \cap \mathcal{X}_S$, under Assumption 2.3, the bias on the test target domain T is:*

$$\begin{aligned}
&\mathbb{E}_{(x,y)\sim p_T}[\text{bias}^2(x,y)] = \textit{Correlation shift} + \textit{Support mismatch} + \mathcal{O}(\epsilon),\\
&\textit{where Correlation shift} = \int_{\mathcal{X}_T \cap \mathcal{X}_S} \left(f_T(x) - f_S(x)\right)^2 p_T(x)dx,\\
&\textit{and Support mismatch} = \int_{\mathcal{X}_T \setminus \mathcal{X}_S} \left(f_T(x) - \bar{f}_S(x)\right)^2 p_T(x)dx.
\end{aligned} \tag{2.4}$$

To understand Equation (2.4), we need to first observe that the labeling functions f_T and f_S verify $f_T(x) = \mathbb{E}_{p_T}[Y|X = x]$ and $f_S(x) = \mathbb{E}_{p_S}[Y|X = x]$, $\forall x \in \mathcal{X}_T \cap \mathcal{X}_S$. Then the first term actually measures correlation shifts in posterior distributions between source and target, as in [Ye+22]; this first term would increase in presence of spurious correlations that would modify the labeling functions. A basic example is the ColoredMNIST dataset, where color and label correlation is reversed at test time. Models that have learned the color-label correlation from the source data will predict labels based on the color of the digit rather than its shape, leading to wrong predictions in test and thus high bias.

In a more realistic scenario, instead of considering the input space, we can think in terms of feature space. Replacing x with a feature z, the bias would increase if the correlation between z and y is different in the source and target distributions. This is why it's important to learn a featurizer encoding inputs into a shared embedding space containing minimal domain-dependent information. This would reduce the bias introduced by potential spurious correlations between extracted features and labels when the model is used in OOD.

Remark 2.3. *Less critical in this thesis, the second term is caused by support mismatch between source and target. It was analyzed in [Rua+22] and shown irreducible in their "No free lunch for learning representations for DG".*

Conclusion. This section proves that the two types of distribution shifts reduce performance in two different ways. First, diversity shift increases the variance term; we show in the following Section 2.3.2 that ensembling is helpful in this setup because ensembling is actually a variance reduction strategy. Second, correlation shift increases the bias term.

Remark 2.4. *The most successful methods to tackle bias and thus correlation shift usually involve data balancing [Idr+22], robust optimization [Sag+20a], or domain invariance [Arj+19]. Specifically, in Fishr [Ram+22a], we introduce a novel regularization enforcing domain* invariance in the space of the gradients *of the loss. Fishr was proven effective on ColoredMNIST and on DomainBed, and to this day [Yu+23], remains one of the best strategies to tackle spurious correlations. We have relegated Fishr to Appendix E as its contributions are orthogonal to the main focus of this thesis:* ensembling.

2.3 Diverse and efficient ensembling

2.3.1 Prediction averaging

In this thesis, we study how to best combine M models. The standard strategy is the functional ensembling (ENS) which averages the networks' predictions, as illustrated in Figure 2.4. This ensembling strategy has been a popular research topic in machine learning [Nil65; Wol92; Bre96; Die00; Rok10], leading to successful strategies such as random forests [Ho95] or XGBoost [Che+16]. This trivially extends to the functional ensembling of DNNs [Han+90; Kro+95; Zho+02]. We thus define the predictive function:

$$f_{\mathrm{ENS}}(\cdot, \{\theta_i\}_{i=1}^{M}) = \frac{1}{M}\sum_{i=1}^{M} f(\cdot, \theta_i). \tag{2.5}$$

Note that most approaches average the logits rather than the probabilities, as it (slightly but consistently) works better empirically [Ju+18]. In particular, the recent DE [Lak+17] highlighted the success of this simple strategy with independently trained DNNs for better uncertainty estimation [Ash+20].

2.3.2 Bias-variance-covariance decomposition

For simplicity, we consider that the M models have the same architecture and that the weights $\{\theta_i\}_{i=1}^{M} = \{\theta(l_S^{(i)})\}_{i=1}^{M}$ are obtained from M identically distributed (*i.d.*) learning procedures. Then, Equation (BVC) extends Equation (BV) to ensembling: to take into account the M models, the expectation is over the joint distribution $L_S^M = \{l_S^{(i)}\}_{i=1}^{M}$, thus

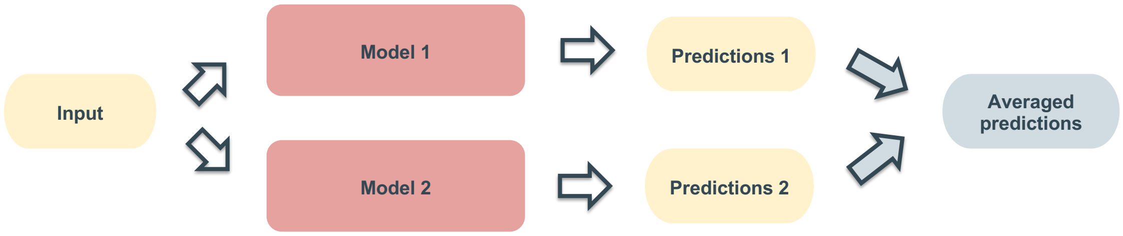


Figure 2.4. – **The traditional functional ensembling**, averaging the predictions for $M = 2$ models.

we denote $f_{\text{ENS}}\left(\cdot, L_S^M\right)$. Then, the expected generalization test error is decomposed into three terms: bias, variance and covariance.

Proposition 2.3 (Bias-variance-covariance [Ued+96; Bro+05a]. Proof in Appendix C.1.3.)**.** *Denoting $\bar{f}_S(x) = \mathbb{E}_{l_S}\left[f\left(x, \theta\left(l_S\right)\right)\right]$, under i.d. learning procedures L_S^M, the expected generalization error on domain T of the ensembling of those M models over L_S^M is:*

$$\mathbb{E}_{L_S^M} \mathcal{E}_T\left(f_{ENS}\left(\cdot, L_S^M\right)\right) = \mathbb{E}_{(x,y)\sim p_T}\left[\text{bias}^2(x,y) + \frac{1}{M}\text{var}(x) + \frac{M-1}{M}\text{cov}(x)\right], \tag{BVC}$$

where $\text{cov}(x) = \mathbb{E}_{l_S, l'_S}\left[\left(f(x, \theta(l_S)) - \bar{f}_S(x)\right)\left(f(x, \theta(l'_S)) - \bar{f}_S(x)\right)\right]$ *is the prediction covariance between two models.*

In comparison to Equation (BV), the following observations can be made:

- The bias is the same for the ensemble or for each of its *i.d.* members: combining different models does *not* reduce the bias. Consequently, this suggests that ensembling provides no benefits under correlation shift (where bias is the major issue). According to this theory, ensembling would not work on ColoredMNIST, as latter confirmed in Section 4.6.3.4.
- The variance of the ensembling is divided into two terms: (i) the variance of each of its *i.d.* members divided by M, and (ii) a covariance term measuring the correlations across members' predictions. The covariance will be further analyzed in the subsequent Section 2.3.3, but for now, let's assume that it is small. Then, this theory suggests that ensembling reduces the variance by a factor of M, effectively removing variance for sufficiently large M. Moreover, we have previously noted variance is large under diversity shift; this suggests that ensembling is highly beneficial under diversity shift, for example on PACS. This is confirmed in the experiments along this thesis.

In conclusion, this bias-variance-covariance analysis suggests that:

(i) ensembling is useless for correlation shift
(ii) but efficient for diversity shift,
(iii) as long as the covariance is small.

2.3.3 Diverse ensembling

2.3.3.1 Diversity

> Diversity: the art of thinking independently together
>
> *Malcolm Forbes*

Diversity to reduce covariance. In addition to the bias and variance of the members, the generalization error of an ensemble also depends on the covariance of the predictions $\{f(\cdot, \theta_i)\}_{i=1}^{M}$. In the extreme case where all predictions are identical, the covariance equals the variance and ensembling is no longer beneficial. This is because all models are equivalent and the ensemble is no perform better than a single model. Conversely, if the predictions are totally uncorrelated, then the variance is divided by M, making the ensemble more accurate than its individual members. This highlights that the members should have uncorrelated predictions. This lack of correlation across members' predictions can be summarized into the key notion of *diversity*, as defined in [Dieoo]:

Two classifiers are diverse if they make different errors.

Measure of diversity. The concept of diversity has been widely explored in the ensembling literature [Bro+05b; Woo+23]. However, as pointed out in "Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy" [Kun+03], "measuring diversity is not straightforward because there is no generally accepted formal definition". Thus, numerous diversity measures have been introduced [Kun+03; Aks03; Kor+19]. Considering two classifiers, the most standard ones consider N^{ij}, the number of times the first classifier is (correct if $i = 1$ or wrong if $i = 0$) and the second classifier is (correct if $j = 1$ or wrong if $j = 0$). For example, N^{10} is the number of times that the first classifier is correct but not the second. Then diversity can be measured with:

1. the ratio-error [Aks03], defined as the ratio $\frac{N^{01}+N^{10}}{N^{00}}$ between the number of asynchronous errors and of simultaneous errors.
2. the Q-diversity [Yul00], defined as $2\frac{N^{01}N^{10}}{N^{11}N^{00}+N^{01}N^{10}}$ (or equivalently 1 minus the Q-statistic).

For ensembles with more than two members ($M > 2$), these pairwise similarity measures are averaged over all possible pairs: higher values signify that members are less likely to make errors on the same inputs. In our published papers, we have also considered other prediction diversity measures such as the agreement score (the frequency that both classifiers predict the same class), the Kohavi-Wolpert variance [Koh+96] (the variability of the predicted class), and the entropy diversity (measuring overall disagreement). In this thesis, we also use a diversity measure in features: the centered kernel alignment complement (CKAC) [Kor+19], measuring to what extent the pairwise similarity matrices (computed on domain T) are aligned—where similarity is the dot product between feature embeddings extracted from two different networks (rather than the predictions).

2.3.3.2 Related work for diverse ensembling

The most standard ensembling baseline is deep ensembling (DE) [Lak+17], where *the weights are trained independently* from different seed initializations. The diversity among members primarily relies on the randomness of the initialization [Kol+91] and of the learning procedure, hoping to converge towards different explanations for the training data [For+19a; Wil+20]. This simple strategy already provides diverse models, yet, more sophisticated approaches tried to further increase diversity, either explicitly (with diversity regularizations) or implicitly (with additional randomness).

Explicit diversity. The first research direction is summarized by this quote from [Bro+05a]: "why shouldn't we try to find some way to capture the effect of the covariance in the error function?". Then, many works tried to explicitly reduce covariance by encouraging diversity among the members. Most approaches explicitly regularize the predictions: [Shu+18] force the members to have negatively correlated errors [Liu+99a; Liu+99b]; [Mas20] theoretically motivated the minimization of a second-order PAC-Bayes bounds. Others [Kar+19; Dab+20] enforced diversity in gradients; however, as stated in [Dab+20], "promoting diversity of gradient directions slightly degrades the classification performance on natural examples". Overall, as far as we know, the unique approach more accurate than the DE baseline is ADP [Pan+19], decorrelating only the non-maximal predictions across members.

Remark 2.5. *In contrast with previous diversity methods, some works suggested co-distillation across members. For example, in [Zha+18b] the members learn to mimic each other by reducing the KL between pairs of predictions. This was further extended in [Son+18; Guo+20; Chu+20; Wu+20a; Che+20] and in [Lan+18], the latter using a weighted combination of logits as teacher hence providing better distillation. The problem is that co-distillation reduces diversity by homogenization; in our experiments, all these approaches underperform the DE [Lak+17] baseline.*

Implicit diversity. The second research direction is to implicitly increase diversity; they are usually more practical as they do not change the training objective. Some approaches introduced additional stochasticity into the training by providing *subsets of data* to learners with bagging [Bre96], bootstrapping [Efr+94] or by backpropagating *subsets of gradients* [Lee+16]. Yet, these strategies artificially reduce the number of training samples, hurting performance for DNN that can overfit their training dataset [Nak+19], and thus failed [Nix+20] because of reduced individual accuracies. One could think of boosting [Fri01] strategies, yet [Lak+17] argued that sequential training is not suitable for DNNs (mostly because of lack of parallelization). Some more successful approaches applied different *transformations* [Dvo+19; Sti+20a], used *different data augmentations* [Wen+21] or *hyperparameters* [Sin+16; Rui+20; Yan+20d; Wen+20]: these simple diversity tricks will be fundamental in this thesis.

2.3.4 Efficient ensembling

Functional ensembling cost. Ensembling is empirically successful; yet, the inherent issue with ensembling is its cost, both in terms of time and memory, both during training and during inference. These overheads increase linearly with the number of members: handling M models requires M times more resources than handling a single model.

Related work for efficient ensembling. Despite this, naive functional ensembling methods can still be competitive. For example, [Chi+20; Lob+20; Wan+20a; Zha+20b] all found a memory split advantage (MSA): an ensemble of multiple smaller networks typically outperforms a single larger network when compared at an equal number of weights. However, this overhead still limits applicability in real-world applications, where scalability is a crucial factor, and overall suggests a promising research direction:

How could we imitate the performance of ensembling within a single model?

Previous works tried to solve this challenge by proposing different efficient methods.

Weight sharing. Seminal methods simply share part of the architecture and weights across models. These methods include the *branch-based* TreeNets architecture [Lee+15; Son+18; Lan+18] where different classification branches are deployed on top of shared low-level features. Monte Carlo dropout [Gal+16] and matrix factorization [Wen+19; Dus+20] methods also reduce the memory costs, yet they still require multiple forward passes at inference, and actually usually perform poorly [Ash+20].

Subnetwork. [Gao+19; Hav+21; Sof+20; Dur+20; Yan+20c] offer another efficient ensembling strategy, by fitting diverse subnetworks inside one large base network. The idea is that over-parameterized DNNs [Fra+19; Mol+17] can be pruned without loss in performance, and thus their use of parameterization could (in theory) be better optimized. Then the question is how to prevent homogenization among those subnetworks. Multiple strategies were proposed: [Gao+19] includes stochastic channel recombination; [Dur+20] relies on predefined binary masks; in GradAug [Yan+20c], subnetworks only leverage the first channels up to a given percentage. The multi-input multi-output (MIMO) method [Hav+21] is notable, as it does not need structural differences among subnetworks. MIMO's main idea in to teach each subnetwork to classify only one of the multiple inputs simultaneously provided; the subnetworks then learn to build their own paths in the base architecture, without homogenization. Such a strategy [Sof+20] can be motivated with arguments from information theory.

Remark 2.6. *This simple yet effective MIMO technique motivated our* MixMo *approach [Ram+21b], where we study how to best mix the multiple inputs given simultaneously during training; this leads to replacing the suboptimal summing operation in MIMO with an improved mixing mechanism based on patches and CutMix [Yun+19]. The key limitation is that MIMO* *and MixMo require training from scratch, and thus cannot transfer knowledge from foundation models. Thus, for for sake of brevity, the details of MixMo are relegated in Appendix D.*

Training sharing. Methods such as snapshot ensembles [Hua+17] and MotherNets [Was+20] share part of the training process to reduce the training overhead. Specifically, snapshot ensembles [Hua+17] create an ensemble of diverse snapshots visited along a single training run with a cyclical learning rate. These methods effectively reduce computational costs during training, yet they do not address the inference overhead.

Weight averaging. An effective strategy to reduce the inference overhead is to average the weights of multiple models, rather than the predictions. Despite the non-linearities in the architecture, and thus perhaps surprisingly, this weight averaging (WA) enables combining into one single model the abilities from multiple models. This idea was initially used in moving average strategies [Izm+18; Zha+19a] as a cheap alternative to snapshot ensembles [Hua+17]. Then, this WA showed promising results in OOD [Cha+21a; Arp+21]. [Cha+21a] argued that WA succeeds in OOD because it provides flatter solution in the loss landscape. Though indeed WA reduces the maximum eigenvalue of the Hessian, in Chapter 4 we will challenge this flatness-based explanation, and show that WA actually improves generalization because of its similarity with functional ensembling. Another issue of existing WA strategies is that the averaged models were highly similar, as they were collected along a single training trajectory, thus limiting diversity and performance. These discovered limitations form the basis of our works on WA, and will be further detailed along this thesis.

2.4 Conclusion

The first goal of this chapter is to provide theoretical guarantees for the empirical success of functional ensembling. Based on the bias-variance-covariance decomposition of the error, we recall that averaging the predictions from multiple models reduces variance. Moreover, we prove that, for sufficiently large DNNs, the variance is actually primarily caused by diversity shifts in input marginal distributions. Therefore, this explains why *ensembling especially helps under diversity shift*.

The second goal of this chapter is to explain that diversity in ensembling is fundamental, yet complex to optimize directly. Following the limited success of existing diversity regularization in predictions, we propose in the next Chapter 3 a novel explicit regularization enforcing diversity at the *feature embedding* level.

The third goal of this chapter is to highlight that traditional functional ensembling lacks scalability due to its high computational cost. The existing methods for efficient ensembling correlate errors among members, thus reducing diversity. In Chapters 4 to 6, we push the frontier of weight averaging methods and propose strategies to find *accurate and diverse models that remain averageable in weights*.

Chapter 3

DICE: Diversity in Ensembles via Conditional Redundancy Adversarial Estimation

3.1 Introduction

Averaging the predictions of several models can significantly improve the generalization ability of a predictive system. Indeed, ensembling reduces the variance (see Proposition 2.3) thanks to the diversity among their members. In deep ensembling (DE) [Lak+17], *the models are traditionally trained independently*. The objective of this chapter is to further increase the diversity across members to improve the ensemble performance.

As previously detailed in Section 2.3.3.2, several works [Mas20; Pan+19] explicitly regularize the predictions to increase diversity, yet with limited success; they systematically reduced the individual performances of the members, highlighting the *trade-off between ensemble diversity and individual accuracies*. Our idea is to encourage all members *to predict the same thing, but for different reasons*. Therefore we explicitly enforce diversity in the *feature* space rather than directly within predictions. Intuitively, to maximize the impact of a new member, extracted features should bring information about the label that is absent at this time so unpredictable from features extracted by other members. The goal is to remove *irrelevant redundant information across members*, *e.g.,* information shared among features extracted by different members, but useless for label prediction. This redundancy may be caused by a detail in the image background (making members predict badly simultaneously); the key point is that this detail can not be found in features extracted from other images, even when they belong to the same class, as shown in Figure 3.1.

Our new learning framework is called DICE and is driven by information theory and information bottleneck (IB) [Tis01; Ale+17] principles. Specifically, we follow the minimum necessary information (MNI) principle [Fis20] which states that features should be *minimal* (*i.e.,* compressed) while keeping the necessary (*i.e.,* relevant) information about the label. The goal is to prevent overfitting on noise (memorization) while enabling the learning of a robust predictive mechanism. Specifically, DICE applies the MNI criterion [Fis20] to the ensemble of M members, and thus seeks to reduce (i) the mutual information (MI) between features and inputs, but also (ii) the information shared between members' features conditioned upon the label. This second point prevents extracted features from being redundant. Intuitively, this comes back to increasing the MI distance between pairs of members, and thus benefitting from MI ability to detect arbitrary dependencies between random variables (such as symmetry, see Figure 3.1). Lastly, a key

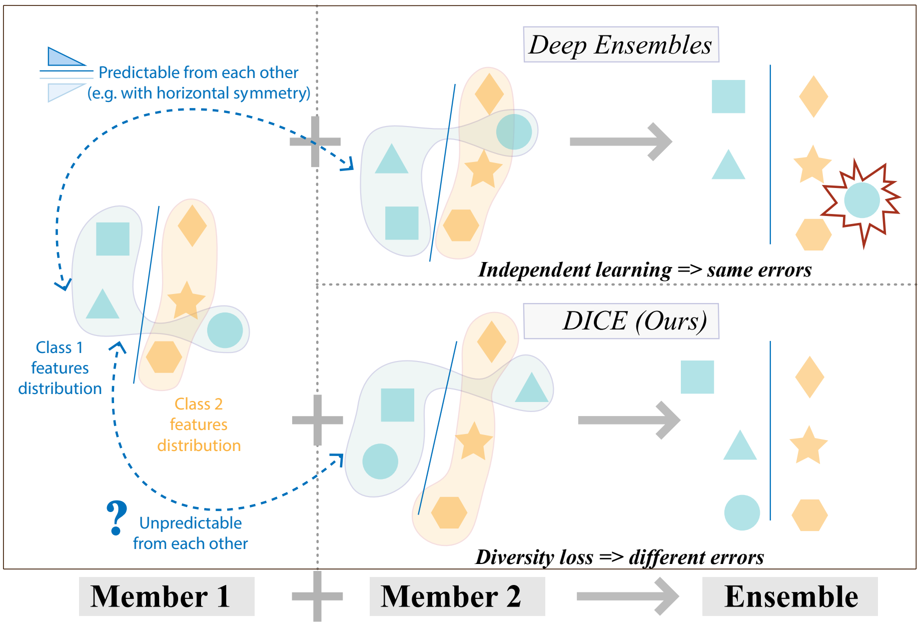


Figure 3.1. – **Motivation**. Our DICE regularization prevents features extracted by different members from being predictable from each other *conditionally* on the class. Intuitively, features extracted by members (1, 2) from one input (●,●) should not share more information than features from two inputs in the same class (●,▲): we show this is equivalent to preventing the features extracted by the first member for a first input (●,-) to be informative enough to differentiate between (i) the features extracted by the second member on the same input (-,●) and (ii) the features extracted by the second member for another input from the same class (-,▲).

contribution lies in the conditioning upon the label target, which we argue protects the relevancy and informativeness of extracted features.

The uniqueness of *mutual information* as a distance measure between variables has been applied in countless machine learning projects [Kim+19; Kem+20; Hje+19]. Yet, MI remains challenging to estimate between high-dimensional features. In this work, we build upon recent advances in neural estimation of MI [Bel+18], based on the Donsker-Varadhan representation of the KL formulation of MI. Overall, to approximate our regularization, we end up implementing an adversarial framework [Goo+14], where a discriminator prevents features from being conditionally predictable from each other. In summary, DICE increases diversity by *adversarially reducing irrelevant redundant information among features*.

- We introduce DICE, a novel objective to train ensembles of neural networks; based on arguments from information theory, DICE seeks to minimize the conditional redundancy between features (Section 3.2.1).

- We propose an implementation of DICE with a neural estimation of the conditional redundancy, leveraging an adversarial learning framework (Section 3.2.2).

This chapter has led to the publication of the following paper: Alexandre Ramé and Matthieu Cord. "DICE: Diversity in Deep Ensembles via Conditional Redundancy Adversarial Estimation". In: *ICLR*. 2021.

3.2 DICE

> Nobody knows what entropy really is
>
> *John Van Neumann* to *Claude Shannon*

Notations. Given an input random variable X, a network parameterized by weights θ is trained to extract the best possible feature embedding Z to model the distribution $p_\theta(Y|X)$ over the targets, which should be close to the Dirac on the true label. Our approach is designed for ensembles with M members $\{\theta_i\}_{i=1}^{M}$ extracting features $\{Z_i\}_{i=1}^{M}$. We average the M (logit) predictions during inference. In this section, for clarity and simplicity, we consider $M = 2$; yet the published paper [Ram+21a] trivially extends DICE to $M > 2$ members.

Quick overview of DICE. In Section 3.2.1, we justify the DICE objective with information theory. Then in Section 3.2.2, we will detail our implementation of this DICE objective. In brief, our training strategy will (i) train each member separately for classification with information bottleneck (IB) while (ii) removing irrelevant shared information by adversarial training with a discriminator. In conclusion, members should learn to classify with conditionally uncorrelated features and thus increased diversity.

3.2.1 DICE training objective

3.2.1.1 Baseline: non-conditional objective

The MNI criterion from [Fis20] states that the learned features Z should ideally capture only *minimal* compressed information from X, while preserving the *necessary* relevant information about the label Y. We apply this principle to the ensembling of $M = 2$ members by considering the two Markov chains $Z_i \leftarrow X \leftrightarrow Y$ for $i \in \{1, 2\}$. Using entropy as the measure of information, to satisfy the necessary constraint, $I(Y; Z_i)$ should be maximized. Then, regarding the minimality constraint, a first *non-conditional* formulation is to consider that $I(X; Z_i)$ should be minimized. Mutual information being non-negative, we transform these constraints into a first objective, recovering the IB objective from [Ale+17], but simply applied independently to the M ensembling's members.

$$\begin{aligned}\mathrm{IB}_{\beta_{ib}}(Z_1, Z_2) = \frac{1}{\beta_{ib}} \overbrace{[I(X; Z_1) + I(X; Z_2)]}^{\text{Compression}} - \overbrace{[I(Y; Z_1) + I(Y; Z_2)]}^{\text{Relevancy}} \\ = \mathrm{IB}_{\beta_{ib}}(Z_1) + \mathrm{IB}_{\beta_{ib}}(Z_2).\end{aligned} \tag{3.1}$$

The specificities of ensembling appear when we consider cross terms $I(Z_1;Z_2)$, that should be minimal according to the minimality constraint of the MNI.

$$\begin{aligned}\mathrm{IBR}_{\beta_{ib},\delta_r}(Z_1,Z_2) &= \frac{1}{\beta_{ib}}\overbrace{[I(X;Z_1)+I(X;Z_2)]}^{\text{Compression}} - \overbrace{[I(Y;Z_1)+I(Y;Z_2)]}^{\text{Relevancy}} + \delta_r \overbrace{I(Z_1;Z_2)}^{\text{Redundancy}} \\ &= \mathrm{IB}_{\beta_{ib}}(Z_1) + \mathrm{IB}_{\beta_{ib}}(Z_2) + \delta_r I(Z_1;Z_2).\end{aligned} \tag{3.2}$$

Analysis. In this baseline non-conditional criterion, relevancy encourages Z_1 and Z_2 to capture information about Y. Compression & redundancy (R) split the information from X into two compressed & independent views. The relevancy-compression-redundancy trade-off depends on the values of β_{ib} & δ_r.

3.2.1.2 DICE: conditional objective

The problem is that the previous non-conditional compression and redundancy terms in IBR also reduce necessary information related to Y: indeed, it is detrimental to *fully* disentangle Z_1 and Z_2 while training them to predict the same Y. This is shown on Figure 3.2 where redundancy regions (blue horizontal stripes ▤) overlap with relevancy regions (red diagonal stripes ▨ ▨). Indeed, as argued in [Fis+20], the minimality constraints need to be conditioned on the label; thus, we derive the following *conditional* constraints given Y:

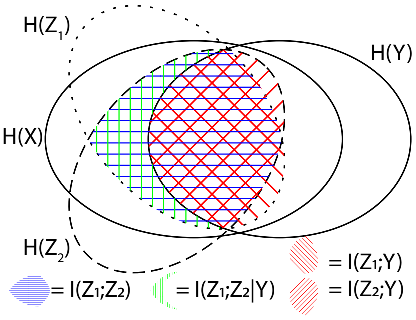


Figure 3.2. – **Venn information diagram**. DICE minimizes conditional redundancy (green vertical stripes) with no overlap with relevancy (red diagonal stripes).

$$I(X;Z_1|Y) = I(X;Z_2|Y) = I(Z_1;Z_2|Y) = 0.$$

We transform these constraints into our main DICE objective:

$$\begin{aligned}&\mathrm{DICE}_{\beta_{ceb},\delta_{cr}}(Z_1,Z_2) \\ &= \frac{1}{\beta_{ceb}}\underbrace{[I(X;Z_1|Y)+I(X;Z_2|Y)]}_{\text{Conditional Compression}} - \underbrace{[I(Y;Z_1)+I(Y;Z_2)]}_{\text{Relevancy}} + \delta_{cr} \underbrace{I(Z_1;Z_2|Y)}_{\text{Conditional Redundancy}} \\ &= \mathrm{CEB}_{\beta_{ceb}}(Z_1) + \mathrm{CEB}_{\beta_{ceb}}(Z_2) + \delta_{cr} I(Z_1;Z_2|Y),\end{aligned} \tag{3.3}$$

with $\delta_{cr} > 0$ and $\beta_{ceb} > 0$, where we recover two conditional entropy bottleneck (CEB) [Fis20] components, $\mathrm{CEB}_{\beta_{ceb}}(Z_i) = \frac{1}{\beta_{ceb}} I(X;Z_i|Y) - I(Y;Z_i)$.

Analysis. Like previously in Equation (3.2), the relevancy terms force features to be informative about the task Y. The difference lies in the conditioning in the bottleneck constraints, only minimizing information irrelevant to Y. More specifically, the conditional compression first removes in Z_i information from X not relevant to Y. Then, the conditional redundancy (CR) forces the members to have independent features *conditionally upon the class*. It encourages diversity without affecting members' individual precision as it protects in Z_i information related to the target label Y. The key intuition explaining why this CR term is important is:

> Information from X irrelevant to predict Y should certainly not be in Z_1 *or* Z_2, but it is even worse if they are in Z_1 *and* Z_2 simultaneously as it would cause shared errors.

One could argue that reducing $I(X, Z_i|Y)$ indirectly controls $I(Z_1, Z_2|Y)$, as by chain rule we have $I(Z_1; Z_2|Y) \leq I(X; Z_i|Y)$. Yet we will empirically observe that it is more efficient to directly target this intersection region through the CR term. In a final word, DICE is to IBR in ensembling as CEB [Fis+20] is to IB [Ale+17] for a single network.

3.2.2 DICE implementation

We now provide an empirical strategy to approximate the two CEB terms and the CR term in the DICE objective from Equation (3.3).

3.2.2.1 Variational approximation of conditional entropy bottleneck

To approximate the independent CEB terms for $i \in \{1, 2\}$, we exactly follow [Fis20], as further detailed in Appendix C.2.2. Specifically, we consider Markov assumptions in $Z_i \leftarrow X \leftrightarrow Y$ and the training dataset $\mathcal{D}_S = \{x_n, y_n\}_{n=1}^{n_S}$ of n_S *i.i.d.* points. Then [Fis20] showed that $CEB_{\beta_{ceb}}(Z_i)$ is variationally upper bounded by:

$$\text{VCEB}_{\beta_{ceb}}(\{e_i, b_i, d_i\}) = \frac{1}{n_S} \sum_{n=1}^{n_S} \frac{1}{\beta_{ceb}} D_{\text{KL}}\left(e_i\left(\cdot|x_n\right) \| b_i\left(\cdot|y_n\right)\right) - \mathbb{E}_{z \sim e_i(\cdot|x_n)}\left[\log d_i\left(y_n|z\right)\right]. \tag{3.4}$$

This loss is applied separately on each member $\theta_i = \{e_i, d_i, b_i\}$ made of an encoder, a classifier and a backward encoder. Specifically, the feature z follows the distribution $e_i(\cdot|x)$ generated by the encoder, $d_i(y|z)$ is a variational approximation of the true label distribution $p(y|z)$ by the classifier, and $b_i(\cdot|y)$ is a variational approximation of the feature distribution for z ($p(\cdot|y)$) by the backward encoder (conditioned only on the label).

Practically, we parameterize all distributions with Gaussians. The encoder e_i is a traditional DNN featurizer (*e.g.*, ResNet-32) that learns *distributions* (means and covariances) rather than deterministic points in the feature space. That's why e_i transforms an image into 2 tensors; a features-mean $e_i^\mu(x)$ and a diagonal features-covariance $e_i^\sigma(x)$ each of size d (*e.g.*, 64). The classifier d_i is a dense layer that transforms a features-sample z (following $e_i\left(\cdot|x_n\right)$) into logits to be aligned with the target y through conditional cross-entropy. z is obtained via reparameterization trick: $z = e_i(x, \epsilon) = e_i^\mu(x) + \epsilon e_i^\sigma(x)$ with $\epsilon \sim N(0, 1)$. Finally, the backward encoder b_i is implemented as an embedding layer of size (K, d) mapping the K classes to class-features-means $b_i^\mu(z|y)$ of size d, as we set the class-features-covariance to $\mathbb{1}$. The Gaussian parametrization also enables the exact computation of the D_{KL} (see Appendix C.2.1), that forces (i) features-mean $e_i^\mu(x)$ to converge to the class-features-mean $b_i^\mu(z|y)$ and (ii) the predicted features-covariance $e_i^\sigma(x)$ to be close to $\mathbb{1}$. The *advantage of VCEB versus VIB* [Ale+17] (detailed in Appendix C.2.2) is the class conditional $b_i^{\mu}(z|y)$ versus non-conditional $b_i^{\mu}(z)$, protecting class information.

3.2.2.2 Adversarial estimation of conditional redundancy

Theoretical problem. We now focus on estimating $I(Z_1; Z_2|Y)$, with no such Markov properties. Despite being a pivotal measure, mutual information (MI) estimation historically relied on nearest neighbors [Sin+03; Kra+04; Gao+18] or density kernels [Kan+15] that scale badly in high dimensions. We benefit from recent advances in *neural estimation of MI* [Bel+18], built on optimizing [Don+75] dual representations of the KL divergence. [Muk+20] extended this formulation for conditional MI estimation.

$$\begin{aligned}
\mathrm{CR} = I(Z_1; Z_2|Y) &= D_{\mathrm{KL}}(P(Z_1, Z_2, Y)\|P(Z_1, Y)p(Z_2|Y)) \\
&= \sup_f \mathbb{E}_{x\sim p(z_1,z_2,y)}[f(x)] - \log\left(\mathbb{E}_{x\sim p(z_1,y)p(z_2|y)}[\exp(f(x))]\right) \\
&= \mathbb{E}_{x\sim p(z_1,z_2,y)}[f^*(x)] - \log\left(\mathbb{E}_{x\sim p(z_1,y)p(z_2|y)}[\exp(f^*(x))]\right),
\end{aligned}$$

where f^* computes the pointwise likelihood ratio, *i.e.*, $f^*(z_1, z_2, y) = \frac{p(z_1,z_2,y)}{p(z_1,y)p(z_2|y)}$.

Empirical neural estimation. We estimate CR (i) using the empirical data distribution and (ii) replacing $f^* = \frac{w^*}{1-w^*}$ by the output of a discriminator w, trained to imitate the optimal w^*. Let $\mathcal{B}_J$ be a batch sampled from the observed joint distribution $p(z_1, z_2, y) = p(e_1(z|x), e_2(z|x), y)$; we select the features extracted by the two members from one input. Let $\mathcal{B}_P$ be sampled from the product distribution $p(z_1, y)p(z_2|y) = p(e_1(z|x), y)p(z_2|y)$; we select the features extracted by the two members from two different inputs that share the same class. We train a multi-layer network w on the binary task of distinguishing these two distributions with the standard cross-entropy loss:

$$\mathcal{L}_{ce}(w) = -\frac{1}{|\mathcal{B}_J| + |\mathcal{B}_P|}\left[\sum_{(z_1,z_2,y)\in\mathcal{B}_J} \log w(z_1, z_2, y) + \sum_{(z_1,z_2',y)\in\mathcal{B}_P} \log(1 - w(z_1, z_2', y))\right]. \tag{3.5}$$

If w is calibrated, a consistent [Muk+20] estimate of CR, with $f = \frac{w}{1-w}$, is:

$$\hat{\mathcal{I}}_{DV}^{CR} = \frac{1}{|\mathcal{B}_J|}\sum_{(z_1,z_2,y)\in\mathcal{B}_J} \underbrace{\log f(z_1, z_2, y)}_{\text{Diversity}} - \log\left(\frac{1}{|\mathcal{B}_P|}\sum_{(z_1,z_2',y)\in\mathcal{B}_P} \underbrace{f(z_1, z_2', y)}_{\text{Fake correlations}}\right). \tag{3.6}$$

Intuition. By training our members to minimize $\hat{\mathcal{I}}_{DV}^{CR}$, we force triples from the joint distribution to be indistinguishable from triples from the product distribution. Let's imagine that two features are conditionally correlated, some irrelevant information is shared between features only when they are from the same input and not from two inputs (from the same class). This correlation can be informative about a detail in the background, an unexpected shape in the image, that is rarely found in other samples from the same class. In that case, the product and joint distributions are easily distinguishable

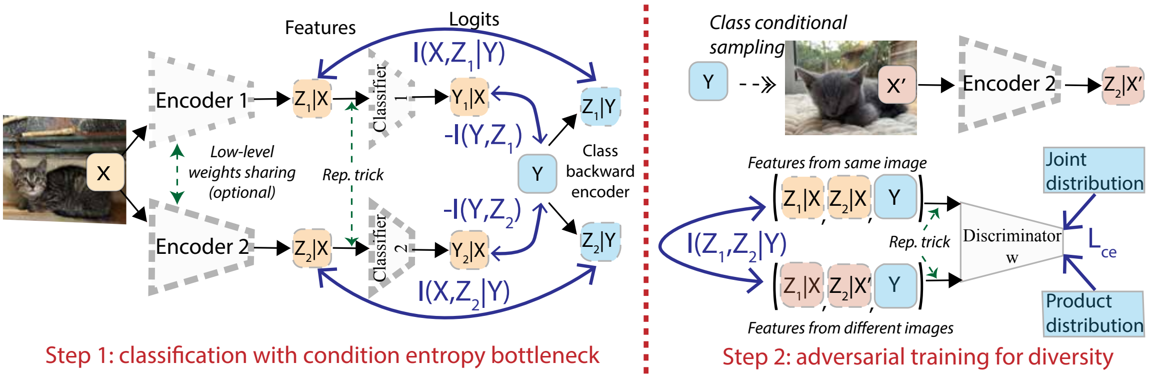


Figure 3.3. – **Learning strategy overview**. Blue arrows represent training criteria: (i) classification with conditional entropy bottleneck applied separately on each member, and (ii) adversarial training to delete irrelevant redundant information between members and increase diversity. X and X' belong to the same Y for *conditional redundancy* minimization.

by the discriminator. The first adversarial component will force the extracted features to reduce the correlation, and ideally one of the two features loses this information: it reduces redundancy and increases diversity. The impact of the second term is more complex; it may create fake correlations between features from different inputs. As we are not interested in a precise estimation of the CR, we get rid of this second term that, empirically, did not increase diversity. We end up with the following empirical estimation of CR.

$$\hat{\mathcal{L}}_{DV}^{CR}(e_1, e_2) = \frac{1}{|\mathcal{B}_J|} \sum_{(z_1,z_2,y)\in\mathcal{B}_J\sim p(e_1(z|x),e_2(z|x),y)} \log f(z_1, z_2, y). \tag{3.7}$$

Summary. First, we train each member for classification with VCEB from Equation (3.4), as shown in Step 1 from Figure 3.3. Second, following Step 2 from Figure 3.3, the discriminator, conditioned on the class Y, learns to distinguish features sampled from one image versus features sampled from two images belonging to the same Y. Simultaneously, both members adversarially [Goo+14] delete irrelevant shared information to reduce CR estimation from Equation (3.7) with differentiable signals. The full DICE loss is finally:

$$\mathcal{L}_{DICE}(\theta_1, \theta_2) = \text{VCEB}_{\beta_{ceb}}(\theta_1) + \text{VCEB}_{\beta_{ceb}}(\theta_2) + \delta_{cr}\hat{\mathcal{L}}_{DV}^{CR}(e_1, e_2). \tag{3.8}$$

3.3 Experiments

In this experimental section, we show that DICE improves accuracy by increasing diversity. In the published paper [Ram+21a] we provide more experiments for calibration, uncertainty estimation, out-of-distribution detection and co-distillation.

3.3.1 Baselines and concurrent works.

The strategies motivated by IB principles involve two main components:

- compression; the encoder can be deterministic, or variationally compressed non-conditionally (VIB) or conditionally (VCEB),
- redundancy; we can have no regularization, a non-conditional redundancy (R) or a conditional redundancy (CR) component.

To ablate the importance of each component, we compare the following approaches.

1. Ind. DE [Lak+17] refers to the *independent* deterministic deep ensembling without interactions between members. Other approaches below follow IB principles and thus their members have probabilistic encoders.
2. IB [Ale+17] from Equation (3.1) compresses with VIB.
3. CEB [Fis20] compresses with VCEB.
4. IBR follows Equation (3.2) and optimizes:

$$\mathcal{L}_{IBR}(\theta_1, \theta_2) = \text{VIB}_{\beta_{ib}}(\theta_1) + \text{VIB}_{\beta_{ib}}(\theta_2) + \delta_r \hat{\mathcal{L}}_{DV}^R(e_1, e_2), \tag{3.9}$$

   where: $\hat{\mathcal{L}}_{DV}^R(e_1, e_2) = \frac{1}{|\mathcal{B}_J|} \sum_{(z_1,z_2)\in\mathcal{B}_J} \log f(z_1, z_2)$ is simply the approximation of the non-conditional redundancy R (same as Equation (3.7) but without the label y).

5. CEBR benefits from VCEB with approximation of non-conditional redundancy R.

$$\mathcal{L}_{CEBR}(\theta_1, \theta_2) = \text{VCEB}_{\beta_{ceb}}(\theta_1) + \text{VCEB}_{\beta_{ceb}}(\theta_2) + \delta_r \hat{\mathcal{L}}_{DV}^R(e_1, e_2). \tag{3.10}$$

6. DICE combines VCEB with approximation of conditional redundancy CR.

Regarding the concurrent works, we consider two co-distillation strategies (ONE [Lan+18] and OKDDip [Che+20]), discussed in Remark 2.5. We also consider the best diversity regularization in predictions, ADP [Pan+19], decorrelating only the non-maximal predictions. Regarding the other works that enforce *diversity in features*, we found that increasing (L_1, L_2, $-\cos$) distances [Kim+18] between features fail: we speculate this is because they are not invariant to variables's symmetry. The work most similar to ours is [Sin+20b], proposing a diversity-inducing adversarial loss highly similar to our IBR.

3.3.2 Results on CIFAR

Table 3.1 reports the classification accuracy averaged over 3 runs for CIFAR-100 [Kri+09], while Table 3.2 focuses on CIFAR-10. {3,4,5}-{branch,net} refers to the training of {3,4,5} members {with,without} low-level weights sharing. We detail our implementation in the paper. We took most hyperparameter values from [Che+20]: those for adversarial training and information bottleneck were fine-tuned on a validation dataset made of 5% of the training dataset. **Bold** highlights best score.

Table 3.1. – CIFAR-100 ensemble classification accuracy (Top-1, %).

Name	Components		ResNet-32				ResNet-110		WRN-28-2		
	Div.	IB	3-branch	4-branch	5-branch	4-net	3-branch	4-branch	3-branch	4-branch	3-net
Ind. DE [Lak+17]			76.28$_{\pm 0.12}$	76.78$_{\pm 0.19}$	77.24$_{\pm 0.25}$	77.38$_{\pm 0.12}$	80.54$_{\pm 0.09}$	80.89$_{\pm 0.31}$	78.83$_{\pm 0.12}$	79.10$_{\pm 0.08}$	80.01$_{\pm 0.15}$
ONE [Lan+18]			75.17$_{\pm 0.35}$	75.13$_{\pm 0.25}$	75.25$_{\pm 0.22}$	76.25$_{\pm 0.32}$	78.97$_{\pm 0.24}$	79.86$_{\pm 0.25}$	78.38$_{\pm 0.45}$	78.47$_{\pm 0.32}$	77.53$_{\pm 0.36}$
OKDDip [Che+20]			75.37$_{\pm 0.32}$	76.85$_{\pm 0.25}$	76.95$_{\pm 0.18}$	77.27$_{\pm 0.31}$	79.07$_{\pm 0.27}$	80.46$_{\pm 0.35}$	79.01$_{\pm 0.19}$	79.32$_{\pm 0.17}$	80.02$_{\pm 0.14}$
ADP [Pan+19]	Pred.		76.37$_{\pm 0.11}$	77.21$_{\pm 0.21}$	77.67$_{\pm 0.25}$	77.51$_{\pm 0.25}$	80.73$_{\pm 0.38}$	81.40$_{\pm 0.27}$	79.21$_{\pm 0.19}$	79.71$_{\pm 0.18}$	80.01$_{\pm 0.17}$
IB [Ale+17]		VIB	76.01$_{\pm 0.12}$	76.93$_{\pm 0.24}$	77.22$_{\pm 0.19}$	77.72$_{\pm 0.12}$	80.43$_{\pm 0.34}$	81.12$_{\pm 0.19}$	79.19$_{\pm 0.35}$	79.15$_{\pm 0.12}$	80.15$_{\pm 0.13}$
CEB [Fis20]		VCEB	76.36$_{\pm 0.06}$	76.98$_{\pm 0.18}$	77.35$_{\pm 0.14}$	77.64$_{\pm 0.15}$	81.08$_{\pm 0.12}$	81.17$_{\pm 0.16}$	78.92$_{\pm 0.08}$	79.20$_{\pm 0.13}$	80.38$_{\pm 0.18}$
IBR Equation (3.9)	R	VIB	76.68$_{\pm 0.13}$	77.25$_{\pm 0.13}$	77.77$_{\pm 0.21}$	77.84$_{\pm 0.12}$	81.34$_{\pm 0.21}$	81.38$_{\pm 0.08}$	79.33$_{\pm 0.15}$	79.90$_{\pm 0.10}$	80.22$_{\pm 0.10}$
CEBR Equation (3.10)	R	VCEB	76.72$_{\pm 0.08}$	77.30$_{\pm 0.12}$	77.81$_{\pm 0.10}$	77.82$_{\pm 0.11}$	81.52$_{\pm 0.11}$	81.55$_{\pm 0.33}$	79.25$_{\pm 0.15}$	79.98$_{\pm 0.07}$	80.35$_{\pm 0.15}$
DICE Equation (3.8)	CR	VCEB	**76.89**$_{\pm 0.09}$	**77.51**$_{\pm 0.17}$	**78.08**$_{\pm 0.18}$	**77.92**$_{\pm 0.08}$	**81.67**$_{\pm 0.14}$	**81.93**$_{\pm 0.13}$	**79.59**$_{\pm 0.13}$	**80.05**$_{\pm 0.11}$	**80.55**$_{\pm 0.12}$

DICE surpasses concurrent approaches for ResNet and Wide-ResNet architectures. On CIFAR-100, DICE outperforms DE by $\{+0.60, +0.73, +0.84\}$ for {3,4,5}-branches ResNet-32. We also bring significant and systematic improvements to ADP [Pan+19]: *e.g.*, $\{+0.52, +0.30, +0.41\}$ for {3,4,5}-branches ResNet-32, $\{+0.94, +0.53\}$ for {3,4}-branches ResNet-110 and finally $+0.34$ for 3-networks WRN-28-2. Moreover, in Figure 3.4, an ensemble of 5 networks trained with DICE matches an ensemble of 7 networks trained independently; these results confirm that diversity approaches better leverage size. Finally, we observe that our gains are more important in the branch setup. This makes the TreeNet [Lee+15; Sze+15] architecture attractive, as it reduces memory cost at only a slight cost in diversity, which can be compensated with our DICE regularization.

Table 3.2. – CIFAR-10 ensemble classification accuracy (Top-1, %).

Archi	Structure	Ind. DE	ONE	OKDDip	ADP	IB	CEB	IBR	CEBR	DICE
ResNet-32	4-branch	94.75$_{\pm 0.08}$	94.41$_{\pm 0.05}$	94.86$_{\pm 0.08}$	94.92$_{\pm 0.04}$	94.76$_{\pm 0.12}$	94.93$_{\pm 0.11}$	94.91$_{\pm 0.14}$	94.94$_{\pm 0.12}$	**95.01**$_{\pm 0.09}$
ResNet-110	3-branch	95.62$_{\pm 0.06}$	95.25$_{\pm 0.08}$	95.21$_{\pm 0.09}$	95.43$_{\pm 0.12}$	94.54$_{\pm 0.07}$	94.65$_{\pm 0.05}$	95.68$_{\pm 0.05}$	95.67$_{\pm 0.06}$	**95.74**$_{\pm 0.08}$

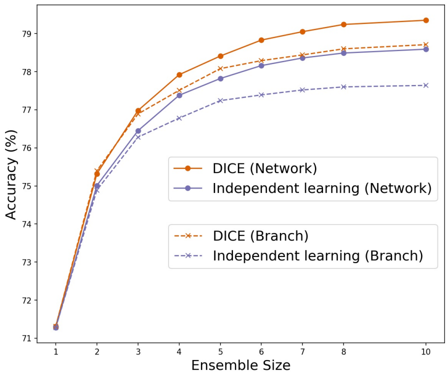


Figure 3.4. – DICE better leverages ensemble size on CIFAR-100 for ResNet-32. Without weights sharing, 5 networks trained with DICE match 7 networks trained independently. With low-level weights sharing, 4 branches trained with DICE match 7 traditional branches.

3.3.3 Individual accuracy-diversity trade-off

We now analyze how the two components of DICE modify the accuracy-diversity trade-off, with a 4-branches ResNet-32 on CIFAR-100. We measure diversity with the ratio-error [Aks03], and observe similar results for other diversity measures in the paper [Ram+21a]. In Figure 3.5, CEB has slightly higher diversity than independent trainings: diversity benefits from compression. ADP reaches higher diversity but sacrifices individual accuracies. On the contrary, co-distillation OKDDip [Che+20] sacrifices diversity for individual accuracies. DICE curve is above all others, and notably $\delta_{cr} = 0.2$ induces an *optimal trade-off between ensemble diversity and individual accuracies* on validation. CEBR reaches same diversity with lower individual accuracies, because some information relevant about Y was removed during non-conditional redundancy regularization.

Figure 3.6 shows the training dynamics for different values of δ_{cr} in DICE; in particular, $\delta_{cr} = 0.0$ corresponds to CEB. Starting from random initializations, diversity always begins small. Then larger values of δ_{cr} minimizes the estimated CR in features and increases diversity in predictions compared to CEB. Specifically a very high value ($\delta_{cr} = 0.6$) creates too much diversity as it drastically reduces individual performances. On the contrary, a negative value ($\delta_{cr} = -0.025$) can decrease diversity.

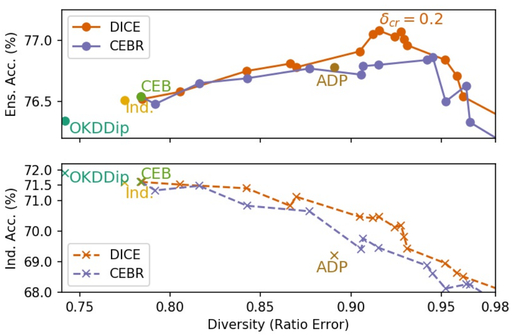


Figure 3.5. – **Ensemble diversity/individual accuracy trade-off** for different strategies. DICE (r. CEBR) is learned with different δ_{cr} (r. δ_r).

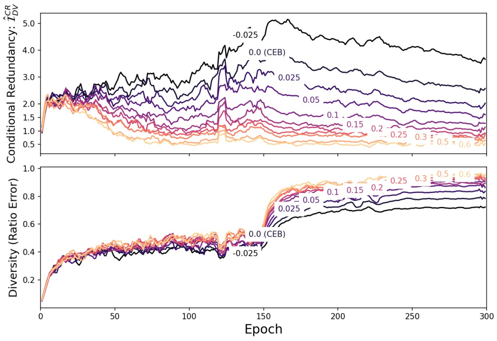


Figure 3.6. – **Impact of the diversity coefficient** δ_{cr} in DICE on the training dynamics on validation: CR is negatively correlated with diversity.

3.3.4 Results on DomainBed

Previous experiments were conducted on CIFAR with a ID test dataset: in Table 3.3, we now provide OOD results on two datasets from DomainBed [Gul+21] where diversity shifts dominate. On PACS, the training domains are "Cartoon", "Photo" and "Sketch"; the OOD accuracy is reported on PACS's "Art" domain. On OfficeHome, the training domains are "Clipart", "Product" and "Real"; the OOD accuracy is reported on OfficeHome's "Art" domain. All methods use 20 runs with hyperparameters sampled from the mild range in Table 2.1, the ensembling methods use $M = 2$ ResNet-50, and we con-

Table 3.3. – OOD accuracy (%, ↑) on DomainBed. "Art" is the OOD domain for both datasets.

Algorithm	PACS	OfficeHome
ERM	87.6 ± 0.4	62.9 ± 1.3
Ind. DE	88.2 ± 0.2	65.3 ± 0.6
DICE	**88.4** ± 0.2	**65.7** ± 0.5

sider DICE with $\delta_{cr} = 0.2$. We consistently observe that ensembling and DICE improve performances on these datasets. The subsequent chapters will enrich our analysis on DomainBed [Gul+21], which actually appeared after DICE [Ram+21a] was published.

3.4 Conclusion

In this chapter, motivated by arguments from information theory, we derive a novel adversarial training strategy for ensemble. We tackle the trade-off between individual accuracies and ensemble diversity by deleting irrelevant information across members. We improve the accuracy on CIFAR-10 and CIFAR-100 compared to the DE baseline. We show that DICE can be combined with efficient ensembling strategies such as TreeNets [Lee+15], where members share low-level weights.

Perspectives. Despite its merits, this initial approach has two critical limitations. First, DICE requires to learn simultaneously all models, inhibiting parallelization during training. Moreover, subsequent research [Wor+23; Abe+23] revealed the limitations of diversity regularizations when applied to larger datasets such as ImageNet. Given these limitations, we will explore implicit strategies to bolster diversity in the rest of this thesis.

Chapter 4

DIVERSE WEIGHT AVERAGING FOR OUT-OF-DISTRIBUTION GENERALIZATION

4.1 Introduction

The framework of *foundation models* [Bom+21] is fueling a spectacular adoption of ML solutions for real-world applications: these models are pre-trained on large-and-diverse data [Fan+22; Ngu+22; Abn+22] and easy [Oqu+14] to adapt to downstream tasks. Though this transfer learning helps, models still struggle to generalize on out-of-distribution (OOD) samples [Hen+19a; Tao+20; Gul+21; Hen+21]. Increased OOD generalization would enable the responsible use of ML in real-world applications where robustness are critical [Tay+16; Zec+18; DeG+21], as previously described in Section 1.3. Thus, how to best fine-tune foundation models for OOD generalization is a key topic of research.

On the reference DomainBed benchmark [Gul+21] evaluating different fine-tuning strategies for OOD generalization, the standard empirical risk minimization (ERM) was recently outperformed by moving average (MA) [Cha+21a; Arp+21]: MA simply weight averages (WA) the various checkpoints (a.k.a. snapshots) collected along a training trajectory [Izm+18]. [Cha+21a] argue that this WA succeeds because it finds solutions in flatter regions of the loss landscape.

In this chapter, we challenge this flatness-based analysis, and show its limitations. We then propose a new explanation for the success of WA in OOD based on its similarities with functional ensembling ENS: we show that averaging the models or the weights behave similarly as long as the weights remain sufficiently close. Based on this similarly and the bias-variance-covariance Equation (BVC) for ENS, we obtain a bias-variance-covariance-locality decomposition of WA's expected error. It contains four terms:

1. *first* the bias increasing under correlation shift [Ye+22], see Section 2.2.2,
2. *second*, the variance increasing under diversity shift [Ye+22], see Section 2.2.1,
3. *third*, the covariance decreasing when models are diverse, see Section 2.3.3,
4. *finally*, a new locality condition on the weights, enforcing averageability.

This explains WA's success (i) for diversity shift (where variance dominates), (ii) as long as the models are functionally diverse (to reduce variance) (iii) yet close in the weight space (to ensure averageability).

Based on this analysis, we aim at enhancing diversity while preserving weight averageability. We thus propose Diverse Weight Averaging (DiWA) averaging weights obtained from independent fine-tunings, all starting from a shared pre-trained initialization, yet with different hyperparameters and data orders. The motivation is that those models are more diverse than those obtained along a single run [For+19a; Gon+22]. Moreover, because the shared initialization is pre-trained, the weights of the models remain close enough to be averaged. This follows the linear mode connectivity (LMC) [Fra+20; Nag+19], as previously explored for fine-tuning in [Ney+20].

- We reveal (theoretical and empirical) limitations from existing flatness-based analysis of weight averaging (WA) (Section 4.2.2).
- We propose a new analysis of WA based on its similarity with ensembling, leading to a new bias-variance-covariance-locality decomposition of its expected error (Section 4.3).
- We propose DiWA to enhance the diversity across averaged models by decorrelating their training procedures: in practice, these models are obtained from independent runs with different hyperparameters (Section 4.4).
- Experimentally, DiWA improves performances on the competitive DomainBed benchmark, without any inference overhead (Section 4.6).

This chapter has led to the publication of Alexandre Ramé, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. “Diverse Weight Averaging for Out-of-Distribution Generalization”. In: *NeurIPS*. 2022.

4.2 Context

4.2.1 Fine-tuning for OOD generalization

Problem and notations. We start by describing our setup. We train a deep neural network (DNN) $f(\cdot, \theta) : \mathcal{X} \to \mathcal{Y}$ with weights θ, that should maximize the test accuracy $\mathrm{acc}_T(\theta)$, or equivalently minimize the test generalization error $\mathcal{E}_T(\theta)$. To this end, $f(\cdot, \theta)$ should approximate the labelling function f_T on $\mathcal{X}_T$. However, this is complex in OOD because we only have data from domain S in training, related yet different from T. The differences between S and T are due to distribution shifts (*i.e.*, the fact that $p_S(X, Y) \neq p_T(X, Y)$) which are decomposed per [Ye+22] into (i) *diversity shift*, when marginal distributions differ (*i.e.*, $p_S(X) \neq p_T(X)$), and (ii) *correlation shift*, when posterior distributions differ (*i.e.*, $p_S(Y|X) \neq p_T(Y|X)$ and $f_S \neq f_T$).

Vanilla fine-tuning. For OOD generalization, transfer learning [Oqu+14; Kir+22] from a foundation model [Bom+21] and then supervised fine-tuning with ERM [Vap92] is the standard strategy [Gul+21]. Specifically, from a featurizer pre-trained on a large dataset such as ImageNet [Rus+15], users usually launch multiple fine-tunings on the target task with different hyperparameters, and then select the best one based on some validation metric [Gul+21].

Weight averaging over epochs. Recently, WA strategies came to the foreground [Sze+16; Izm+18; Dra+18], as previously described in Section 2.3.4. While fine-tuning a pre-trained model, they saved and averaged checkpoints every few epochs to build the final model. Due to the non-linear nature of DNNs, the efficacy of WA was a surprising observation, that [Fra+20] latter called the linear mode connectivity (LMC).

Observation 4.1 (LMC with different epochs [Izm+18])**.** *Two weights θ_1 and θ_2, obtained at two different epochs of the same fine-tuning, satisfy the LMC: for all $\lambda \in [0, 1]$,*

$$\mathrm{acc}_T((1-\lambda)\cdot\theta_1 + \lambda\cdot\theta_2) \gtrsim (1-\lambda)\cdot\mathrm{acc}_T(\theta_1) + \lambda\cdot\mathrm{acc}_T(\theta_2). \tag{4.1}$$

The LMC holds if the accuracy of the interpolated weights is above the interpolated accuracy. Consistently with Observation 4.1, recent works [Arp+21; Cha+21a; Wor+22b; Kad22] weight average checkpoints along training to improve accuracies. In particular, [Arp+21] showed that the simple moving average (MA) strategy, which samples uniformly along training, is sota on DomainBed [Gul+21].

4.2.2 Flatness analysis of WA

4.2.2.1 Flatness generalization bound

Despite its empirical success, WA strategies lack theoretical foundations. Previous analysis [Cha+21a] argues that WA succeeds because it flattens the loss landscape. Specifically, the "flatness" Theorem 1 in [Cha+21a] upper bounds target generalization error by a sum of three terms.

1. The first and most important term involves the solution's ID flatness, and is usually estimated by the trace of the Hessian [Din+17b; Pet+21; Yao+20], *i.e.*, the sum of its eigenvalues.
2. The second term is a domain divergence between the source and target marginal distributions, which grows under distribution shifts.
3. The last (and not analyzed) term involves the VC dimension of the networks.

4.2.2.2 Limitations of the flatness analysis

In Figure 4.1, we analyze flatness via the Hessian trace, computed with [Yao+20], on the OfficeHome dataset [Ven+17], using "Art" as the OOD test domain. We confirm that WA flattens the loss landscape; yet this does *not* mean that WA succeeds because of it. Indeed, several theoretical inconsistencies and unanswered empirical observations remain.

1. **Flatness and domain shifts.** First, the flatness-based analysis is not specific to OOD. Indeed, the flatness and domain divergence in the upper bound of [Cha+21a] from Section 4.2.2.1 are not interacting; more flatness does not reduce domain divergence,

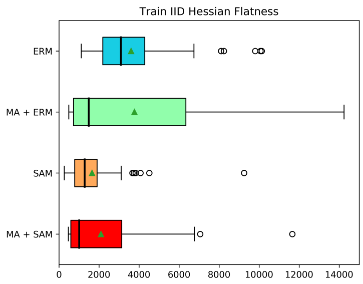


Figure 4.1. – Train Hessian trace (↓)

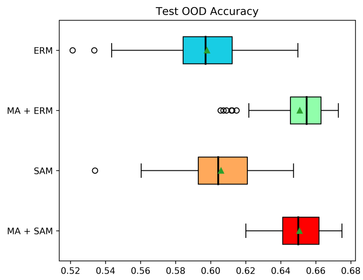


Figure 4.2. – Test OOD accuracy (↑).

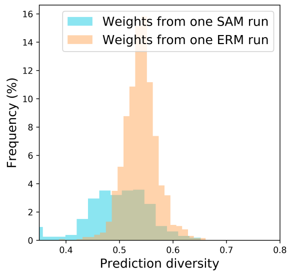


Figure 4.3. – Test diversity in ratio-error [Aks03] (↑).

and the OOD error is actually uncontrolled. In other words, additional flatness may improve generalization in general, but this bound tells us nothing about an hypothetic specific impact for OOD.

2. **WA vs. SAM.** Second, the upper bound does not clarify why MA outperforms flatness-based methods such as sharpness-aware minimization (SAM) [For+21], which is an optimization strategy that directly optimizes flatness along training. In Figure 4.1, we observe that SAM actualy finds flatter minimas than MA [Arp+21]. Yet, this is not reflected in OOD accuracies in Figure 4.2 where MA outperforms SAM. These results have been confirmed empirically in [Cha+21a]. In conclusion, flatness is not sufficient to explain why WA works so well in OOD, because SAM has better flatness but worse OOD results.

3. **Combining WA and SAM.** We investigate another empirical inconsistency when combining MA and SAM. As argued in [Kad+22], we confirm in Figure 4.1 that MA + SAM usually leads to flatter minimas than MA and WA alone. Yet, MA does not benefit from SAM in Figure 4.2. Moreover, [Cha+21a] also showed in their Table 4 that SWAD + ERM outperforms SWAD + SAM. This is not explained by the upper bound from [Cha+21a], which argues that more flatness should improve OOD results.

4.3 Bias-variance-covariance-locality analysis

To replace this flatness-based analysis, we propose a new one based on the similarity between WA and functional ensembling, as shown in Section 4.3.1. We then decompose WA's expected error in Section 4.3.2. This analysis suggests three (conflicting) success criterion for WA, as the averaged models should be:

1. individually accurate (to reduce the bias term),

2. diverse functionally (to reduce the covariance term),

3. close in the weight space (or at least satisfying the linear mode connectivity (LMC), to reduce the locality term).

4.3.1 WA approximates ensembling

Notations. We recall the notations previously introduced in Chapter 2. When needed, we explicitly write $\theta(l_S)$ to refer to the weights obtained after stochastic minimization on $\mathcal{D}_S$ w.r.t. θ under $l_S = \{\mathcal{D}_S, c\}$ a learning procedure, where:

- the training dataset $\mathcal{D}_S$ from S is composed of n_S *i.i.d.* samples from $p_S(X, Y)$,
- the configuration c contains all other sources of randomness in the learning.

We study the benefits of averaging M weights $\{\theta_i\}_{i=1}^M = \{\theta(l_S^{(i)})\}_{i=1}^M$ obtained from the M learning procedures $L_S^M = \{l_S^{(i)}\}_{i=1}^M$ (potentially correlated yet) identically distributed (*i.d.*). Their WA is $f_{\text{WA}} = f(\cdot, \theta_{\text{WA}})$ where $\theta_{\text{WA}} = \theta_{\text{WA}}(L_S^M) = \frac{1}{M}\sum_{i=1}^M \theta_i$.

Ensembling. To decompose WA's error, we leverage its similarity with functional ensembling $f_{\text{ENS}} = \frac{1}{M}\sum_{i=1}^M f(\cdot, \theta_i)$. Specifically, Lemma 4.1 establishes that f_{WA} is a first-order approximation of f_{ENS} when $\{\theta_i\}_{i=1}^M$ are close in the weight space. The proof is in Appendix C.3.

Lemma 4.1 (WA and ENS. Adapted from [Izm+18; Wor+22a].). *Given $\{\theta_i\}_{i=1}^M$ with learning procedures $L_S^M = \{l_S^{(i)}\}_{i=1}^M$. Denoting $\Delta_{L_S^M} = \max_{i=1}^M \|\theta_i - \theta_{WA}\|_2$, $\forall (x, y) \in \mathcal{X} \times \mathcal{Y}$:*

$$f_{WA}(x) = f_{ENS}(x) + \mathcal{O}(\Delta^2_{L_S^M}) \text{ and } \ell\left(f_{WA}(x), y\right) = \ell\left(f_{ENS}(x), y\right) + \mathcal{O}(\Delta^2_{L_S^M}). \qquad (4.2)$$

4.3.2 Bias-variance-covariance-locality decomposition

This similarity is useful since Equation (BV) was extended into a bias-variance-covariance decomposition for ENS in Equation (BVC) following [Ued+96; Bro+05a]. Thus, by inserting Equation (BVC) into Equation (4.2), we obtain the following Proposition 4.1.

Proposition 4.1 (Bias-variance-covariance-locality decomposition of WA's error.). *With $\bar{f}_S(x) = \mathbb{E}_{l_S}[f(x, \theta(l_S))]$, under i.d. $L_S^M = \{l_S^{(i)}\}_{i=1}^M$, the expected generalization error on domain T of WA over the joint distribution of L_S^M, is:*

$$\mathbb{E}_{L_S^M}\mathcal{E}_T(\theta_{WA}(L_S^M)) = \mathbb{E}_{p_T}[\text{bias}(x, y)^2 + \frac{\text{var}(x)}{M} + \frac{M-1}{M}\text{cov}(x)] + \mathcal{O}(\bar{\Delta}^2), \qquad \text{(BVCL)}$$

where:

$$\begin{aligned}
\text{bias}(x, y) &= y - \bar{f}_S(x), \\
\text{var}(x) &= \mathbb{E}_{l_S}\left[\left(f(x, \theta(l_S)) - \bar{f}_S(x)\right)^2\right], \\
\text{cov}(x) &= \mathbb{E}_{l_S, l'_S}(f(x, \theta(l_S)) - \bar{f}_S(x))(f(x, \theta(l'_S)) - \bar{f}_S(x)), \\
\bar{\Delta}^2 &= \mathbb{E}_{L_S^M}\Delta^2_{L_S^M} \quad \textit{where} \quad \Delta_{L_S^M} = \max_{i=1}^M \|\theta_i - \theta_{WA}\|_2.
\end{aligned}$$

The locality term $\bar{\Delta}^2$ is the expected squared maximum distance between weights and their average.

4.3.3 Locality, diversity and linear mode connectivity

Equation (BVCL) involves four terms. The three first terms are those from functional ensembling, previously analyzed in Section 2.3.2; namely the bias (the same as each of its *i.d.* members), a variance divided by M and a covariance term (analyzed in Section 2.3.3, measuring diversity).

In contrast, the last locality term $\mathcal{O}(\bar{\Delta}^2)$ is specific to weight averaging; it ensures that WA approximates ENS by constraining the weights to remain close. Overall, locality and diversity are two antagonistic terms: to reduce WA's error, we thus seek a good trade-off between locality and diversity. In practice, we consider that the main goal of this locality term is to ensure that the weights are averageable despite the non-linearities in the DNN such that WA's error does not explode. This is why in Section 4.4, we empirically relax this locality constraint and simply require that the weights satisfy the linear mode connectivity (LMC) [Fra+20]. We empirically verify later in Figure 4.5 that the approximation $f_{\text{WA}} \approx f_{\text{ENS}}$ remains valid even in this case.

Conclusion. We have showed in Section 2.2 that bias dominates under correlation shift and variance dominates under diversity shift. Therefore, in a similar fashion to functional ensembling, our analysis suggests that *WA would be effective against diversity shift when M is large and when its members are diverse but linearly mode connected.*

4.3.4 Superiority of our analysis

We summarize why analyzing WA through the prism of functional ensembling and variance reduction fixes failures from previous flatness-based analysis [Cha+21a].

1. First, our analysis explains why WA performs well under diversity shift, and less in-distribution (as observed in previous works [Arp+21; Wor+22a]) or not under correlation shift (as validated in Section 4.6.3.4). This is because variance is mostly an issue under diversity shift, and that variance reduction is useless under correlation shift where bias dominates. This was not predicted by the flatness analysis. We also explicitly explain why increasing the number of weights M helps.

2. Our analysis explains why WA and SAM are not complementary in OOD, as visible in Figure 4.2, in contradiction with what was argued in [Cha+21a]. Indeed, we observe in Figure 4.3 that the diversity across two checkpoints collected along a SAM trajectory is much lower than along a standard SGD trajectory. Therefore, the gain in individual accuracies of models trained with SAM cannot compensate the decrease in diversity.

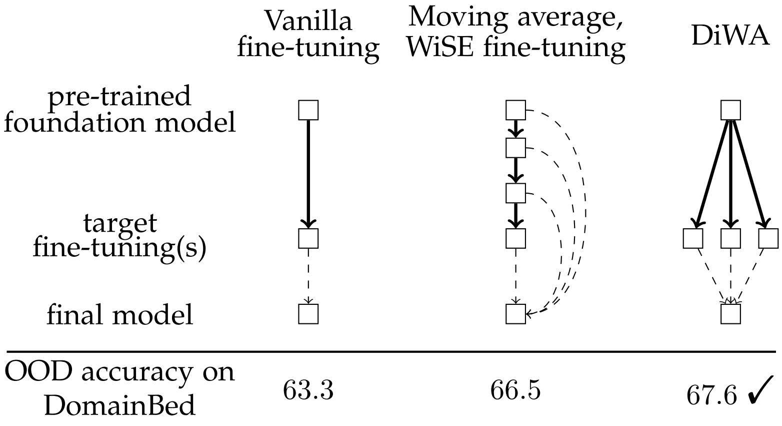

Vanilla fine-tuning
Moving average, WiSE fine-tuning
DiWA
pre-trained foundation model
target fine-tuning(s)
final model
OOD accuracy on DomainBed
63.3
66.5
67.6 ✓

Figure 4.4. – The different fine-tuning strategies discussed in this chapter: vanilla fine-tuning [Oqu+14], moving average (MA) [Izm+18] and variants [Wor+22b], and our novel DiWA [Ram+22b]. They all start from a pre-trained foundation model, before fine-tuning on the target task (thick solid arrows ⟶). The fine-tuned weights are used as is, or are averaged (dashed arrows - - ➝) into a final model.

4.4 DiWA

4.4.1 Motivation: weight averaging from different runs for more diversity

Limitations of single-run WA. Our analysis suggests that WA performances can be improved by increasing diversity across the averaged models. Yet, previous WA methods [Cha+21a; Arp+21] only average weights obtained along a single run. This corresponds to highly correlated procedures sharing the same initialization, hyperparameters, batch orders, data augmentations and noise, that only differ by the number of training steps. The models are similar, which does not leverage the full potential of WA.

DiWA. Our DiWA pushes the envelope of WA techniques, and reduce the OOD expected error in Equation (BVCL) by decorrelating the learning procedures $\{l_S^{(i)}\}_{i=1}^M$. The weights are obtained from $M \gg 1$ different runs, with different hyperparameters (learning rate, weight decay and dropout probability), batch orders, data augmentations (*e.g.*, random crops, horizontal flipping, color jitter, grayscaling), stochastic noise and number of training steps. Thus, the models averaged in DiWA collected from those multiple runs would be more diverse than the snapshots collected along a single training run. However, this gain in diversity may break the locality requirement from Section 4.3.3; below, we detail the empirical conditions under which DiWA succeeds.

4.4.2 On the importance of pre-training for weight averaging

The success of DiWA relies on the following observation from [Ney+20]: "there is no performance barrier between two instances of models trained from pre-trained weights". Two *independent* fine-tunings from a shared pre-trained model satisfy the LMC, and thus can be connected along a linear path where error remains low [Nag+19; Fra+20]. Formally:

Observation 4.2 (LMC with different runs [Ney+20])**.** *The LMC holds between θ_1 and θ_2 fine-tuned on the target task initialized from a shared pre-trained model.*

Shared pre-trained initialization. This Observation 4.2 means that DiWA would work if the models from different runs are initialized from a shared pre-trained model. In our experiments, our featurizer will be pre-trained on ImageNet [Kri+12], following the standard setup from DomainBed. Regarding the classifier initialization, we test two methods. The first is the random initialization, which may distort the features [Kum+22]. The second is linear probing (LP) [Kum+22]: it first learns the classifier (while freezing the featurizer) to serve as a shared initialization; then, LP fine-tunes the featurizer and the classifier together in the M subsequent runs. This two-step procedure would reduce the locality term, as [Kum+22] showed that fine-tuned weights then tend to remain closer.

4.4.3 Mild hyperparameter search

In addition to the shared pre-trained initialization, we add a constraint on the hyperparameters. This is based on the observation from Figure 4.10 that extreme hyperparameter ranges may to lead poor performances after WA; we speculate that weights obtained from extremely different hyperparameters (in particular different learning rates) may belong to different regions of the loss landscape. In our experiments, we thus use the mild hyperparameter ranges defined in Table 2.1. This mild hyperparameter search empirically guarantees high diversity and averageability.

4.4.4 Weight selection

The last stage of our approach is to choose which weights to average among those available. Our standard *uniformly* averages all weights; it is practical but may underperform when some runs have failed. A possible refinement proposed in [Wor+22a] is a *greedy* selection, restricting the number of selected weights: weights are ranked in decreasing order of ID validation accuracy and sequentially added only if they improve DiWA's ID validation accuracy.

4.5 Related work.

The LMC is at the heart of DiWA but also of other recent works [Wor+22b; Mat+22; Wor+22a] that average weights with fewer constraints [Wor+22b; Mat+22; Wor+22a; Gup+20; Cho+22b; Wor+21] than traditional moving average [Izm+18]. To increase diversity across averaged models, [Mad+19] used a high learning rate; [Ben+21] encouraged the weights to encompass more volume; [Wor+21] minimized cosine similarity between weights; [Izm+19] used a tempered posterior. In particular, the concurrent "model soups" introduced by Wortsman *et al.* [Wor+22a] two months before DiWA follows a similar strategy: yet, the theoretical analysis and the goals of these two works are different. Regarding the motivation, DiWA aims at combining diverse weights, and proposes a general framework to average weights obtained in various ways. In contrast, [Wor+22a] challenges the standard model selection after a grid search. Regarding the task, [Wor+22a] and DiWA complement each other: while [Wor+22a] demonstrate robustness on several ImageNet variants, we improve the sota on the multi-domain DomainBed benchmark against other established OOD methods (Section 4.6.1). Thus, DiWA and [Wor+22a] are theoretically complementary with different motivations and applied successfully for different tasks.

4.6 Experiments

4.6.1 Experimental setup

Datasets. We first show DiWA improves performance on DomainBed [Gul+21], previously described in Section 2.1.3, including 5 multi-domain real-world datasets: PACS [Li+17], VLCS [Fan+13], OfficeHome [Ven+17], TerraIncognita [Bee+18] and DomainNet [Pen+19]. Criticially, [Ye+22] showed that *diversity shift dominates in these datasets*. Each domain is successively considered as the target T while other domains are merged into the source S. The validation dataset is sampled from S.

Baselines. ERM is the standard empirical risk minimization. CORAL [Sun+16] is the best approach based on domain invariance. SWAD (Stochastic Weight Averaging Densely) [Cha+21a] and MA [Arp+21] average weights along one training trajectory but differ in their weight selection strategy. Specifically, SWAD [Cha+21a] uses a "loss-aware" strategy involving three additional hyperparameters (a patient parameter, an overfitting patient parameter and a tolerance rate); in contrast, MA [Arp+21] is easy to implement as it simply combines all checkpoints uniformly starting from batch 100 until the end of training. We also report the scores obtained in [Arp+21] for the costly deep ensembling (DE*) [Lak+17] of $M = 6$ models with different classifier initializations, where the symbol "$*$" marks the large inference overhead.

Our runs. ERM and DiWA share the same training protocol in DomainBed: yet, instead of keeping only one run from the grid-search, DiWA averages M weights. In practice, we sample 20 configurations from the mild hyperparameter distributions detailed in Table 2.1 and report the mean and standard deviation across 3 data splits. For each run, we select the weights of the epoch with the highest ID validation accuracy. The ENS* averages the predictions of all $M = 20$ models (with shared initialization). DiWA-greedy selects $1 \leq M \leq 20$ weights while DiWA-uniform averages all $M = 20$ weights. DiWA† averages uniformly the $M = 3 \times 20 = 60$ weights from all 3 data splits.

4.6.2 Results on DomainBed

We report our main results in Table 4.1: best results are in **bold** and the second best are underlined. With a randomly initialized classifier, DiWA†-uniform is the best on PACS, VLCS and OfficeHome: DiWA-uniform is the second best on PACS and OfficeHome. On TerraIncognita and DomainNet, DiWA is penalized by some bad runs, filtered in DiWA-greedy which improves results on these datasets. Classifier initialization with LP [Kum+22] improves all methods on OfficeHome, TerraIncognita and DomainNet. On these datasets, DiWA† increases MA by 1.3, 0.5 and 1.1 points respectively. After averaging, DiWA† with LP reaches 68.0%, improving SWAD by 1.1 points.

Table 4.1. – **Accuracies ($\%, \uparrow$) on DomainBed** [Gul+21] benchmark evaluating OOD generalization. The classifiers are initialized randomly or with linear probing (LP) [Kum+22]. The symbol "$*$" indicates inference overhead in functional ensembling. The symbol "†" indicates the averaging of all weights across 3 data splits.

	Algorithm	Weight selection	Init	PACS	VLCS	OfficeHome	TerraInc	DomainNet	Avg
	ERM	ID val	Random	85.5 ± 0.2	77.5 ± 0.4	66.5 ± 0.3	46.1 ± 1.8	40.9 ± 0.1	63.3
	CORAl [Sun+16]	ID val		86.2 ± 0.3	78.8 ± 0.6	68.7 ± 0.3	47.6 ± 1.0	41.5 ± 0.1	64.6
	SWAD [Cha+21a]	Loss-aware		88.1 ± 0.1	79.1 ± 0.1	70.6 ± 0.2	50.0 ± 0.3	46.5 ± 0.1	66.9
	MA [Arp+21]	Uniform		87.5 ± 0.2	78.2 ± 0.2	70.6 ± 0.1	50.3 ± 0.5	46.0 ± 0.1	66.5
	DE* [Arp+21]	Uniform: $M = 6$		87.6	78.5	70.8	49.2	**47.7**	66.8
Our runs	ERM	ID val	Random	85.5 ± 0.5	77.6 ± 0.2	67.4 ± 0.6	48.3 ± 0.8	44.1 ± 0.1	64.6
	MA [Arp+21]	Uniform		87.9 ± 0.1	78.4 ± 0.1	70.3 ± 0.1	49.9 ± 0.2	46.4 ± 0.1	66.6
	ENS*	Uniform: $M = 20$		88.0 ± 0.1	78.7 ± 0.1	70.5 ± 0.1	51.0 ± 0.5	47.4 ± 0.2	67.1
	DiWA	Greedy: $M \leq 20$		87.9 ± 0.2	79.2 ± 0.1	70.5 ± 0.1	50.5 ± 0.5	46.7 ± 0.1	67.0
	DiWA	Uniform: $M = 20$		88.8 ± 0.4	79.1 ± 0.2	71.0 ± 0.1	48.9 ± 0.5	46.1 ± 0.1	66.8
	DiWA†	Uniform†: $M = 60$		**89.0**	**79.4**	71.6	49.0	46.3	67.1
	ERM	ID val	LP	85.9 ± 0.6	78.1 ± 0.5	69.4 ± 0.2	50.4 ± 1.8	44.3 ± 0.2	65.6
	MA [Arp+21]	Uniform		87.8 ± 0.3	78.5 ± 0.4	71.5 ± 0.3	51.4 ± 0.6	46.6 ± 0.0	67.1
	ENS*	Uniform: $M = 20$		88.1 ± 0.3	78.5 ± 0.1	71.7 ± 0.1	50.8 ± 0.5	47.0 ± 0.2	67.2
	DiWA	Greedy: $M \leq 20$		88.0 ± 0.3	78.5 ± 0.1	71.5 ± 0.2	51.6 ± 0.9	**47.7** ± 0.1	67.5
	DiWA	Uniform: $M = 20$		88.7 ± 0.2	78.4 ± 0.2	72.1 ± 0.2	51.4 ± 0.6	47.4 ± 0.2	67.6
	DiWA†	Uniform†: $M = 60$		**89.0**	78.6	**72.8**	**51.9**	**47.7**	**68.0**

4.6.3 Empirical validation of theoretical insights

We now validate key findings about the empirical similarity between WA and ENS, and then about the role of diversity in WA. To this end, we consider several collections of weights $\{\theta_i\}_{i=1}^{M}$ ($2 \leq M < 10$) trained on the "Clipart", "Product" and "Photo" domains from OfficeHome [Ven+17] with a shared random initialization and mild hyperparameter ranges. These weights are indifferently sampled from a single run (every 50 batches) or from different runs. They are evaluated on "Art", the fourth domain from OfficeHome considered as OOD.

4.6.3.1 WA vs. ENS

Figure 4.5 validates Lemma 4.1 and that weight averaging (WA) and ensembling (ENS) perform similarly: most dots are close to the diagonal. Moreover, a larger M improves results, motivating averaging as many weights as possible, consistently with Equation (BVCL). WA has a fixed inference time which allows it to consider larger M. In contrast, the functional ensembling ENS of M models require M forwards at inference; thus ENS is computationally impractical for large M.

Actually, we observe that WA slightly but consistently beats ENS, (i) in OOD (ii) when weights share the same initialization and (iii) hyperparameters are sampled from mild ranges. We provide a preliminary explanation to this surprising phenomenon in Section 7.2.2. Critically, in Table 4.2, we show that this is no longer the case when we relax the two last constraints. *First*, when the classifiers' initializations vary, ENS improves thanks to this additional diversity; in contrast, DiWA degrades because weights are no longer averageable. *Second*, when the hyperparameters are sampled from extreme ranges (defined in Table 2.1), performance drops significantly for DiWA, but much less for ENS. This highlights a limitation of DiWA, which requires weights that are linearly mode connected. In contrast, ENS are computationally expensive (and even impractical for large M), but can leverage additional sources of diversity, such as diverse initializations and extreme hyperparameters.

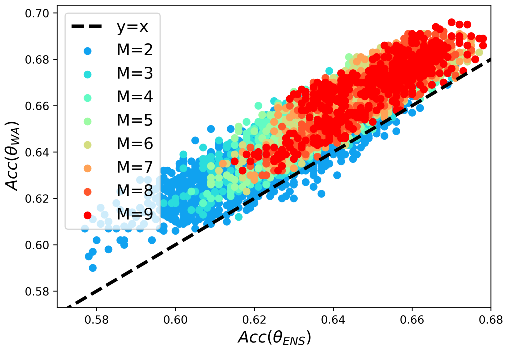


Figure 4.5. – Each dot displays the OOD accuracy (↑) of WA vs. ENS when combining M models.

Table 4.2. – **DiWA vs. ENS** on domain "Art" from OfficeHome when varying initialization and hyperparameter ranges. Best accuracy (%, ↑) on each setting is in **bold**.

Configuration		$M = 20$		$M = 60$	
Shared classifier init	Mild hyperparameter ranges	DiWA	ENS*	DiWA	ENS*
✓	✓	**67.3** ± 0.2	66.1 ± 0.1	**67.7**	66.5
✗	✓	65.0 ± 0.5	**67.5** ± 0.3	65.9	**68.5**
✓	✗	56.6 ± 0.9	**64.3** ± 0.4	59.5	**64.7**

4.6.3.2 Diversity and accuracy

We validate in Figure 4.6 that f_{WA} benefits from diversity. Here, we measure diversity with the ratio-error [Aks03], previously defined in Section 2.3.3.1. A higher average over the $\binom{M}{2}$ pairs means that members are less likely to err on the same inputs thus more diverse. Specifically, the gain of $\text{Acc}(\theta_{\text{WA}})$ over the mean individual accuracy $\frac{1}{M}\sum_{i=1}^{M}\text{Acc}\left(\theta_i\right)$ increases with diversity. For example, when $M = 9$ weights are averaged, the accuracy gain increases by 0.297 per unit of additional diversity in prediction.

Moreover, this phenomenon intensifies for larger M. In Figure 4.7, we indicate the slope of the linear regressions relating diversity to accuracy gain at fixed M (between 2 and 9). We observe that the linear regression's slope (*i.e.*, the accuracy gain per unit of diversity) increases with M. This confirms that diversity becomes more important for large M, consistently with the $(M-1)/M$ factor in front of $\text{cov}\left(x\right)$ in Equation (BVCL).

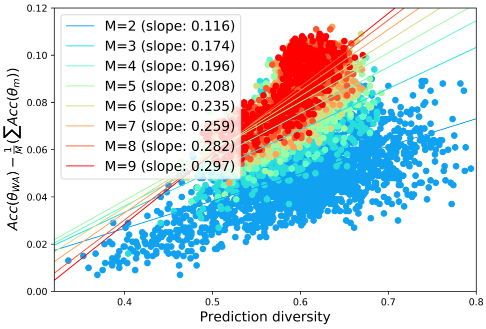


Figure 4.6. – Each dot displays the accuracy (↑) gain of WA over its members vs. the prediction diversity [Aks03] (↑) for M models.

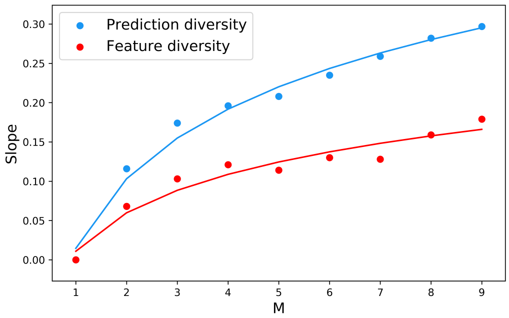


Figure 4.7. – The slopes relating diversity (in prediction [Aks03] or in features [Kor+19]) to accuracy gain, increases with M.

4.6.3.3 Increasing diversity thus accuracy via different runs.

Now we investigate the difference between sampling the weights from a single run or from different runs. Figure 4.8 *first* shows that diversity increases when weights come from different runs. *Second*, this diversity gain is reflected on the OOD accuracies in Figure 4.9; here, we rank by validation accuracy the 60 weights obtained (i) from 60 different runs and (ii) along 1 well-performing run. We then consider the WA of the top M weights as M increases from 1 to 60. Both have initially the same performance and improve with M; yet, WA of weights from different runs gradually outperforms the single-run WA. *Finally*, Figure 4.10 shows that this holds only for mild hyperparameter ranges and with a shared initialization. Otherwise, when hyperparameter distributions are extreme (as defined in Table 2.1) or when classifiers are not similarly initialized, DiWA may perform worse than its members due to a violation of the locality condition. These experiments confirm that *diversity is key as long as the weights remain averageable*.

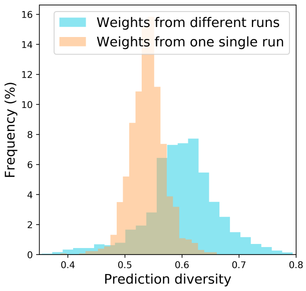


Figure 4.8. – Frequencies of diversities [Aks03] across 2 weights obtained along a single run or from different runs.

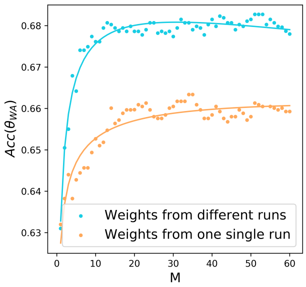


Figure 4.9. – WA accuracy as M increases, when the M weights are obtained along a single run or from different runs.

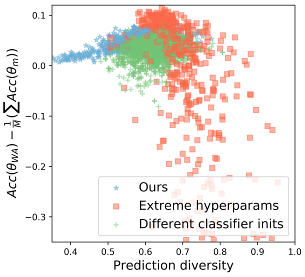


Figure 4.10. – Each dot displays the accuracy gain of WA over its members vs. diversity for $2 \leq M < 10$ models.

4.6.3.4 Failure under correlation shift

Our theory from Section 4.3 suggests that WA can tackle diversity shift (as previously verified) but not correlation shift; combining multiple models should be inefficient when bias dominates (see Section 2.2.2). We verify this failure on the ColoredMNIST [Arj+19] dataset, which is dominated by correlation shift [Pen+19]. Indeed, Colored MNIST is a variant of the MNIST handwritten digit classification dataset [LeC+10] where the correlation strengths between color and label vary across domains. We follow the DomainBed [Gul+21] protocol, with the small CNN architecture specialized for MNIST experiments, and the test-domain model selection argued in [Ye+22] (see the published paper [Ram+22b] for similar results with the train-domain model selection).

In Table 4.3, we observe that DiWA-uniform and MA both perform poorly compared to ERM, confirming the failure of WA strategies under correlation shift. Note that DiWA-greedy does not degrade ERM as it selects only a few models for averaging.

Table 4.3. – **Accuracy ($\%, \uparrow$) on ColoredMNIST**. WA does not improve performance under correlation shift. Random initialization of the classifier. Test-domain model selection.

	Algorithm	Weight selection	+90%	+80%	-90%	Avg
	ERM	OOD val	71.8 ± 0.4	72.9 ± 0.1	28.7 ± 0.5	57.8 ± 0.2
	Coral [Sun+16]	OOD val	71.1 ± 0.2	73.4 ± 0.2	31.1 ± 1.6	58.6 ± 0.5
	IRM [Arj+19]	OOD val	72.0 ± 0.1	72.5 ± 0.3	58.5 ± 3.3	67.7 ± 1.2
	Fishr [Ram+22a]	OOD val	**74.1** ± 0.6	73.3 ± 0.1	**58.9** ± 3.7	**68.8** ± 1.4
Our runs	ERM	N/A	71.5 ± 0.3	**74.1** ± 0.4	21.5 ± 1.9	55.7 ± 0.4
	MA [Arp+21]	Uniform	68.8 ± 0.2	72.1 ± 0.2	10.2 ± 0.0	50.4 ± 0.1
	ENS*	Uniform: $M = 20$	71.0 ± 0.2	72.9 ± 0.2	9.9 ± 0.0	51.3 ± 0.1
	DiWA	Greedy: $M \leq 20$	71.9 ± 0.4	73.6 ± 0.2	21.5 ± 1.9	55.7 ± 0.8
	DiWA	Uniform: $M = 20$	69.1 ± 0.8	72.6 ± 0.4	10.6 ± 0.1	50.8 ± 0.4
	DiWA†	Uniform: $M = 60$	69.3	72.3	10.3	50.6

4.6.4 DiWA with different objectives.

So far we used ERM without leveraging the domain information during training. Table 4.4 shows that DiWA-uniform benefits from weights fine-tuned with Interdomain Mixup [Yan+20a] and CORAL [Sun+16]: diversity increases and accuracy improves as we add more diverse objectives. This suggests a new kind of LMC across models fine-tuned with various objectives and losses; this idea will be further explored in rewarded soups [Ram+23b] detailed in Chapter 6.

Table 4.4. – **Accuracy (%, ↑) on OfficeHome** domain "Art" with various objectives.

Algorithm	No WA	MA	DiWA	DiWA†
ERM	62.9 ± 1.3	65.0 ± 0.2	67.3 ± 0.2	67.7
Mixup	63.1 ± 0.7	**66.2** ± 0.3	67.8 ± 0.6	68.4
CORAL	**64.4** ± 0.4	64.4 ± 0.4	67.7 ± 0.2	68.2
ERM/Mixup	N/A	N/A	67.9 ± 0.7	68.9
ERM/CORAL	N/A	N/A	68.1 ± 0.3	68.7
ERM/Mixup/CORAL	N/A	N/A	**68.4** ± 0.4	**69.1**

4.7 Conclusion

In this chapter, we highlighted the limitations of the previously dominant flatness-based analysis of WA. We proposed a novel bias-variance-diversity-locality analysis, leveraging the ensembling nature of WA, and underscoring a diversity-locality trade-off. This motivates our DiWA approach; by averaging the weights of independently trained models, DiWA improves performances on DomainBed. DiWA is a simple and practical strategy, with a straightforward implementation, making it a valuable tool for a broad range of real-world applications. Essentially, rather than selecting the best model from a hyperparameter search, it suggests that superior results can be obtained by averaging all the fine-tuned weights. Importantly, DiWA is without inference overhead, thereby removing a key limitation of standard ensembling.

Despite its advantages, certain challenges persist. Firstly, DiWA necessitates multiple independent training runs, hence does not curtail the training cost of ensembling. However, this issue might be considered minor as fine-tunings are typically faster than trainings from scratch. Secondly, as seen in Section 4.6.3.1, the shared initialization constraint limits diversity. Compared to functional ensembling that can combine arbitrary architectures, this may be problematic in contexts where multiple foundation models are available. Neuron permutations strategies [Ent+22; Ain+22] tried to enforce connectivity across weights fine-tuned from different initializations, though (so far) with moderate empirical results. Our subsequent Chapter 5 proposes an alternative to relax the shared initialization constraint.

Chapter 5

RATATOUILLE: RECYCLING DIVERSE MODELS FOR OUT-OF-DISTRIBUTION GENERALIZATION

5.1 Introduction

Learning robust models that generalize well is critical for many real-world applications [Zec+18; DeG+21]. Fine-tuning pre-trained foundation models [Bom+21] is now the most popular approach to build deep learning (DL) solutions for these applications. For fine-tuning, empirical risk minimization (ERM) [Vap92] has long remained the best strategy on the reference DomainBed [Gul+21] benchmark. Yet, as previously introduced in Chapter 4, the ability to average neural networks' weights inspired a plethora of modern weight averaging (WA) approaches for fine-tuning. We illustrate some of them in Figure 5.1, such as MA [Izm+18], WiSE fine-tuning [Wor+22b], and our DiWA [Ram+22b].

Fostered by the open-source philosophy in ML, the Internet is then swarmed by a handful of foundation models fine-tuned on many diverse tasks: these individual fine-tunings are available on public repositories such as `torchvision` [Mar+10], `huggingface` [Wol+20], or `timm` [Wig19], yet exist in isolation without benefiting from each other. In our opinion, this is a missed opportunity, as these specialized models contain *rich and diverse* features. Recent inter-training [Pha+18; Pru+20] and fusing [Cho+22b; Don+23a] strategies recycle intermediate fine-tunings on auxiliary tasks to enrich the features before fine-tuning on the target task. However, the success of these recycling strategies usually depend on the similarity between the auxiliary and target tasks. Moreover, as argued in Section 5.2, these strategies fail to fully leverage the diversity in auxiliary tasks.

Thus, the central question of this chapter is:

How can we best recycle diverse fine-tunings of a given foundation model towards strong out-of-distribution performance on our target task?

Our answer is *model ratatouille*[1], a simple fine-tuning strategy illustrated in Figure 5.1 and described in Section 5.3. In a similar fashion to converting waste into reusable material for new uses, we take fine-tunings of the same foundation model on diverse auxiliary tasks and repurpose them as initializations to start multiple fine-tunings on the target task. Specifically, we (i) fine-tune a copy of the foundation model on each auxiliary task,

[1] We named our method after this traditional French dish for two main reasons. Firstly, the ratatouille is often used as a way to recycle leftover vegetables. Secondly, the ratatouille is better prepared by cooking each ingredient separately before mixing them: this technique ensures that each ingredient "will taste truly of itself", as noted by chef Joël Robuchon [Mon20].

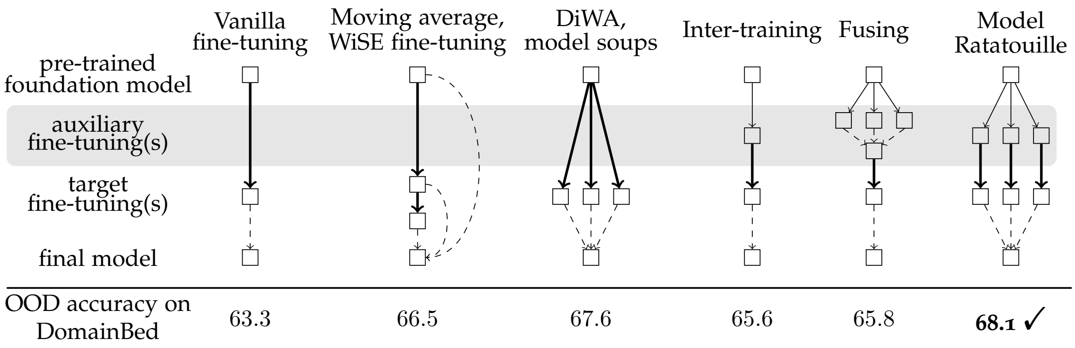


Figure 5.1. – The different fine-tuning strategies discussed in this chapter: vanilla fine-tuning [Oqu+14], moving average (MA) [Izm+18] and variants [Wor+22b], DiWA [Ram+22b] introduced in Chapter 4 and the similar model soups [Wor+22a], inter-training [Pha+18], fusing [Cho+22b] and our novel *model ratatouille*. They start with a pre-trained foundation model. Some strategies fine-tune the pre-trained model on auxiliary tasks (thin solid arrows ⟶): these auxiliary fine-tunings can be performed by different contributors of the community on their own data. Then, all strategies perform fine-tuning on the target task of interest (thick solid arrows ⟶). Finally, the weights fine-tuned on the target task are used as is, or are averaged (dashed arrows ⇢) into a final model. Ratatouille (i) enables compute parallelism, (ii) maximizes the amount of diversity in models' predictions, (iii) achieves sota performance in DomainBed [Gul+21], the standard computer vision benchmark for OOD generalization and (iv) does not incur any inference or training overhead compared to a traditional hyperparameter search.

(ii) fine-tune each auxiliary model on the target task, and (iii) return as the final model the average of all target fine-tuned weights. In brief, while DiWA and model soups average multiple weights fine-tuned from a shared initialization, model ratatouille averages multiple weights fine-tuned from different initializations each inter-trained [Pha+18] on different auxiliary tasks. As we will see, ratatouille works because the fine-tunings remain linearly mode connected (LMC) [Fra+20; Mir+21] in the loss landscape (despite having different initializations) and thus their average improves performance.

We show the efficacy of model ratatouille in Section 5.4, where we set a new sota on DomainBed [Gul+21]. We will show how we leverage the diversity across the auxiliary tasks to construct diverse weights, that can be averaged into a final model. Looking forward, as we discuss in Section 7.2.3, this chapter contributes to the emerging paradigm of *updatable machine learning* [Raf23], where the community collaborates towards incrementally and reliably updating the capabilities of a ML model.

This chapter led to the publication of: Alexandre Ramé, Kartik Ahuja, Jianyu Zhang, Matthieu Cord, Léon Bottou, and David Lopez-Paz. "Model Ratatouille: Recycling Diverse Models for Out-of-Distribution Generalization". In: *ICML*. 2023.

5.2 Context

We start by describing our setup. We train a deep model f, where the featurizer is parametrized by the weights ϕ, the classifier is parametrized by the weights ω, and thus

f is parametrized by the concatenation weights $\theta = (\omega, \phi)$. We are dealing with out-of-distribution (OOD) generalization, and our aim is to find θ maximizing the test accuracy $\mathrm{acc}_T(\theta)$. In this chapter, we only consider diversity shifts [Ye+22].

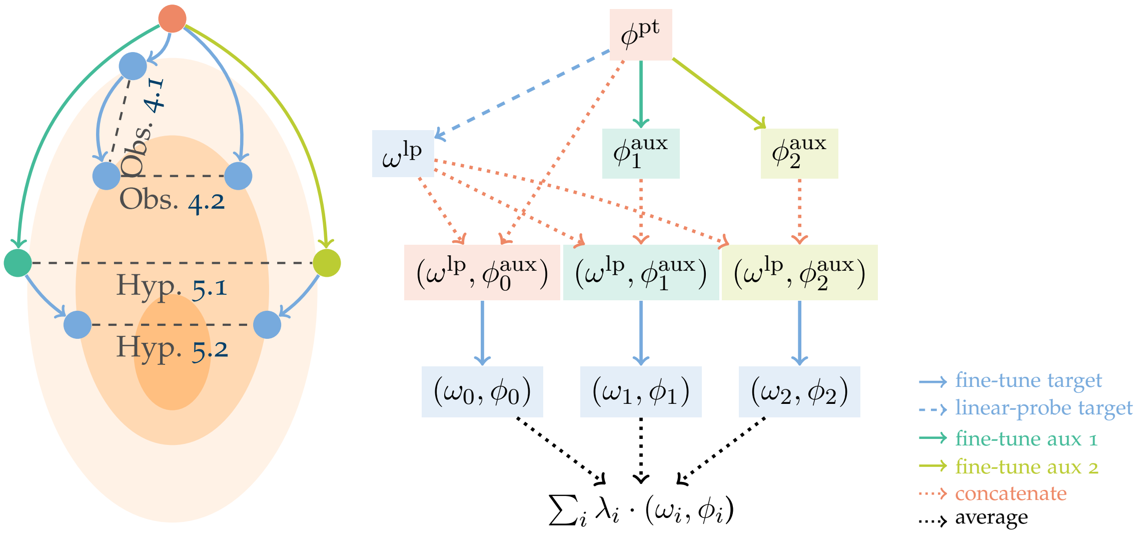
5.2.0.1 ERM and WA

The recipe in transfer learning [Oqu+14; Kir+22] is (i) download a pre-trained featurizer with parameters ϕ^{pt}, (ii) plug a classifier ω^{lp} compatible with the target task, and (iii) fine-tune the network with empirical risk minimization (ERM) [Vap92] on the target task. While the classifier ω^{lp} could be initialized at random, linear probing (LP) (*i.e.,* first learning only the classifier with frozen featurizer) improves results by preventing feature distortion [Kum+22]. Based on Observation 4.1, moving average (MA) strategies and variants [Sze+16; Izm+18; Dra+18; Arp+21; Cha+21a; Wor+22b] average the weights of checkpoints collected every few epochs to build the final weights. More recently, as explained in previous Chapter 4 and consistently with Observation 4.2 from [Ney+20], our DiWA [Ram+22b] proposed to average all the weights obtained from a standard ERM hyperparameter search. DiWA and the similar model soups [Wor+22a] are so far the best approaches for OOD generalization. However, as highlighted in Section 4.7, the shared initialization constraint in DiWA limits models diversity [Kun+03; Aks03]; removing this constraint would be desirable to further reduce variance.

5.2.0.2 Weight averaging over tasks

All the methods described so far fine-tune only on the target task: could auxiliary datasets, increasingly available online, be incorporated into the learning process to learn richer features? Such tasks could be an opportunity to recruit specialized features [Li+21] that match our target task, ease optimization [Zha+22; Zha+23a], or "offer some high-level guidance to bridge the gaps between the pre-training and fine-tuning phases" [Cha+21b]. Following these ideas, *inter-training* approaches [Pha+18; Pru+20; Cho+22a] perform an intermediate fine-tuning of the pre-trained model on some auxiliary task, before tackling the target task. However, the sequential nature of inter-training leads to catastrophic forgetting [Reb+17] of useful knowledge contained in the original pre-trained model. Moreover, the choice of the auxiliary task plays a determinant role, since "when the wrong task is chosen, inter-training hurts results" [Cho+22b]. To address the shortcomings of inter-training, recent works [Cho+22b; Don+23a; Li+22; Mat+22; Ilh+23; Ilh+22] recycle weights fine-tuned on various auxiliary tasks. In particular, concurrent [Cho+22b] operates *fusing* at initialization; they (i) fine-tune one copy of the pre-trained model on each auxiliary task, (ii) average the auxiliary fine-tuning weights, and (iii) use such averaged model as the initialization for the target fine-tuning. By weight interpolation, fusing combines into one single initialization the knowledge from multiple auxiliary tasks; yet fusing provides only marginal empirical gains in Section 5.4.1 on DomainBed.

We posit that model fusing is performing weight averaging prematurely, destroying most diversity from auxiliary tasks even before the target task can benefit from it. To



(a) LMC conditions (b) Diagram of model ratatouille. (c) Arrow legend

Figure 5.2. – Illustrations of (a) different linear mode connectivity (LMC) conditions, and (b) model ratatouille. In subplot (a), we illustrate Observation 4.1, about LMC between two checkpoints along the same target fine-tuning; Observation 4.2, about LMC between two target fine-tunings; Hypothesis 5.1, about LMC between two auxiliary fine-tunings; and Hypothesis 5.2, about LMC between two target fine-tunings initialized from auxiliary weights satisfying Hypothesis 5.1. In subplot (b), we offer a diagram of our proposed ratatouille strategy, where we (i) fine-tune a pre-trained model on auxiliary tasks, (ii) plug a linear probe classifier on the pre-trained model and the auxiliary fine-tunings, (iii) fine-tune on the target task from each auxiliary weights, and (iv) return their weight average as the final model.

address this, next we propose *ratatouille*, a new recycling strategy that performs one target fine-tuning per auxiliary weights, and averages weights only as the very last step.

5.3 Model ratatouille

5.3.1 Recycling diverse initializations

Our model ratatouille is a proposal to recycle diverse auxiliary fine-tunings of the same pre-trained model; it is compared against other fine-tuning strategies in Figure 5.1 and outlined in detail in Figure 5.2(b). Ratatouille recycles these fine-tunings as diverse initializations to parallel fine-tunings on the target task. Compared to fusing [Cho+22b], we delay the weight averaging, and in turn the destruction of diversity. Ratatouille follows this five-step recipe.

1. Download a featurizer ϕ^{pt} pre-trained on task D_0.
2. Fine-tune ϕ^{pt} on each auxiliary task D_i, obtaining $(\omega_i^{aux}, \phi_i^{aux})$ for $i = 0, \ldots, M-1$.
3. Replace each ω_i^{aux} by ω^{lp}, obtained by linear probing the original pre-trained model ϕ^{pt} on the target task D.

4. Fine-tune each $(\omega^{\text{lp}}, \phi_i^{\text{aux}})$ on the target task D, obtaining $\theta_i = (\omega_i, \phi_i)$ for $i = 0, \ldots, M-1$.

5. Return as final model $\sum_{i=0}^{M-1} \lambda_i \cdot \theta_i$. To select the interpolating coefficients, we use the same two weight selection strategies previously described in Section 4.4.4. The first *uniform* averages all weights with $\lambda_i = \frac{1}{M}$. The second *greedy* sorts the θ_i by descending ID validation accuracy, before greedily constructing an uniform average containing θ_i if and only if its addition lowers the ID validation accuracy.

If the weights from step 2 are made available online, ratatouille is without any training overhead compared to a traditional hyperparameter search. When compared to inter-training [Pha+18] and fusing [Cho+22b], model ratatouille avoids the difficult choice of choosing one single initialization [Cho+22a]. The shared LP classifier facilitates LMC by preventing feature distortions [Kum+22]. Note that we consider the pre-training task as the auxiliary task "number zero" D_0; this resembles WiSE fine-tuning [Wor+22b] and aims at preserving the general-purpose knowledge contained in the original pre-trained model.

In essence, auxiliary tasks helps in two ways: through their similarity with the target task, and their diversity. As described in previous Section 4.3, successful weight averaging (WA) requires three conditions [Ram+22b]. The weights should be:

1. individually accurate (to reduce the bias); by inter-training, ratatouille enriches the features and thus increases individual accuracies when the auxiliary tasks are well-chosen [Cho+22a].

2. sufficiently diverse (to reduce variance): by removing the shared initialization constraint from DiWA, ratatouille benefits from the additional diversity brought by specialization on various auxiliary tasks.

3. averageable; for ratatouille to work, it requires a relaxation of the conditions under which the LMC holds, that we detail below.

5.3.2 Novel linear mode connectivity hypotheses

We now extend Observations 4.1 and 4.2 made in Chapter 4 when including fine-tunings on auxiliary tasks. First, we introduce Hypothesis 5.1 that posits LMC between two models whose featurizers were fine-tuned on different auxiliary tasks.

Hypothesis 5.1 (LMC with different tasks). *The LMC holds between* $\left(\omega^{\text{lp}}, \phi_1^{\text{aux}}\right)$ *and* $\left(\omega^{\text{lp}}, \phi_2^{\text{aux}}\right)$ *if* ϕ_1^{aux} *and* ϕ_2^{aux} *are featurizers fine-tuned on two auxiliary tasks initialized from the same pre-trained featurizer* ϕ^{pt}. *Here,* ω^{lp} *is the linear probe of* ϕ^{pt} *on the target task.*

Though this Hypothesis 5.1 was never formulated explicitly, it is underlying in fusing [Cho+22b] and in other works averaging auxiliary weights. Actually, ratatouille relies on the following Hypothesis 5.2, which enriches Hypothesis 5.1 with additional independent fine-tunings on the target task.

Hypothesis 5.2 (LMC with different auxiliary initializations)**.** *The LMC holds between θ_1 and θ_2 fine-tuned on the target task starting from initializations $\left(\omega^{\mathrm{lp}}, \phi_1^{\mathrm{aux}}\right)$ and $\left(\omega^{\mathrm{lp}}, \phi_2^{\mathrm{aux}}\right)$ satisfying Hypothesis 5.1.*

Hypothesis 5.2 is the first to posit the LMC between weights fine-tuned from different initializations. It hints towards a more general inheritance property: *if two initializations satisfy LMC, then the two final weights would too*. We expect Hypotheses 5.1 and 5.2 to hold as long as the pre-training, auxiliary and target tasks are sufficiently similar, and if hyperparameters remain in a mild range. If these LMC hold, then we expect ratatouille to improve generalization abilities. But this, we can only answer empirically through proper experimentation in the upcoming Section 5.4.

5.4 Experiments

Our numerical experiments support five main claims, sorted in decreased granularity. First, Section 5.4.1 showcases the sota results of ratatouille in DomainBed [Gul+21]. Second, Section 5.4.2 illustrates how such gains arise from increased diversity across averaged models. Third, Section 5.4.3 empirically supports Hypotheses 5.1 and 5.2, the LMC conditions enabling weight averaging's success. Then, Section 5.4.4 shows that additional auxiliary tasks improve results. Finally, Section 5.4.5 discusses the impact of ratatouille for in-domain tasks.

5.4.1 Sota performance on DomainBed

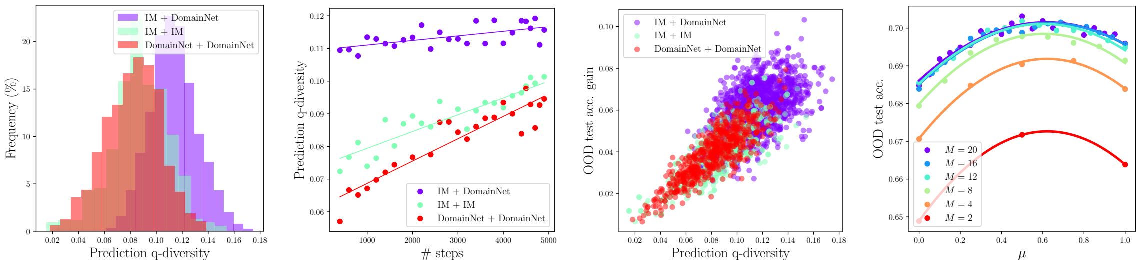
Setup. Table 5.1 shows our main experiment comparing the various fine-tuning strategies on DomainBed [Gul+21]. We follow the same procedure previously described in Section 4.6.1. Our goal is to improve the performances previously obtained by DiWA [Ram+22b], a.k.a. model soups [Wor+22a]. As a reminder, DiWA only differs from ERM vanilla fine-tuning by the selection strategy: rather than selecting the model with highest ID validation accuracy out of the 20 runs, DiWA either uniformly averages all weights or greedily selects some. The key originality of ratatouille is to leverage auxiliary trainings; in practice, given a target dataset, we consider the other DomainBed's datasets as the auxiliary tasks. For example when tackling OfficeHome, out of the 20 runs, 4 are inter-trained on PACS, 4 on VLCS, 4 on TerraIncognita, 4 on DomainNet and 4 are directly transferred from ImageNet. Then, *model ratatouille is to inter-training as DiWA is to vanilla fine-tuning*. As in previous Section 4.6.1, the "†" symbol marks methods averaging $60 = 20 \times 3$ weights from 3 data splits. We further discuss ratatouille's training cost in [Ram+23a]. The procedure to obtain the pool of initializations is agnostic to the target task or the test domain, and thus is done only once; in particular, we argue that ratatouille is without training overhead when auxiliary weights are shared by the community. Functional ensembling strategies (marked by the symbol "$*$") average predictions with large inference overhead: for example, "ENS* of inter-training" averages the predictions of the $M = 20$ models

Table 5.1. – **Accuracies (%, ↑) on DomainBed** [Gul+21] benchmark evaluating OOD generalization. Ratatouille sets a new sota by leveraging diversity in auxiliary tasks. The symbol "$*$" indicates inference overhead in functional ensembling. The symbol "†" indicates the averaging of all weights across 3 data splits. The scores for DiWA are those from Table 4.1 with LP initialization.

	Algorithm	**Weight selection**	**PACS**	**VLCS**	**OfficeHome**	**TerraInc**	**DomainNet**	**Avg**
	Vanilla fine-tuning	ID val	85.5 ± 0.2	77.5 ± 0.4	66.5 ± 0.3	46.1 ± 1.8	40.9 ± 0.1	63.3
	CORAL [Sun+16]	ID val	86.2 ± 0.3	78.8 ± 0.6	68.7 ± 0.3	47.6 ± 1.0	41.5 ± 0.1	64.6
	SWAD [Cha+21a]	Loss-aware	88.1 ± 0.1	**79.1** ± 0.1	70.6 ± 0.2	50.0 ± 0.3	46.5 ± 0.1	66.9
	MA [Arp+21]	Uniform	87.5 ± 0.2	78.2 ± 0.2	70.6 ± 0.1	50.3 ± 0.5	46.0 ± 0.1	66.5
	DE* [Arp+21]	Uniform: $M = 6$	87.6	78.5	70.8	49.2	**47.7**	66.8
Our runs	Vanilla fine-tuning	ID val	85.9 ± 0.6	78.1 ± 0.5	69.4 ± 0.2	50.4 ± 1.8	44.3 ± 0.2	65.6
	ENS*	Uniform: $M = 20$	88.1 ± 0.3	78.5 ± 0.1	71.7 ± 0.1	50.8 ± 0.5	47.0 ± 0.2	67.2
	DiWA	Uniform: $M = 20$	88.7 ± 0.2	78.4 ± 0.2	72.1 ± 0.2	51.4 ± 0.6	47.4 ± 0.2	67.6
	DiWA	Greedy: $M \leq 20$	88.0 ± 0.3	78.5 ± 0.1	71.5 ± 0.2	51.6 ± 0.9	**47.7** ± 0.1	67.5
	DiWA†	Uniform†: $M = 60$	89.0	78.6	72.8	$\underline{51.9}$	**47.7**	68.0
	Inter-training [Pha+18]	ID val	89.0 ± 0.0	77.7 ± 0.0	69.9 ± 0.6	46.7 ± 0.1	44.5 ± 0.1	65.6
	ENS* of inter-training	Uniform: $M = 20$	89.2 ± 0.1	$\underline{79.0} \pm 0.2$	72.7 ± 0.1	51.1 ± 0.3	47.2 ± 0.1	67.8
	Fusing [Cho+22b]	ID val	88.0 ± 1.0	78.5 ± 0.8	71.5 ± 0.5	46.7 ± 1.8	44.4 ± 0.2	65.8
	Ratatouille	Uniform: $M = 20$	89.5 ± 0.1	78.5 ± 0.1	73.1 ± 0.1	51.8 ± 0.4	47.5 ± 0.1	$\underline{68.1}$
	Ratatouille	Greedy: $M \leq 20$	**90.5** ± 0.2	78.7 ± 0.2	$\underline{73.4} \pm 0.3$	49.2 ± 0.9	**47.7** ± 0.0	67.9
	Ratatouille†	Uniform†: $M = 60$	$\underline{89.8}$	78.3	**73.5**	**52.0**	**47.7**	**68.3**

ratatouille-uniform averages in weights. For fusing [Cho+22b], each run is initialized from $\sum_{i=0}^{4} \lambda_i \phi_i^{\text{aux}}$ where the λ_i hyperparameters sum to 1 and ϕ_i^{aux} are inter-trained on one the 4 other DomainBed's datasets or directly transferred from ImageNet.

Results. Table 5.1 shows that ratatouille achieves a new sota on DomainBed: with uniform selection, it achieves 68.1 and improves DiWA by 0.5 points after averaging over all datasets. Precisely, model ratatouille beats DiWA by 0.8 and 1.0 points on PACS and OfficeHome with uniform selection, and by 2.5 and 1.9 with greedy selection. On these two datasets, inter-training and fusing also succeed, yet they fail on TerraIncognita (both reach 46.7%) as all auxiliary tasks are distant from photos of animals in the wild; in contrast on TerraIncognita, ratatouille-uniform (51.8%) matches DiWA-uniform (51.4%). This highlights the key strength of our ratatouille w.r.t. other recycling strategies such as fusing: namely, *the robustness to the choice of auxiliary tasks*. On VLCS, ratatouille is also generally beneficial, except on one domain where the LMC breaks (as shown in [Ram+23a]). For DomainNet, ratatouille is sota though the gains are small w.r.t. DiWA: we suspect this is because the initialization strategy becomes less critical for larger datasets [Cha+21b] with more training epochs (see Figure 5.3(b)). In conclusion, ratatouille consistently improves generalization on DomainBed, and works best with appropriate auxiliary tasks: we remove the need to select only the *best* initialization. This is similar to DiWA, which works best with appropriate hyperparameter ranges; DiWA removed the need to select only the best set of hyperparameters.



(a) Diversity frequency.

(b) Diversity vs. fine-tuning steps.

(c) Acc. gain vs. diversity.

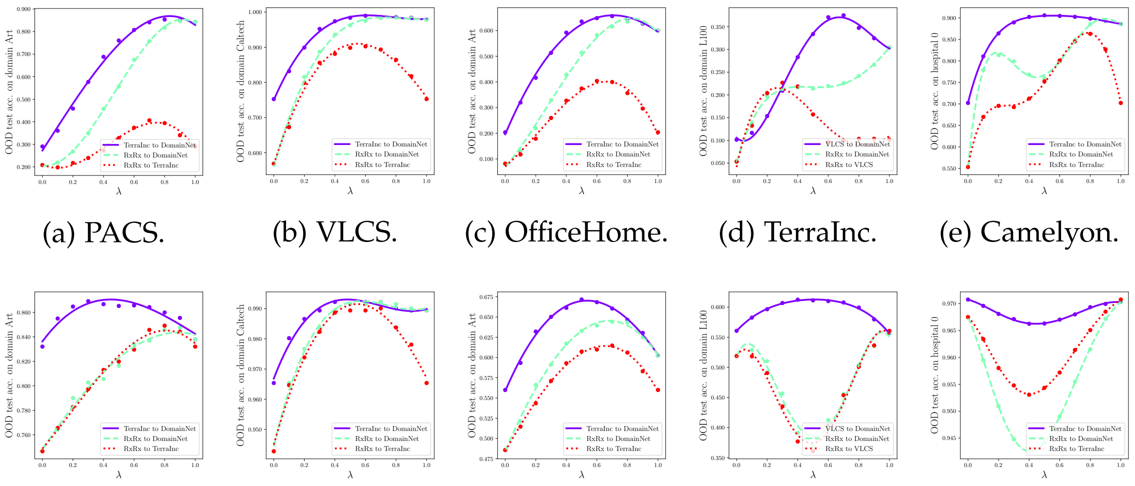
(d) Acc. vs. mixing ratio.

Figure 5.3. – **Explorations on Q-diversity** [Kun+03] and its positive impact on accuracy for the OOD test domain "Art" from OfficeHome. In (a), we compute the diversity between pairs of models either directly fine-tuned from ImageNet, either inter-trained on DomainNet: having one model from each initialization increases diversity. In (b), we plot this diversity along the 5k training steps. In (c), we observe that the more diverse the models, the higher the accuracy gain of their weight average compared to the average of their individual accuracies. In (d), we average M models: a proportion $(1 - \mu)$ start directly from ImageNet, the others μ are inter-trained on DomainNet. The accuracy of the weight average is maximized when $\mu \approx 0.5$.

5.4.2 Increased diversity by recycling

In Figure 5.3, we investigate how the diversity across models fine-tuned on the target task influences the OOD performance of their WA. Here, we measure diversity with the prediction Q-diversity [Kun+03], previously introduced in Section 2.3.3.1, which increases when models fail on different examples; in the paper, we also arrive at similar conclusions using the ratio-error [Aks03]. We follow the same protocol as in Section 4.6.3, and consider OfficeHome as the target task, with "Art" as the test OOD domain; we thus train on the "ClipArt", "Product" and "Photo" domains. We consider models either only pre-trained on ImageNet or also inter-trained on DomainNet.

First, we verify that inter-training influences the diversity across fine-tuned models. Specifically, Figure 5.3(a) confirms that networks with different initializations are more diverse than networks initialized similarly. Then, Figure 5.3(b) verifies that this diversity gain comes from their initialization and remains along fine-tuning on the target task. Moreover, Figure 5.3(c) shows that diversity is positively linearly correlated with OOD generalization: having different initializations improves diversity and thus WA accuracy. Finally, in Figure 5.3(d), we consider averaging M weights: a proportion $(1 - \mu)$ start directly from ImageNet, the others μ were inter-trained on DomainNet. In the simplest case $M = 2$, using one model from each initialization leads to maximum accuracy; for larger M, best performances are obtained around $\mu \approx 0.5$, where the final WA has access to diverse initializations. In conclusion, each auxiliary task fosters the learning of diverse features [Li+21; Gon+22]. Model ratatouille increases diversity and improves performance by removing a key limitation of model soups [Wor+22a] and DiWA [Ram+22b]; the need for all fine-tunings to start from a shared initialization.



(a) PACS. (b) VLCS. (c) OfficeHome. (d) TerraInc. (e) Camelyon.

(f) PACS. (g) VLCS. (h) OfficeHome. (i) TerraInc. (j) Camelyon.

Figure 5.4. – Figures 5.4(a) to 5.4(e) validate Hypothesis 5.1 by plotting $\lambda \to \text{acc}_T\left(\left(w^{\text{lp}}, (1-\lambda)\cdot\phi_1^{\text{aux}} + \lambda\cdot\phi_2^{\text{aux}}\right)\right)$, where w^{lp} is the linear probe of $\phi_{\text{IM}}^{\text{pt}}$, and ϕ_1^{aux} and ϕ_2^{aux} are fine-tuned on the two auxiliary datasets in the legend "Dataset$_1$ to Dataset$_2$". Figures 5.4(f) to 5.4(j) support Hypothesis 5.2 by plotting $\lambda \to \text{acc}_T\left((1-\lambda)\cdot\theta_1 + \lambda\cdot\theta_2\right)$ where θ_1 and θ_2 are fine-tuned on the target task starting respectively from $(w^{\text{lp}}, \phi_1^{\text{aux}})$ and $(w^{\text{lp}}, \phi_2^{\text{aux}})$. We encounter two exceptions to Hypothesis 5.2 (Figures 5.4(i) and 5.4(j)), due to the fact that *neither* the auxiliary (RxRx) *nor* the target task (TerraIncognita and Cameyon) bear enough similarity with the pre-training task (ImageNet).

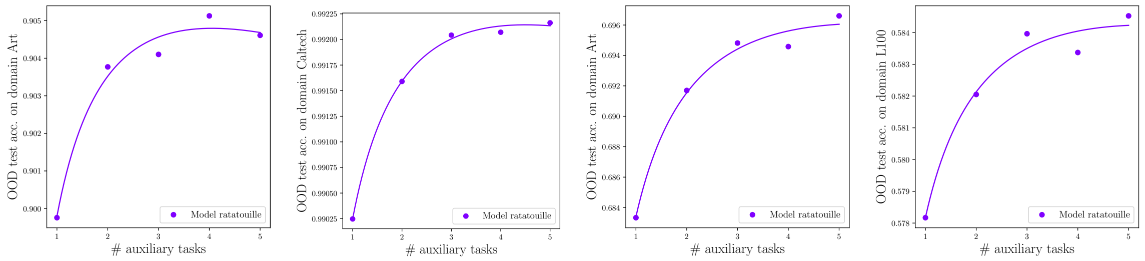
5.4.3 Why ratatouille works

In Figure 5.4, we continue our experiments by validating Hypotheses 5.1 and 5.2 when considering the five datasets from DomainBed. For the sake of completeness, we also analyze some successes and failure cases in *extreme* conditions when considering two distant unrelated medical datasets; RxRx [Tay+19] and Camelyon [Koh+21] from the WILDS [Koh+21] benchmark. For each target task, we consider the first domain as the test OOD; the other domains are used for training.

We validate Hypothesis 5.1 in Figures 5.4(a) to 5.4(e). For each dataset, we plot the test OOD accuracy for the weights $\left(w^{\text{lp}}, (1-\lambda)\cdot\phi_1^{\text{aux}} + \lambda\cdot\phi_2^{\text{aux}}\right)$, where the classifier w^{lp} is a linear probe of the ImageNet pre-trained featurizer $\phi_{\text{IM}}^{\text{pt}}$, and $\lambda \in [0,1]$ interpolates between ϕ_1^{aux} and ϕ_2^{aux}, obtained by fine-tuning on two auxiliary tasks initialized from $\phi_{\text{IM}}^{\text{pt}}$. First, we observe that task similarity influences OOD generalization since the test accuracies in Figure 5.4(c) agree with the fact that OfficeHome is most similar to DomainNet, not as similar to TerraIncognita, and most dissimilar to the medical dataset RxRx. Second, *the accuracy of the interpolated weights is above the interpolated accuracy*: this validates Hypothesis 5.1. The accuracy is even usually concave in λ.

Similarly, we empirically support Hypothesis 5.2 in Figures 5.4(f) to 5.4(j). For each dataset, we plot the test OOD accuracy obtained with weights $(1-\lambda)\cdot\theta_1 + \lambda\cdot\theta_2$, where the coefficient $\lambda \in [0,1]$ interpolates between θ_1 and θ_2, fine-tuned on the target task respectively starting from $(w^{\text{lp}}, \phi_1^{\text{aux}})$ and $(w^{\text{lp}}, \phi_2^{\text{aux}})$. We observe that Hypothesis 5.2 usually holds: for example, even recycling RxRx can help for OfficeHome on Figure 5.4(h). Yet, Hypothesis 5.2 breaks on TerraIncognita and Camelyon in Figures 5.4(i) and 5.4(j) when RxRx is one of the two auxiliary tasks. In light of these results, we argue *that Hypothesis 5.2 holds as long as either the auxiliary or the target task is sufficiently similar to the pre-training task*. We speculate this prevents feature distortion [Kum+22] and escaping a shared loss valley. Better understanding when LMC breaks is a promising research direction [Jun+23; Lub+22]; among other factors, we speculate that larger pre-training corpus (as in [Qin+22]) or larger architectures (as in [Li+22]) may favor WA strategies.

5.4.4 Analysis of the number of auxiliary tasks



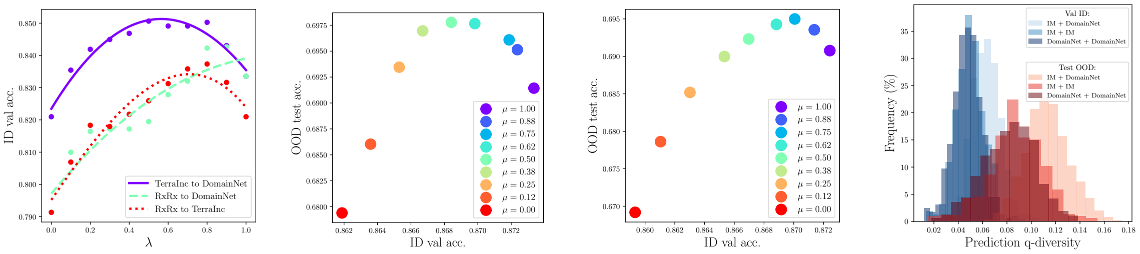
(a) PACS. (b) VLCS. (c) OfficeHome. (d) TerraIncognita.

Figure 5.5. – OOD accuracy (↑) for model ratatouille when increasing the number of auxiliary tasks and uniformly averaging all fine-tuned weights. For each target task, we consider the first domain as the test OOD; the other domains are used for training.

In Table 5.1, ratatouille leverages 5 auxiliary tasks for simplicity: ImageNet (which we consider as the auxiliary task "number zero"), and the 4 other datasets from DomainBed (out of the 5, as we leave out the target task to prevent any information leakage). In following Figure 5.5, we report the scores obtained using 1 to 5 auxiliary tasks: we always average $M = 20$ weights, the only difference is how they were initialized. When we have 1 auxiliary task, they were all inter-trained on this auxiliary task: when we have 2 auxiliary tasks, 10 are inter-trained on the first auxiliary task, 10 on the second *etc*. This validates that a greater number of auxiliary tasks leads to an increase in expected OOD accuracy. We expect that further increasing the number of auxiliary datasets—beyond those from DomainBed—would further improve results.

5.4.5 Ratatouille for ID tasks

Like previous WA strategies [Izm+18; Wor+22a], model ratatouille also works for in-distribution (ID) tasks; in particular, we verify in Figure 5.6(a) that the LMC holds on the ID validation samples, following the same distribution as the training samples. Yet, the gains are smaller in ID than in OOD, as confirmed by the lack of correlation between



(a) LMC in ID. (b) OOD vs. ID acc. (c) OOD vs. ID acc. (d) Diversity.

Figure 5.6. – The models were trained on ID domains “Clipart”, “Product”, and “Photo” from OfficeHome, thus “Art” is the OOD domain. First, in subplot (a), we validate Hypothesis 5.2 on the ID validation split. Then, we analyze the relations between diversity, ID and OOD accuracies. In subplot (b), we report the mean results when averaging $M = 8$ weights: $(1 - \mu)$ are fine-tuned on OfficeHome directly from ImageNet, the others μ are inter-trained on DomainNet. We observe a lack of correlation between ID and OOD accuracies. We observe a similar trend in subplot (c), which mirrors the experiment from subplot (b) with the only difference that the proportion $(1 - \mu)$ are inter-trained on PACS (rather than just transferred from ImageNet). In subplot (d), we compute the diversity [Kun+03] between models either directly fine-tuned from ImageNet, either inter-trained on DomainNet. Though having different initializations increases diversity both in ID and in OOD, the diversity in ID remains smaller.

ID and OOD accuracies [Ten+22] in Figures 5.6(b) and 5.6(c). This is explained by the fact that variance reduction from WA is less beneficial in ID than in OOD. Theoretically, this is because, variance is smaller without distribution shift, as proved in Section 2.2.1. Empirically, this is consistent with models’ diversity being smaller in ID, as shown in Figure 5.6(d). Overall, *diversity procedures are less useful in ID than in OOD*. Ratatouille performs well OOD thanks to the diversity brought by diverse inter-trainings; for ID, we should sacrifice some diversity and select one single optimal initialization. This also explains occasional failures of the greedy selection (notably for TerraIncognita in Table 5.1): based on the ID validation accuracy, only a few runs are selected and averaged, causing smaller OOD accuracy than with the uniform selection.

5.5 Conclusion

This chapter introduces model ratatouille, broadening the foundation model paradigm by recycling weights fine-tuned on a variety of auxiliary tasks. Ratatouille extends DiWA by relaxing the shared initialization constraint; the fine-tunings only need to start from initializations that are themselves linearly mode connected. This relaxation allows to average a more diverse set of weights, which in turn leads to better OOD generalization.

The main limitation of this chapter is that we explore OOD generalization in supervised learning scenarios, where objectives are well-defined and ground-truth labels exist. This is not always the case in real-world applications, where defining desired behaviours are not trivial. In the next Chapter 6, we will investigate whether WA can be beneficial for RL tasks, specifically in the context of RLHF with diverse rewards.

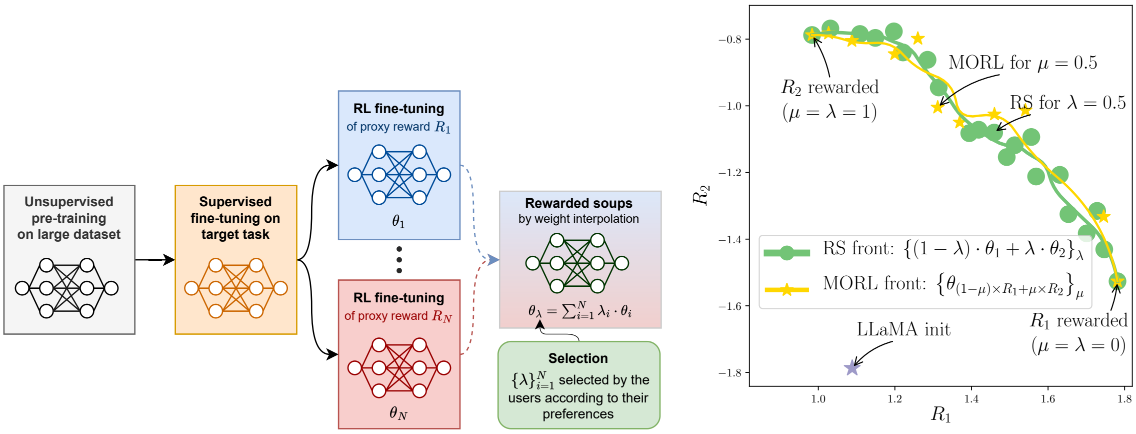
Chapter 6

REWARDED SOUPS: TOWARDS PARETO-OPTIMAL ALIGNMENT BY INTERPOLATING WEIGHTS FINE-TUNED ON DIVERSE REWARDS

6.1 Introduction

Foundation models [Bom+21] are usually pre-trained through self-supervision [Dev+19; Bro+20; Car+21; Rad+21] and then fine-tuned [Oqu+14; Yos+14] via supervised learning [Vap99]. This was the main scenario in previous chapters, where classification labels were properly defined on the target task. Yet, collecting supervised labels is often expensive, and not always possible for more subjective tasks with multiple correct answers or involving human concepts [Kwo+23] such as helpfulness [Bai+22a; Ask+21]; thus supervision may not cover all possibilities, and fail to perfectly align [Amo+16; Tay+16; Ngo+22] the network with the intended applications. Recent works [Sti+20b; Ouy+22; Pin+23] showed that deep reinforcement learning (DRL) helps by learning from various types of rewards. A prominent example is reinforcement learning from human feedback (RLHF) [Sti+20b; Chr+17b; Zie+19; Wu+21], which appears as the current go-to strategy to refine large language models (LLMs) into powerful conversational agents such as ChatGPT [Ouy+22; Ope23]. After pre-training on next token prediction [Rad+18] using Web data, the LLMs are fine-tuned to follow instructions [Wei+22a; Wan+22c; Tao+23] before reward maximization. This third RL step enhances alignment by evaluating the entire generated sentence instead of each token independently [Gol23]. Similar strategies have been useful in computer vision (CV) [Pin+23; Ren+17], for instance to integrate human aesthetics into image generation [Lee+23; Wu+23a; Zha+23b].

Diversity of proxy rewards. RL is usually seen as more challenging than supervised training [Dul+21], notably because the real reward—reflecting the users' *true* preferences—is often not specified at training time. Proxy rewards are therefore developed to guide the learning, either as hand-engineered metrics [Pap+02; Lin+03; Ved+15] or more recently in RLHF as models trained to reflect human preferences [Chr+17b; Kwo+23; Xu+23]. Nonetheless, designing reliable proxy rewards for evaluation is difficult. This *reward misspecification* [Amo+16; Pan+22] between the proxy reward and the users' actual rewards can lead to unforeseen consequences [Mic+20]. Moreover, the diversity of objectives in real-world applications exacerbates the issue. In particular, human opinions can vary significantly [Wil87; Coe00; Sch+12] on subjects such as aesthetics [Nad+19], politics or fairness [Lop+22]. Humans have also different expectations from machines: for example, while [Gan+22] stressed aligning LLMs towards harmless feedback, [Bai+22b] requested



(a) Our proposed rewarded soup.

(b) LLaMA RLHF.

Figure 6.1. – Figure 6.1(a) details the different steps in rewarded soup. After unsupervised pre-training and supervised fine-tuning, we launch N independent RL fine-tunings on the proxy rewards $\{R_i\}_{i=1}^N$. Then we combine the trained networks by interpolation in the weight space. The final weights are adapted at test time by selecting the coefficient λ. Figure 6.1(b) shows our results (extended in Figure 6.2(a)) with LLaMA-7b [Tou+23a] instruct fine-tuned on Alpaca [Tao+23], when RL fine-tuning for news summarization [Sti+20b] with $N = 2$ reward models assessing diverse preferences: R_1 rewards completeness while R_2 rewards faithfulness. With only two trainings (R_1 and R_2 rewarded on Figure 6.1(b)), the λ-interpolation ($0 \leq \lambda \leq 1$) reveals the green front of Pareto-optimal solutions, *i.e.*, that cannot be improved for one reward without sacrificing the other. RS matches the costly yellow front of MORL [Bar+08; Li+20b] requiring multiple trainings on different linear weightings over the rewards $(1-\mu) \times R_1 + \mu \times R_2$ with $0 \leq \mu \leq 1$.

helpful non-evasive responses, and others' [Irv+23] interests are to make LLMs engaging and enjoyable. Even hand-engineered metrics can be in tension: generating shorter descriptions with higher precision can increase the BLEU [Pap+02] score but decrease the ROUGE [Lin+03] score due to reduced recall.

Towards multi-policy strategies. Considering these challenges, a single model cannot be aligned with everyone's preferences [Ouy+22]. Existing works align towards a consensus-based user [Bak+22; Ova23], relying on the "wisdom of the crowd" [Bai+22a], inherently prioritizing certain principles [Kov+23; Joh+22], resulting in unfair representations of marginalized groups [Wei+21; Kir+23; Dur+23]. The trade-offs [Pan+23] are decided a priori before training, shifting the responsibility to the engineers, reducing transparency and explainability [Hay+22], and actually aligning towards the "researchers designing the study" [Ouy+22; San+23]. These limitations, discussed in Section 6.4, highlight the inability of single-policy alignment strategies to handle human diversity. Yet, "human-aligned artificial intelligence is a multi-objective problem" [Vam+18]. Thus, we draw inspiration from the multi-objective reinforcement learning (MORL) literature [Bar+08; Li+20b; Tan+03; Van+14; Roi+13; Răd+20; Mar+23; Wu+23c] and [Hay+22]; they argue that embracing the heterogeneity of diverse rewards requires shifting from single-policy to multi-policy approaches. As optimality depends on the relative preferences across those rewards, the goal is not to learn a single network optimized on a single a priori reward, but rather to uncover a **set of Pareto-optimal networks** [Par64] across the entire space of preferences.

In this chapter, we propose **rewarded soup** (RS) [1], an efficient and flexible multi-policy strategy to fine-tune any foundation model. As shown in Figure 6.1(a), we first use RL to learn one network for each proxy reward independently; then, we combine these expert networks according to user preferences. This a posteriori selection allows for better-informed trade-offs, improved transparency and increased fairness [Hay+22; Man+21a]. The method to combine those networks is a key contribution: rather than functional ensembling, we perform *weight interpolation*. This is in line with the previous findings from previous chapters on linear mode connectivity (LMC) [Fra+20; Ney+20]: weights fine-tuned from a shared pre-trained initialization remain linearly connected and thus can be interpolated. Unlike previous chapters which focused on supervised learning, we explore LMC in RL, in a challenging setup where each training run uses a different reward. Then, we can trade off the capabilities of multiple weights in a single final model, *without any computational overhead*; this enables the creation of custom weights for any preference over the diverse rewards.

- We propose rewarded soup for RL fine-tuning of foundation models when considering diverse rewards (Section 6.2.1).
- We analyze the LMC between weights RL fine-tuned on diverse rewards and the Pareto-optimality of our strategy (Section 6.2.2)
- We demonstrate the effectiveness of rewarded soups across a variety of tasks: RLHF fine-tuning of LLaMA for text-to-text (summarization, QA, helpful assistant, review), multimodal text-image tasks (image captioning, text-to-image generation with diffusion models, visual grounding, VQA), as well as control (locomotion) tasks (Section 6.3).

This chapter has led to the submission of the following paper: Alexandre Ramé, Guillaume Couairon, Mustafa Shukor, Corentin Dancette, Jean-Baptiste Gaya, Laure Soulier, and Matthieu Cord. “Rewarded soups: towards Pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards”. In: *arXiv preprint* (2023). Further information and resources related can be found on our website.

6.2 Rewarded soups

6.2.1 RL fine-tuning with diverse rewards

We consider a deep neural network f of a fixed non-linear architecture. It defines a policy by mapping inputs x to $f(x,\theta)$ when parametrized by θ. For a reward $\hat{R}$ (evaluating the correctness of the prediction according to some preferences) and a test domain T of deployment, our goal is to maximize $\int_{x\in T}\hat{R}\left(f\left(x,\theta\right)\right)$. For example, with f a LLM, x would

[1] The name *rewarded soup* follows the terminology of *model soup* [Wor+22a], as we combine various *ingredients* each rewarded differently.

be textual prompts, $\hat{R}$ would evaluate if the generated text is harmless [Ask+21], and T would be the distribution of users' prompts. Learning the weights θ is now commonly a three-step process: unsupervised pre-training, supervised fine-tuning, and reward optimization. Yet $\hat{R}$ is usually not specified before test time, meaning we only have a proxy reward R during reward optimization. This *reward misspecification* between R and $\hat{R}$ may hinder the alignment of the network with $\hat{R}$. Moreover, the *diversity of human preferences* complicates the design of R.

Rather than optimizing one single proxy reward, our paper's first key idea is to consider a family of N diverse proxy rewards $\{R_i\}_{i=1}^N$. Each of these rewards evaluates the prediction according to different (potentially conflicting) criteria. The goal then becomes obtaining a coverage set of policies that trade-off between these rewards. To this end, we first introduce the costly MORL baseline. Its inefficiency motivates our rewarded soups, which leverages our second key idea: weight interpolation (WI).

MORL baseline. The standard multi-objective reinforcement learning (MORL) scalarization strategy [Bar+08; Li+20b] (recently used in [Wu+23c] to align LLMs) linearizes the problem by interpolating the proxy rewards using M different weightings. Specifically, during the *training phase*, M trainings are launched, with the j-th optimizing the reward $\sum_{i=1}^N \mu_i^j R_i$, where $\forall j \in \{1, ..., M\}, \{\mu_i^j\}_{i=1}^N \in \Delta_N$ the N-simplex s.t. $\sum_{i=1}^N \mu_i^j = 1$ and $0 \leq \mu_i^j \leq 1$. Then, during the *selection phase*, the user's reward $\hat{R}$ becomes known and the j-th policy that maximizes $\hat{R}$ on some validation dataset is selected. We typically expect to select j such that $\sum_{i=1}^N \mu_i^j R_i \approx \hat{R}$ linearly approximates the user's reward. Finally, this j-th weight is used during the *inference phase* on test samples. Yet, a critical issue is that "minor [preference] variations may result in significant changes in the solution" [Vam+08]. Thus, a high level of granularity in the mesh of Δ_N is necessary. This requires explicitly maintaining a large set of $M \gg N$ networks, practically one for each possible preference. Ultimately, this MORL strategy is unscalable in DL due to the *computational, memory, and engineering costs* involved.

Rewarded soups (RS). In this chapter, we draw inspiration from previous chapters and leverage *weight interpolation*. The idea is to learn expert weights and interpolate them linearly to combine their abilities. Specifically, we propose RS, illustrated in Figure 6.1(a) and whose recipe is described below. By design, RS alleviates MORL's scaling issue as it requires only $M = N$ trainings, while being flexible and transparent.

1. During the *training phase*, we optimize N expert weights $\{\theta_i\}_{i=1}^N$, each corresponding to one of the N proxy rewards $\{R_i\}_{i=1}^N$, from a shared pre-trained initialization.
2. For the *selection phase*, we linearly interpolate those weights to define a continuous set of rewarded soups policies: $\{\sum_{i=1}^N \lambda_i \cdot \theta_i\}_{\{\lambda_i\}_{i=1}^N \in \Delta_N}$. Practically, we uniformly sample M interpolating coefficients $\{\{\lambda_i^j\}_{i=1}^N\}_{j=1}^M$ from the N-simplex Δ_N and select the j-th that maximizes the user's reward $\hat{R}$ on validation samples, *i.e.*, $\arg\max_{j=1}^M \hat{R}\left(\sum_{i=1}^N \lambda_i^j \theta_i\right)$.
3. For the *inference phase*, we predict using the network f parameterized by $\sum_{i=1}^N \lambda_i^j \theta_i$.

While MORL interpolates the rewards, RS interpolates the weights. This is a considerable advantage as the appropriate interpolating coefficients λ, which depends on the desired trade-off, can be selected *a posteriori*; the selection is achieved without additional training, only via inference on some samples. In the next Section 6.2.2 we explicitly state the Hypotheses 6.1 and 6.2 underlying in RS. Their empirical verification will be the main motivation for our experiments on various tasks in Section 6.3.

6.2.2 Exploring the properties of rewarded soup

6.2.2.1 Linear mode connectivity of weights fine-tuned on diverse rewards

We consider $\{\theta_i\}_{i=1}^N$ fine-tuned on $\{R_i\}_{i=1}^N$ from a shared pre-trained initialization. Previously in Observation 4.1, the LMC was defined w.r.t. a single performance measure (*e.g.*, accuracy) in supervised learning. We now extend this notion in RL with N rewards, and define that the LMC holds if all rewards for the interpolated weights exceed the interpolated rewards. It follows that the LMC condition which underpins RS's viability is the Hypothesis 6.1 below.

Hypothesis 6.1 (LMC). $\forall \{\lambda_i\}_i \in \Delta_N$ *and* $k \in \{1, \dots, N\}$, $R_k\left(\sum_i \lambda_i \cdot \theta_i\right) \geq \sum_i \lambda_i R_k\left(\theta_i\right)$.

6.2.2.2 Pareto optimality of rewarded soups

The Pareto front (PF) is the set of undominated weights, for which no other weights can improve a reward without sacrificing another, *i.e.*,

$$PF = \{\theta \mid \nexists \theta' \in \Theta \text{ s.t. } \left\{R_i\left(\theta'\right)\right\}_{i=1}^N >_N \left\{R_i\left(\theta\right)\right\}_{i=1}^N\}, \tag{6.1}$$

where $>_N$ is the dominance relation in $\mathcal{R}^N$. In practice, we only need to retain one policy for each possible value vector, *i.e.*, a Pareto coverage set (PCS). We now introduce the key Hypothesis 6.2.

Hypothesis 6.2 (Pareto optimality). *The set* $\{\sum_i \lambda_i \cdot \theta_i | \{\lambda_i\}_i \in \Delta_N\}$ *is a PCS of* $\{R_i\}_i$.

Hypothesis 6.2 holds if the rewarded soups solutions, uncovered by interpolation, are Pareto-optimal. Overall, we empirically validate Hypothesis 6.1 and Hypothesis 6.2 in Section 6.3. Moreover, we theoretically prove in Lemma 6.1 that Hypothesis 6.2 holds when rewards are replaced by their second-order Taylor expansion with Hessians proportional to the identity (or only co-diagonalizable in Appendix C.4), a simplified setup justifiable when weights remain close.

Lemma 6.1. *Let* $\hat{\mu} = (\hat{\mu}_1, \dots, \hat{\mu}_N) \in \Delta_N$. *We assume that the* N *rewards are quadratic; thus we can write for* $i \in \{1, \dots, N\}$:

$$\forall \theta \in \Theta, \quad R_i(\theta) = R_i(\theta_i) - \eta_i \|\theta - \theta_i\|^2 \tag{6.2}$$

Then the reward $R_{\hat{\mu}} = \sum_i \hat{\mu}_i \times R_i$ *is maximized on the convex hull of* $\{\theta_1, \dots, \theta_N\}$.

Proof. The function $R_{\hat{\mu}}$ is quadratic thus has an unique global maximum $\hat{\theta}$:

$$\begin{aligned}\nabla_\theta R_{\hat{\mu}}(\hat{\theta}) = 0 &\implies \sum_{i=1}^{N} \mu_i \eta_i \cdot (\hat{\theta} - \theta_i) = 0 \\ &\implies \hat{\theta} = \frac{\sum_{i=1}^{N} \hat{\mu}_i \eta_i \cdot \theta_i}{\sum_{i=1}^{N} \hat{\mu}_i \eta_i}\end{aligned}$$

Since all the $\hat{\mu}_i \eta_i$ are positive or zero, and at least one is greater than zero, $\hat{\theta}$ is indeed in the convex hull of $\{\theta_1, \ldots, \theta_N\}$. □

Remark 6.1. *Hypotheses 6.1 and 6.2 rely on a good pre-trained initialization, making RS particularly well-suited to fine-tune foundation models. This is because pre-training prevents the weights from diverging during training [Ney+20]. When the weights remain close, we can theoretically justify Hypotheses 6.1 and 6.2 by leveraging the similarity between WI and weighted functional ensembling, as confirmed empirically in Figure 6.4(c). In contrast, the LMC would not hold when training from scratch [Ney+20]. Moreover, pre-training addresses stability and exploration issues [Xie+22; Yan+23b; Sek+20] in RL.*

Remark 6.2. *Pareto-optimality in Hypothesis 6.2 is defined w.r.t. a set of possible weights Θ. Yet, in full generality, improvements in initialization, RL algorithms, data, or specific hyperparameters could enhance performances. In other words, for real-world applications, the true PF is unknown and needs to be defined w.r.t. a training procedure. In this case, Θ represents the set of weights attainable by fine-tuning within a shared procedure. As such, in Section 6.3 we analyze Hypothesis 6.2 by comparing the fronts obtained by RS and scalarized MORL while keeping everything else constant.*

6.2.2.3 Consequences of Pareto optimality for linear preferences

Lemma 6.2 (Reduced reward misspecification)**.** *If Hypothesis 6.2 holds, and for linear reward $\hat{R} = \sum_i \hat{\mu}_i R_i$ with $\{\hat{\mu}_i\}_i \in \Delta_N$, then $\exists \{\lambda_i\}_i \in \Delta_N$ such that $\sum_i \lambda_i \cdot \theta_i$ is optimal for $\hat{R}$.*

In simpler terms, Lemma 6.2 implies that if Hypothesis 6.2 is true, then RS can mitigate reward misspecification. The proof directly follows the definition of Pareto optimality. For any preference $\hat{\mu}$, there exists a λ such that the λ-interpolation over weights maximizes the $\hat{\mu}$-interpolation over rewards. In practice, as we will see in Figure 6.4(a), we can set $\lambda = \hat{\mu}$, or cross-validate λ on other samples. Yet, this theoretically holds only for $\hat{R}$ linear over the proxy rewards. This follows the *linear utility functions* setup from the MORL literature [Răd+20], with limitations: for example, all human preferences can not be expressed linearly [Vam+08]. This motivates having sufficiently rich and diverse proxy rewards to capture the essential aspects of all possible users' rewards. Despite the lack of theoretical guarantees, we will show in Figure 6.4(b) that weight interpolation improves results even for non-linear $\hat{R}$.

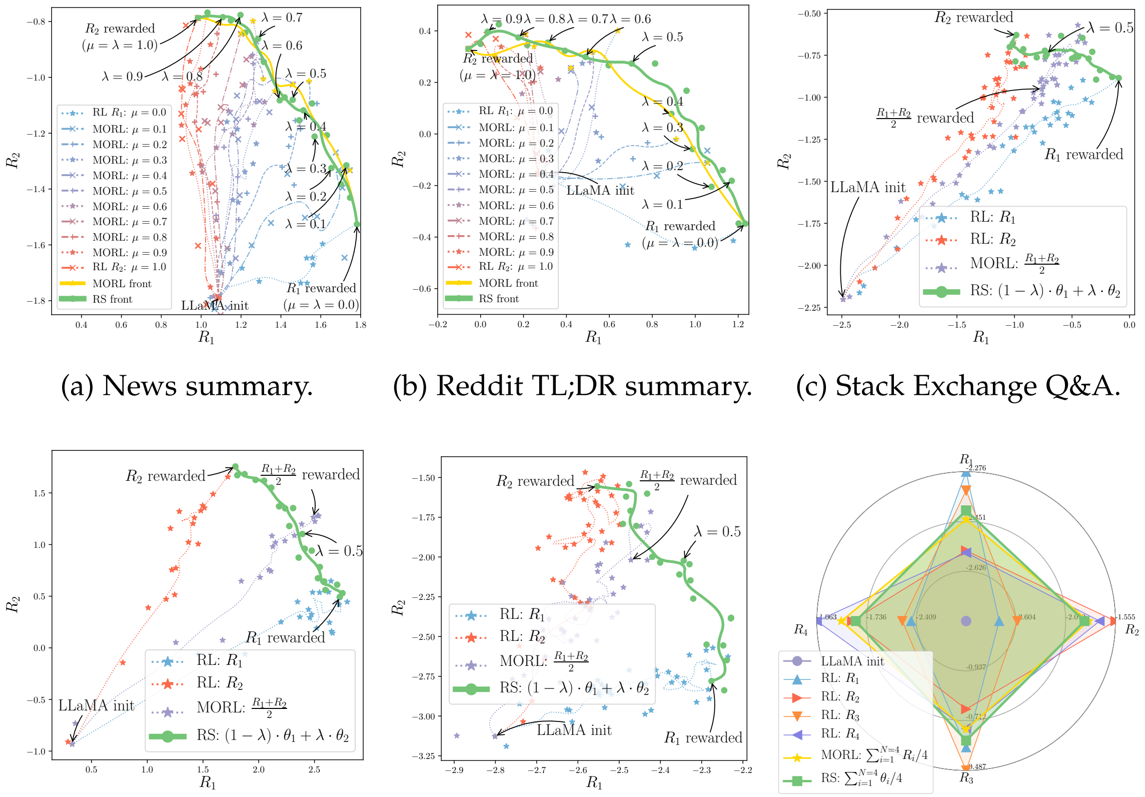
6.3 Experiments

In this section we implement RS across a variety of standard learning tasks: text-to-text generation, image captioning, image generation, visual grounding, visual question answering, and locomotion. We use either model or statistical rewards. We follow a systematic procedure. *First*, we independently optimize diverse rewards on training samples. For all tasks, we employ the default architecture, hyperparameters and RL algorithm; the only variation being the reward used across runs. *Second*, we evaluate the rewards on the test samples: the results are visually represented in series of plots. *Third*, we verify Hypothesis 6.1 by examining whether RS's rewards exceed the interpolated rewards. *Lastly*, as the true PF is unknown in real-world applications, we present empirical support for Hypothesis 6.2 by comparing the front defined by RS (sliding λ between 0 and 1) to the MORL's solutions optimizing the μ-weighted rewards (sometimes only $\mu = 0.5$ for computational reasons). Our website provides additional qualitative results.

6.3.1 Text-to-text: LLaMA with diverse RLHFs

Given the importance of RLHF to train LLMs, we begin our experiments with text-to-text generation. Our pre-trained network is LLaMA-7b [Tou+23a], instruction fine-tuned [Wei+22a; Wan+22b] on Alpaca [Tao+23]. For RL training with PPO [Sch+17], we employ the trl package [Wer+20] and the setup from [Bee+23] with low-rank adapters (LoRA) [Hu+22b] for efficiency. We first consider summarization [Sti+20b; Wu+21] tasks on two datasets: Reuter news [Ahm17] in Figures 6.1(b) and 6.2(a) and Reddit TL;DR [Völ+17] in Figure 6.2(b). We also consider answering Stack Exchange questions [Lam+23] in Figure 6.2(c), movie review generation in Figure 6.2(d), and helpfulness as a conversational assistant [Bai+22a] in Figures 6.2(e) and 6.2(f). To evaluate the generation in the absence of supervision, we utilized $N = 2$ different reward models (RMs) for each task, except in Figure 6.2(f) where $N = 4$. These RMs were trained on human preferences datasets [Chr+17b] and all open-sourced on HuggingFace [Wol+20]. For example in summarization, R_1 [Sti+20b] mostly evaluates completeness while R_2 [Che+21] evaluates faithfulness; these two criterions are in tension, as improving one often degrades the other. For other tasks, we rely on diverse RMs from OpenAssistant [Köp+23]; though they all assess if the answer is adequate, they differ by their architectures and procedures.

The results are reported in Figure 6.2. The green front, defined by RS between the two weights specialized on R_1 and R_2, is above the straight line connecting those two points, validating Hypothesis 6.1. Second, the front passes through the point obtained by MORL fine-tuning on the average of the two rewards, supporting Hypothesis 6.2. Moreover, when comparing both full fronts, they have qualitatively the same shape; quantitatively in hypervolume [Yen+13] (lower is better, the area over the curve w.r.t. an optimal point), RS's hypervolume is 0.367 vs. 0.340 for MORL in Figure 6.2(a), while it is 1.176 vs. 1.186 in Figure 6.2(b). Finally, in Figure 6.2(f), we use $N = 4$ RMs for the assistant task and



(a) News summary.

(b) Reddit TL;DR summary.

(c) Stack Exchange Q&A.

(d) Useful movie review.

(e) Assistant.

(f) Assistant: spider map.

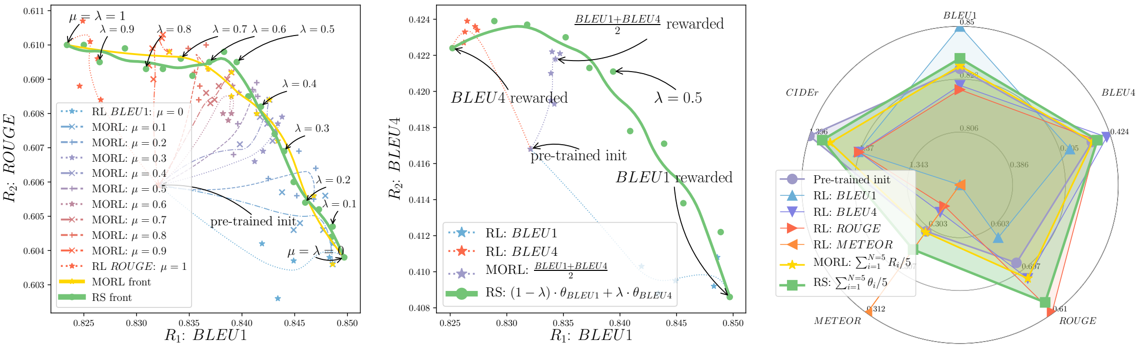
Figure 6.2. – RLHF results in NLP with LLaMA-7b [Tou+23a] and reward models R_i from HuggingFace [Wol+20]. The blue line reports checkpoints' results along the training trajectory of θ_1 rewarding R_1, the red line θ_2 rewarding R_2, and the purple line the MORL rewarding $\frac{R_1+R_2}{2}$. Our rewarded soup (RS) linearly interpolates between the weights θ_1 and θ_2; sliding the interpolation coefficient λ from 0 to 1 reveals the green solid front of rewarded soups solutions. In Figures 6.2(a) and 6.2(b), we additionally show the multiple MORL runs rewarding $(1-\mu) \times R_1 + \mu \times R_2$ with preferences $0 \leq \mu \leq 1$. It reveals a similar yellow front, yet more costly. In Figure 6.2(f), we uniformly ($\lambda_i = \frac{1}{4}$) average the weights fine-tuned for the assistant task on $N = 4$ reward models.

uniformly average the $N = 4$ weights, confirming that RS can scale and trade-off between more rewards.

6.3.2 Image-to-text: captioning with diverse statistical rewards

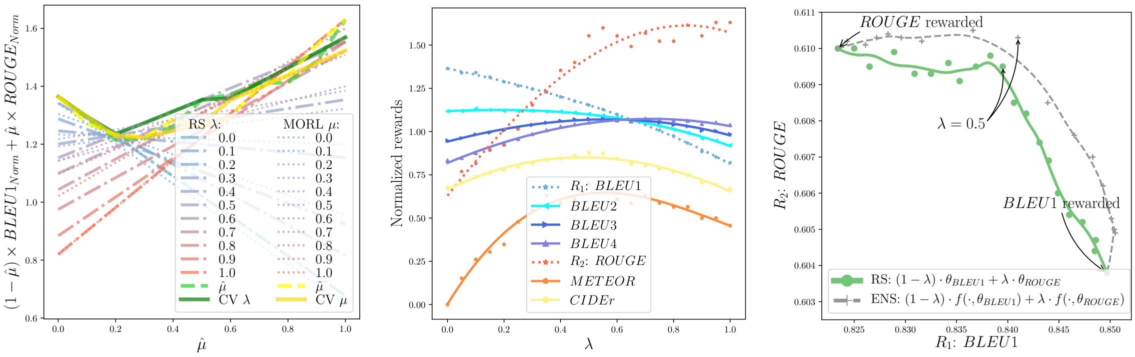
RL training is also effective for multimodal tasks [Pin+23], for example in image captioning [Ren+17] where the task is to generate textual descriptions of images. Precisely evaluating the quality of a prediction w.r.t. a set of human-written captions is a challenging task, thus the literature relies on various hand-engineered, non-differentiable metrics: *e.g.*, the precision-focused BLEU [Pap+02], the recall-focused ROUGE [Lin+03], METEOR [Ban+05] handling synonyms and CIDEr [Ved+15] using TF-IDF. As these metrics are proxies for human preferences, good trade-offs are desirable. We conduct our experiments on COCO [Lin+14], with an ExpansionNetv2 [Hu+22a] network and a Swin Transformer [Liu+22] visual encoder, initialized from the sota weights of [Hu+22a] optimized on CIDEr. We then utilize the code of [Hu+22a] and their self-critical [Ren+17] procedure (a variant of REINFORCE [Wil92]) to reward the network on BLEU1, BLEU4, ROUGE or METEOR.

We observe in Figure 6.3 that tuning solely BLEU1 sacrifices some points on ROUGE or BLEU4. Yet interpolating between θ_1 and θ_2 uncovers a convex set of solutions approximating the ones obtained through scalarization of the rewards in MORL. When comparing



(a) BLEU1 and ROUGE. (b) BLEU1 and BLEU4. (c) Captioning: spider map.

Figure 6.3. – Results in image captioning on COCO [Lin+14]. As rewards R_1 (blue stars every epoch) and R_2 (red stars), we consider standard statistical metrics: BLEU1 (1-gram overlap), BLEU4 (4-grams overlap), ROUGE, METEOR and CIDEr. Figure 6.3(a) include the MORL training trajectories optimizing $(1-\mu)\times BLEU1+\mu\times ROUGE$, uncovering a yellow front similar to RS's green front. In Figure 6.3(c), RS uniformly averages the 5 weights (one for each reward), resulting in the largest area and the best trade-off between the 5 rewards.



(a) Analysis of Lemma 6.2. (b) Performances w.r.t. λ. (c) RS vs. ensembling.

Figure 6.4. – Refined results in captioning with $R_1 = BLEU1$ and $R_2 = ROUGE$. Figure 6.4(a) empirically validates Lemma 6.2 by reporting results of RS (for varying λ) and of MORL (for varying μ) for varying user's preference $\hat{\mu}$. In Figure 6.4(b), all rewards are used for evaluation as a function of the interpolating coefficient. In Figure 6.4(c), we report the front of the costly functional ensembling [Han+90; Lak+17] of predictions (rather than the weight interpolation).

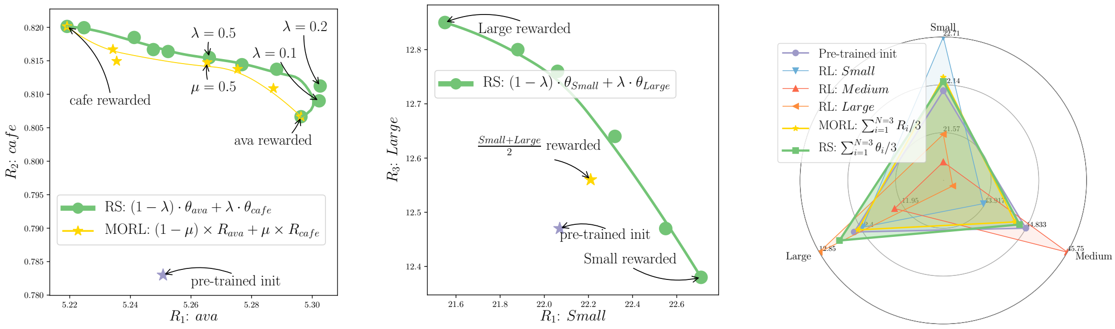
both full fronts in Figure 6.3(a), they qualitatively have the same shape, and quantitatively the same hypervolume [Yen+13] of 0.140. One of the strengths of RS is its ability to scale to any number of rewards. In Figure 6.3(c), we uniformly ($\lambda_i = \frac{1}{5}$) average $N = 5$ weights fine-tuned independently. It improves upon the initialization [Hu+22a] and current sota on all metrics, except for CIDEr, on which [Hu+22a] was explicitly optimized.

Figure 6.4 refines our analysis of RS. In Figures 6.4(a) and 6.4(b), rewards are normalized to 1 for the initialization and 0 for the worst model. Figure 6.4(a) validates Lemma 6.2: for any linear preference $\hat{\mu}$ over the proxy rewards, there exists an optimal solution in the set described by RS. Two empirical strategies to set the value of λ are close to optimal: selecting $\lambda = \hat{\mu}$ if $\hat{\mu}$ is known, or cross-validating (CV λ) if a different data split [Kar+15] is available. Moreover, Figure 6.4(b) investigate all metrics as evaluation. Excluding results' variance, we observe monotonicity in both training rewards, linear in BLEU1 and quadratic in ROUGE. For other evaluation rewards that **cannot be linearly expressed** over the training rewards, the curves' concavity shows that RS consistently improves the endpoints, thereby mitigating reward misspecification. The optimal λ depends on the similarity between the evaluation and training rewards: *e.g.*, best BLEU2 are with small λ. Lastly, as per Lemma 4.1, Figure 6.4(c) shows that *RS succeeds because WI approximates functional ensembling*, interpolating the predictions rather than the weights. Actually, ensembling performs better, but it cannot be fairly compared as its inference cost is doubled.

6.3.3 Text-to-image: diffusion models with diverse RLHFs

Beyond text generation, we now apply RS to align text-to-image generation with human feedbacks [Lee+23; Wu+23a; Xu+23]. Our network is a diffusion model [Ho+20] with 2.2B parameters, pre-trained on a dataset from MetaAI of 300M images; it reaches similar quality as Stable Diffusion [Rom+22], which was not used for copyright reasons. To represent the subjectivity of human aesthetics, we employ $N = 2$ open-source reward models: *ava*, trained on the AVA dataset [Mur+12], and *cafe*, trained on a mix of real-life and manga images. We first generate 10000 images; then, for each reward, we remove half of the images with the lowest reward's score and fine-tune 10% of the parameters [Xie+23] on the reward-weighted negative log-likelihood [Lee+23]. As a side note, on-policy RL would require performing loops of image generations and model fine-tunings [Don+23b], but we only perform a single *offline* iteration for simplicity.

The results displayed in Figure 6.5(a) validate Hypothesis 6.1, as the front described by RS when sliding λ from 0 and 1 is convex. Moreover, RS gives a better front than MORL, validating Hypothesis 6.2. Interestingly, the *ava* reward model seems to be more general-purpose than *cafe*, as RL training on *ava* also enhances the scores of *cafe*. In contrast, the model θ_{cafe} performs poorly in terms of *ava* in Figure 6.5(a). Nonetheless, RS with $(1-\lambda)\cdot\theta_{ava} + \lambda\cdot\theta_{cafe}$ outperforms θ_{ava} alone, not only in terms of *cafe*, but also of *ava* when $\lambda \in \{0.1, 0.2\}$. These findings confirm that RS can better align text-to-image models with a variety of aesthetic preferences. This ability to adapt at test time paves the way for a new form of user interaction with text-to-image models, beyond prompt engineering.



(a) Image generation.

(b) VG: Small and Large.

(c) VG: spider map.

Figure 6.5. – Figure 6.5(a) reports our RLHF experiments on text-to-image generation with diffusion models. From the pre-trained initialization, we learn θ_{ava} and θ_{cafe} by optimizing the two reward models *ava* and *cafe*. Interpolation between them reveals the green Pareto-optimal front, above the yellow MORL front. Figures 6.5(b) and 6.5(c) report our results in visual grounding on RefCOCO+ [Yu+16], where we optimize to predict boxes with IoU> 0.5 w.r.t. the ground-truth, for objects of either small, medium or large size.

6.3.4 Text-to-box: visual grounding of objects with diverse sizes

We now consider visual grounding (VG) [Yu+16]: the task is to predict the bounding box of the region described by an input text. We use a seq-to-seq unified model predicting the box auto-regressively as a sequence of location tokens [Wan+22a]. This model is pre-trained on a large image-text dataset, then fine-tuned with cross-entropy for VG; finally, we use a weighted loss between the cross-entropy and REINFORCE in the RL stage. As the main evaluation metric for VG is the accuracy (*i.e.*, intersection over union (IoU) > 0.5), we consider 3 non-differentiable rewards: the accuracy on small, medium, and large objects. We design this experimental setup because improving results on all sizes simultaneously is challenging, as shown in Figure 6.5(c), where MORL performs similarly to the initialization.

The results in Figure 6.5(b) confirm that optimizing for small objects degrades performance on large ones; fortunately, interpolating can trade-off. In conclusion, we can adapt to users' preferences at test time by adjusting λ, which in turn changes the object sizes that the model effectively handles. On the one hand, if focusing on distant and small objects, a large coefficient should be assigned to θ_{Small}. On the other hand, to perform well across all sizes, we can recover initialization's performances by averaging uniformly (in Figure 6.5(c)).

6.3.5 Text&image-to-text: VQA with diverse statistical rewards

We explore visual question answering (VQA), where the task is to answer questions about images. we use the OFA model [Wan+22a] (generating the answers token-by-token), on the VQA v2 dataset, pre-trained with cross-entropy, and fine-tuned with REINFORCE during the RL stage. During the RL fine-tuning, we explore as rewards the BLEU (1-gram) and METEOR metrics: these metrics enable assigning partial credit if the ground-truth and predicted answers are not identical but still have some words in common.

Our results in Figure 6.6 validate the observations already made in previous experiments: RL is efficient to optimize those two rewards, and RS reveals a Pareto-optimal front to balance between them.

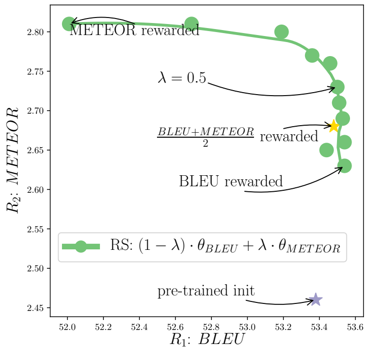


Figure 6.6. – VQA results.

6.3.6 Locomotion with diverse engineered rewards

Teaching humanoids to walk in a human-like manner [Dua+16] serves as a benchmark to evaluate RL strategies [Ng+99] for continuous control. One of the main challenges is to shape a suitable proxy reward [Dor+94; Dew14], given the intricate coordination and balance involved in human locomotion. It is standard [Tod+12] to consider dense rewards of the form $R = velocity - \alpha \times \sum_t a_t^2$, controlling the agent's velocity while regularizing the actions $\{a_t\}_t$ taken over time. Yet, the penalty coefficient α is challenging to set. To address this, we devised two rewards in the Brax physics engine [Fre+21]: a risky R_1 with $\alpha = 0$, and a more cautious R_2 with $\alpha = 1$.

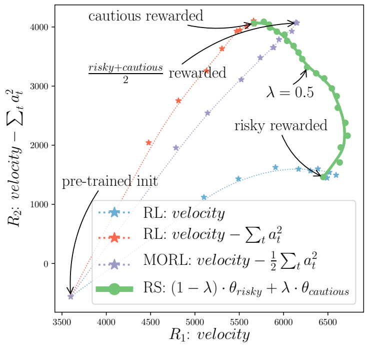


Figure 6.7. – Locomotion results.

Like in all previous tasks, RS's front in Figure 6.7 exceeds the interpolated rewards, as per Hypothesis 6.1. Moreover, the front defined by RS indicates an effective balance between risk-taking and cautiousness, providing empirical support for Hypothesis 6.2, although MORL with $\mu = 0.5$ (*i.e.*, $\alpha = 0.5$) slightly surpasses RS's front. For a more qualitative and intuitive assessment, we provide animations of our RL agent's locomotion on our website.

6.4 Discussion and related work

We previously consider the LMC when fine-tuning with different losses in Section 4.6.4 (as in [Cro+23]) or on different tasks in Chapter 5 (as in [Ilh+22; Don+23a; Ram+23a; Dim+22]): the key contribution of this Chapter 6 is to investigate the LMC in RL. Like in supervised learning, we confirm in Figure 6.4(c) that WI succeeds due to its similarity with ensembling. The most similar works are for control system tasks: [Law+23] averaging decision transformers and [Gay+22] explicitly enforcing connectivity in subspaces of policies trained from scratch on a single reward.

When dealing with multiple objectives in DL, the common strategy is to combine them into a single reward [Roi+13; Rǎd+20]. For example, [Gla+22] sum the predictions of a preference RM (as a proxy for helpfulness) and a rule RM (detecting harmful contents, by utilizing itself a set of diverse rules); [Wu+23c] assign different weightings to the relevance/factuality/completeness rewards, thereby customizing how detailed and lengthy the LLMs responses should be. The main reason why **single-policy** approaches are not suitable is that they optimize over a single set of preferences. In contrast, we build a coverage set of Pareto-optimal policies. This is important for the following reasons, mostly first discussed in Kirk *et al.* [Kir+23] and in Hayes *et al.* [Hay+22]. First, the user's true reward is highly uncertain before training. This "semi-blind" [Hay+22] manual process forces a priori and uncertain decisions about the required trade-offs. Critically, RLHF may cause the "tyranny of the crowdworker" [Kir+23], convering the opinions of a few individuals. Moreover, biases are caused by chaotic engineering choices, and "are exacerbated by a lack of [...] documentation" [Kir+23]. In this *dynamic utility function* scenario, RS can quickly adapt with fewer data, by simply adjusting the λ to match new preferences. Finally, RS could also improve the *interpretability* and *explainability* of the decisions.

The **multi-policy** alternatives [Bar+08; Li+20b; Tan+03; Van+14; Mar+23] are usually not suitable because of their *computational costs* required to learn a dense set of policies. To reduce the cost, [Won+20; Yan+20b; Abd+20; Lin+22] build experts and then train a new model to combine them; [Mos+16; Wil+07; Ngu+20] share weights across experts; [Cas+13; Yan+19b; Abe+19; Pes+21] directly train a single model; the recent and more similar work [Hua+23] learns one linear embedding per (locomotion) task that can be interpolated. Yet, these works are mostly for academic benchmarks [Tod+12; Vam+11]; adapting them to larger tasks (*e.g.*, RLHF for foundation models with PPO) is challenging as they modify the training procedure. Among the ensembling-like RL strategies [Wan+10; Mor+15; Raj+17] such as population-based training [Jad+17; Jad+19], [Par+20; Osa+22] explicitly increase the diversity across policies, yet never considering foundation models nor weight interpolation. In contrast, RS is compatible with the inherent *iterative engineering process* of alignment. Indeed, RS can continually include adjusted opinions while preventing forgetting of the old behaviours. This relates to the *continual learning* challenge, and the empirical observations that weight averaging can reduce catastrophic forgetting [Sto+22; Eec+22]. Our strategy only trains the proxy rewards independently and enables the selection of the interpolating coefficient a posteriori. This is especially useful with large number of rewards and thus growing number of combinations. Finally, its distributed nature makes RS parallelizable thus practical in a federated learning setup [McM+17] where data must remain private.

Finally, our a posteriori alignment with users facilitates **personalization** [Sal+23] of models. As discussed in [Kir+23], this could increase usefulness by providing tailored generation, notably to under-represented groups. This is all the more important as human preferences change from time to time. Yet, this personalization comes with risks for individuals of "reinforcing their biases [...] and narrowing their information diet" [Kir+23]. This may worsen the polarization of the public sphere. Under these concerns, we concur with the notion of "personalization within bounds" [Kir+23], with these boundaries potentially set by weights fine-tuned on diverse and carefully inspected rewards.

6.5 Conclusion

As AI systems are increasingly applied to crucial real-world tasks, there is a pressing issue to align them to our specific and diverse needs, while making the process more transparent. In this chapter, we propose rewarded soup, a weight interpolation strategy that efficiently yields Pareto-optimal solutions to mitigate reward misspecification. Our experiments have consistently validated our working LMC hypotheses for various significant large-scale tasks, involving multiple modalities and different fine-tuning strategies. This confirms that weight interpolation can be a practical strategy to approximate functional ensembling for real-world applications, such as RLHF to transform LLMs into helpful and harmless conversational agents. We hope to inspire further research in exploring how the generalization literature in DL can help for alignment, to create AIs handling the diversity of opinions. As we discuss in the closing Chapter 7 of this thesis, this alignment is critical to ensure safe and ethical progress in the future.

Chapter 7

CONCLUSION

We summarize the contributions of this thesis, offer some future directions and finally discuss the risks posed by the recent and rapid progress in AI.

7.1 Contributions

Our works throughout this thesis deal with the efficient ensembling of diverse models, mostly to improve OOD generalization in DL. Theoretical considerations were essential to drive the experiments: yet, for the sake of clarity, we separate theoretical and empirical contributions in this concluding section. We also discuss the societal impact of our work in Appendix B.

7.1.1 Theoretical contributions

Our main theoretical contribution is the initial discussion in Section 2.2 relating variance to diversity shift and bias to correlation shift. This novel *understanding of the two kinds of distribution shifts* opened up ways to predict when ensembling could be beneficial, and identified scenarios where other strategies (such as those based on invariance developed in Appendix E) would be necessary. We clarify *the role of diversity in ensembling*, through its similarity with covariance, and further analyzed it from an information theory perspective in Chapter 3. We also provide new guarantees for the *effectiveness of weight averaging* under distribution shifts, based on its similarity with functional ensembling. We depart from the traditional flatness-based arguments, critically examined and dismantled in Section 4.2.2. Overall, we highlight *trade-offs* in DL: between bias and variance in supervised learning, between individual accuracy and diversity in ensembling, between diversity and locality in weight averaging, and between different rewards in reinforcement learning.

7.1.2 Empirical contributions

deep learning is (above all) an empirical science, where bounds and theoretical arguments often serve more as indicators than absolute truths. We have worked extensively on the empirical benefits of *weight averaging (WA)*: it is scalable, effective in OOD generalization, and aligns well with the inherent *iterative engineering process* of DL. Actually, the main strength of WA lies in its simplicity, *facilitating its adoption by the community*, in line with the "bitter lesson" from [Sut19]. Moreover WA is compatible with other tools and is boosted by the massive adoption of the foundation models [Bom+21] paradigm in the DL community. Actually, we emphasize a novel benefit from this foundation model paradigm; though it is commonly used to increase individual performance, we highlight its ability to tackle the computational burden of ensembling:

- at training: fine-tuning is more efficient than training from scratch, making $M \gg 1$ fine-tunings realistic,
- at inference: as fine-tuning enforces the LMC, allowing us to simply average the M weights rather than the predictions.

A significant part of this thesis revolves around the concept of *diversity*, especially useful to improve OOD generalization. The conclusion from Section 5.4.5 is that, when aiming at OOD with ensembling strategies, there actually exists a trade-off between diversity and ID accuracy. This finding contrasts with [Mil+21] and goes against the prescription in [Wen+22] that, "to make the model more robust on OOD data, the main focus should be to improve the ID classification error". This emphasis on diversity goes beyond the ensembling setup: in Chapter 6, we embrace the diversity of opinions in RL, and show that handling diverse rewards is necessary for better alignment; in Appendix E, we show that diversity in training domains and learned features enhance the robustness of models under correlation shift. Overall, we helped popularize diversity as a key concept.

To summarize, we provide below an analytical formulation with equations of the different fine-tuning strategies considered along this thesis, where θ represents the weights, A_i the auxiliary tasks, D the target task, and R_i the potential different rewards.

$$\theta = \text{Train}\left(\theta^{\text{pt}}, D\right), \qquad \text{[Vanilla fine-tuning [Oqu+14]]}$$

$$\theta = \text{Train}\left(\theta^{\text{pt}}, D, \text{collect\_ckpts} = \text{True}\right), \qquad \text{[Moving average [Izm+18]]}$$

$$\theta = \frac{1}{M}\sum_{i=0}^{M-1} \text{Train}\left(\theta^{\text{pt}}, D\right), \qquad \text{[DiWA [Ram+22b] Chapter 4]}$$

$$\theta = \text{Train}\left(\text{Train}\left(\theta^{\text{pt}}, A_i\right), D\right), \qquad \text{[Inter-training [Pha+18]]}$$

$$\theta = \frac{1}{M}\sum_{i=0}^{M-1} \text{Train}\left(\text{Train}\left(\theta^{\text{pt}}, A_i\right), D\right), \qquad \text{[Model ratatouille [Ram+23a] Chapter 5]}$$

$$\theta = \sum_{i=0}^{M-1} \lambda_i \text{TrainRL}\left(\theta^{\text{pt}}, D, R_i\right), \qquad \text{[Rewarded soups [Ram+23b] Chapter 6]}$$

7.2 Future work

The works presented in this thesis has laid the groundwork for numerous exciting research directions.

7.2.1 Improved generalization

I believe the potential of *RL for OOD generalization* is still relatively unexplored. RL could be harnessed to directly optimize certain invariance or fairness metrics, bypassing the need for differentiable proxies. Notably, this could be applied to optimize directly the true invariance objective IRM [Arj+17] (rather than a proxy differentiable loss) or expected calibration error (ECE) [Guo+17].

Moreover, current strategies for correlation shift always implicitly incorporate fundamental a priori assumptions (within the data split or the validation dataset) concerning which features should be considered spurious or not. Moreover, they often assume the uniqueness of the spurious information (*e.g.*, color in ColoredMNIST), and would fail otherwise (*e.g.*, color and background). Combining such methods with weight interpolation (as in Chapter 6) could provide *Pareto-optimal fairness* solutions [Che+23]. This would facilitate a posteriori decision-making when balancing between (potentially conflicting) diverse fairness constraints [Lop+22], and simplify external regulation.

7.2.2 Weight averaging

Though WA is now widely used by the DL community [Wor23], its *theoretical understanding* is still lacking, even in the simplest case of moving averages [Izm+18]. Our best theory relies on the similarities between weight interpolation and functional ensembling, with guarantees only when weights remain close. Interestingly, this does not explain the following fact: *WA consistently outperforms ENS in OOD* (as shown in Figure 4.5) but the opposite is true ID (as illustrated in [Wor+22a]). To better understand this surprising insight, we have already conducted a preliminary "label corruption" experiment suggesting that WA mitigates memorization. More precisely, we trained two models on the same dataset, but we occasionally corrupt the labels for the first model: we observe that averaging the predictions with ENS can memorize the corrupted labels, yet that averaging the weights with WA actually forgets them. Could this phenomenon be linked to a form of *invariance across different runs*? This represents an intriguing direction to better understand WA robustness.

So far we mostly used WA to improve OOD generalization and alignment, yet WA has numerous other potential *applications*. A fascinating phenomenon is WA's capacity to combine the capabilities of various models. For instance, [Ilh+23] showed that we can edit models with *task arithmetic*; negative coefficients in interpolation can eliminate specific undesired behaviors. Another study [Jan+23] demonstrated that averaging an English summarizer and an English-to-French translator can produce summaries in French. We briefly examined this in our workshop paper [Ram+22c]; yet, this phenomenon remains an active area of research [Ort+23]. More generally, WA has the potential to be a crucial component of *efficient scaling* strategies as a simple parallelization method [Wor+23], eliminating complex multi-node synchronization. This could be refined with data selection strategies such as pruning [Ro+23] and partitioning [Li+22]. As a side note, this is consistent with recent rumors suggesting that GPT-4 [Ope23] is actually an ensemble of sixteen experts GPT-3.5, trained on different corpus; one for maths, one for legal, *etc*.

7.2.3 Towards updatable ML

Furthermore, WA could facilitate *iterative and updatable ML* [Raf23]. We envision a future where DNNs can be incrementally improved and recombined, allowing for the collaborative creation of increasingly sophisticated AI systems. The core idea is to consider networks as pieces of software [Kar17] and mirror the pipelines used for open-source development of software engineering via version control. Recent works [Mat+22; Li+22; Don+23a; Cho+22a] and ratatouille proposed in Chapter 5 give some primitives to learn DNNs in collaboration. Put simply, *git clone* is pre-training, *git commit* are fine-tunings performed by individual contributors on their specific tasks, *git merge* is replaced by WA, and *git test* would try to measure models' statistics and failures on external datasets. Actually, the recent Git-Theta [Kan+23] follows these principles. In terms of privacy, such a federated learning setup [Li+19; Adn+22] where datasets can be kept private does indeed seem desirable. In terms of computation and sustainability, minimal communication across servers enable embarrassingly simple parallelization [Li+22; Wor+23], reducing costs and CO2 emissions. The ability to collaboratively improve weights shifts from *proprietary network training* to *open-source collaborative network* building, leveraging distributed computing resources such as single-GPU desktop machines. We see this as an exciting possibility for the future of AI.

7.2.4 Towards more general AI

A key challenge is transitioning from classification towards more general tasks, involving rewards challenging to define [Kwo+23]. An intriguing approach I plan to explore is reinforcement learning from AI feedback (RLAIF) [Bai+22b], where the rewards are generated by other AI reward models. The key difference with RLHF is that those reward models do not necessarily require human feedback. Factual generation serves as an interesting example; in [Roi+23], the reward model was pre-trained on textual entailment assessment [Dag+05]. Another central idea from [Bai+22b] is to inscribe foundational principles in the rewards, similar to Asimov's universal laws for robots. The *self-recursive* approach [Gou+23; Mad+23; Sun+23] uses AI's own predictions, generating a critique followed by a revision, leveraging LLMs' in-context capabilities that make them generate superior responses when guided with specific "step by step" prompts [Wei+22c; Yao+23]. Those rewards models could be expanded by various tools [Sch+23; Yan+23a], ranging from web access [Nak+21], to employing citation experts [Men+22], expert networks [She+23], or planners [Liu+23]. In relation to this thesis, debates and consensus among different LLMs could also help [Irv+18]. Moreover, rewards usually evaluate the full generation, without clarifying which parts contain what types of errors; a promising idea is to *reward the process* [Lig+23; Wu+23b] rather than the output, to precisely explicitly indicate "which sentence is false, which sub-sentence is irrelevant" [Wu+23b]. The diversity of those potential fine-grained reward models strengthen the *multi-objective* approach introduced in Chapter 6. For example, [Wu+23b] incorporates multi-dimensional evaluation such as factuality, irrelevance, and information incompleteness. This work [Wu+23b] and our rewarded soup [Ram+23b] are limited to linear preferences over those proxy rewards; considering more complex combinations [Vam+18; Vam+08] is a promising direction, as in [Tou+23b] where they leverage a piecewise combination of rewards.

Finally, LLMs have significantly moved us closer to creating general-purpose AI; nevertheless, their current limitation to a single modality (text) restricts their understanding and interaction with the world. The intuition is that an AI, in order to become general, must embody itself, take shape, in order to face the unexpected and solve new problems. To handle diverse tasks across numerous modalities, we thus need robust *multimodal models*, such as the unified models [Ala+22; Wan+22a] and our proposed UniVAL [Shu+23]. To embed those novel AI prior to real-world deployment, they could be deployed in open-world synthetic environments [Zhu+23], both to reduce computational costs and provide better understanding of ethical dilemmas [Pan+23]. This discussion leads us seamlessly into the final and critical part of this thesis: AI safety.

7.2.5 AI safety

The recent and rapid scaling of LLMs poses both opportunities and major concerns [Amo+16], causing a wave of inquiry about the potential negative impacts; scenarios relegated to science fiction just a few years (months?) back seem now less far-fetched. OpenAI [Jan23] summarized it this way:

> *Superintelligence will be the most impactful technology humanity has ever invented, and could help us solve many of the world's most important problems. But the vast power of superintelligence could also be very dangerous, and could lead to the disempowerment of humanity or even human extinction.*

Recently, some leading experts advocated for a *pause* [Lif23] in scaling and deployment. Therefore, without forgetting the fantastic possibilities opened by AI, we need to draw our attention towards the potentially negative societal consequences.

7.2.5.1 Risks

Drawing from [Eve+18; Wei+22d; Bru15], we enumerate below a progression from current harms to more speculative risks.

- Biased discrimination, hate-speech and exclusion.
- Misinformation harms: hallucinations and spread of false news.
- Environmental harms: energy consumption and hardware manufacturing.
- Socioeconomic harms: increased inequality, job displacement, reduction in human creativity, monopoly on hardware.

- Information hazards: privacy breaches and data leaks.
- Individual malicious uses: cyber attacks, fraud.
- Governmental oppressions: surveillance, censorship.
- Human-computer interaction harms: anthropomorphizing and trust exploitation.
- Power-seeking behaviors: (functional) deception and manipulation.

Though there is scientific consensus regarding the current harms first listed, the plausibility of the more speculative risks remains a subject of ongoing debates in the community [AI 23]. While Meredith Whittaker has stated "There's no more evidence now than there was in 1950 that AI is going to pose these existential risks,", there are actually recent works suggesting the contrary [Hen+23; Hen+22; Hen23].

7.2.5.2 On the difficulty of alignment

The high uncertainty surrounding these questions calls for caution, and highlight the importance of *building human-aligned AI that behaves in accordance with what we want*. This *alignment* [Tay+16; Ken+21; Ngo+22] is complex, with challenges usually split into two categories: *outer alignment* and *inner alignment*.

Outer alignment. How can we design a robust and reliable reward aligned with the intended goal of its designers? This comes with several challenges:

- Proxy gaming. There is a risk that reward models could be exploited adversarially [Ska+22]. Then an AI may find loopholes or shortcuts to maximize its rewards without achieving the desired behaviour, a phenomenon known as wireheading.
- Lack of robustness [Gao+22]. There is a risk that the reward does not sufficiently cover all potential harms. Then an AI could meet the desired objective while neglecting long-term consequences and negative side-effects.

More broadly, these issues refer to Goodhart's law [Smi21]: "when a measure becomes a target, it ceases to be a good measure".

Inner alignment. The second question is: assuming a suitable reward, how can we make sure an AI truely optimizes it on test samples? This comes with several challenges:

- Underfitting, due to a lack of exploration (an inherent challenge in RL) leading to poor performances (even in train).
- Overfitting, due to the optimization of a specific reward on specific training samples, leading to poor generalization on samples not seen in training.
- Goal misgeneralization [Sha+22; Di +22] caused by spurious correlations between domains and rewards in training.

Actually, without constraints on the test distribution, complete inner alignment may be impossible [Wol+23], for instance, for LLMs with prompts of arbitrary (long) length.

Remark 7.1. *An intriguing yet-to-developed solution to this inner-alignment challenge is* objective-driven AI*, as advocated by Yann LeCun [LeC22]; instead of the current RL fine-tuning paradigm, they would use a reward-guided decoding, maximizing objectives during inference time when acting in the world; we would only accept model predictions if they meet specific safety constraints (as in sample-and-rank [Kul+20]), leading to a controllable AI that behave appropriately.*

7.2.5.3 Alignment as OOD generalization?

The encouraging aspect is that these challenges are intricately related to the focus of this thesis: how to prevent failures under novel test distributions.

- Outer-alignment appears as an OOD challenge from the perspective of the reward model. Indeed, reward models are trained on a preference dataset whose inputs are generated by a base LLM and annotated by humans; then, they are used to evaluate generations of this LLM but updated multiple times. The temporal gap and those updates indicate a distribution shift, and thus requiring robust reward models generalizing beyond the preference dataset. To tackle the distribution shift, seminal alignment strategies [Chr+17b] used ensembling of reward models. More recently in LLaMA 2 [Tou+23b], to ensure that the reward models remain within distribution, they continually accumulate new reward modeling data.
- Inner alignment appears as an OOD challenge from the perspective of the AI. Indeed, most issues stem from a situation where the AI aligns with the reward on training samples, but this alignment does not persist due to distribution shift between the fine-tuning distribution and the real-world deployment.

Given this, I believe that *tools from the OOD generalization literature could help to tackle the alignment challenge*. For instance, in Chapter 6 we reduce reward misspecification using WA strategies first developed in Chapters 4 and 5 to reduce model underspecification [DAm+20] for OOD generalization. That was a first step towards improved alignment, requiring further investigation.

Openings

The rising importance of AI in our daily lives renders these ethical and technical questions increasingly critical. As I conclude this thesis, I would like to offer a more personal reflection on the matter. I posit that AI will soon become a crucial point of debate in our society, akin to the discourses surrounding the current ecological crisis. More specifically, I anticipate that AI will emerge as a new axis of left-right political polarization, with the left advocating for stringent regulation to protect workers and ensure fairness, and the right adopting a more liberal approach and optimistic view on the benefits of AI, to stimulate economic growth and improve the quality of life. Those issues are multifaceted and certainly call for nuanced perspectives.

Need for better risk evaluation. Then it is our goal as scientists to provide well-founded answers to those technical questions, assisting regulators, policymakers, and the general public in making informed decisions. A crucial aspect of this is striving for a scientific understanding and consensus about the inner workings (potentially through mechanistic interpretability [Nan+23]) and the abilities (through scalable oversight [Bow+22]) of these systems; this will enable us to accurately estimate any potential risks they may pose. We could learn a lot from climate scientists' efforts, failures and successes, and their international panels such as the IPCC (a.k.a. GIEC). Fostering synergy between big tech, public labs, and governments could aid us in ensuring safety precautions and in circumventing any potential *prisoner's dilemma* situations.

Need for more robust models. As scientists, our responsibility also lies in enhancing the robustness of these models and providing better safety guarantees, while mitigating the (currently present) sociological and environmental harms, as further discussed in Appendix B. This could pave the way for a novel era of prosperity and well-being, and assist humanity in addressing the numerous challenges we currently face. Moving forward, I plan to continue working on generalization, and investigating how the generalization literature can be used for alignment. I eagerly look forward to contributing to the development of AI systems that bring benefits to our society as a whole.

BIBLIOGRAPHY

[Aba+16] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. “Deep learning with differential privacy”. In: *ACM SIGSAC*. 2016 (cit. on p. 131).

[Abd+20] Abbas Abdolmaleki, Sandy Huang, Leonard Hasenclever, Michael Neunert, Francis Song, Martina Zambelli, Murilo Martins, Nicolas Heess, Raia Hadsell, and Martin Riedmiller. “A distributional view on multi-objective policy optimization”. In: *ICML*. 2020 (cit. on p. 75).

[Abe+19] Axel Abels, Diederik Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. “Dynamic weights in multi-objective deep reinforcement learning”. In: *ICML*. 2019 (cit. on p. 75).

[Abe+23] Taiga Abe, E Kelly Buchanan, Geoff Pleiss, and John P Cunningham. “Pathologies of Predictive Diversity in Deep Ensembles”. In: *arXiv preprint* (2023) (cit. on p. 35).

[Abn+22] Samira Abnar, Mostafa Dehghani, Behnam Neyshabur, and Hanie Sedghi. “Exploring the Limits of Large Scale Pre-training”. In: *ICLR*. 2022 (cit. on p. 37).

[Adn+22] Mohammed Adnan, Shivam Kalra, Jesse C Cresswell, Graham W Taylor, and Hamid R Tizhoosh. “Federated learning and differential privacy for medical image analysis”. In: *Nature Scientific reports* (2022) (cit. on p. 80).

[Adv14] National Center for Advancing Translational Sciences. *Tox21 Data Challenge*. https://tripod.nih.gov/tox21/challenge/index.jsp. 2014 (cit. on p. 2).

[Aga18] Abien Fred Agarap. “Deep learning using rectified linear units (relu)”. In: *arXiv preprint* (2018) (cit. on p. 11).

[Ah-10] Julien Ah-Pine. “Normalized kernels as similarity indices”. In: *PAKDD*. 2010 (cit. on pp. 17, 134).

[Ahm+21] Faruk Ahmed, Yoshua Bengio, Harm van Seijen, and Aaron Courville. “Systematic generalisation with group invariant predictions”. In: *ICLR*. 2021 (cit. on p. 163).

[Ahm17] Hadeer Ahmed. “Detecting opinion spam and fake news using n-gram analysis and semantic similarity”. PhD thesis. 2017 (cit. on p. 69).

[Ahu+21] Kartik Ahuja, Jun Wang, Amit Dhurandhar, Karthikeyan Shanmugam, and Kush R. Varshney. “Empirical or Invariant Risk Minimization? A Sample Complexity Perspective”. In: *ICLR*. 2021 (cit. on p. 163).

[AI 23] Center for AI safety. *Statement on AI Risk*. https://www.safe.ai/statement-on-ai-risk. 2023 (cit. on p. 82).

[Ain+22] Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. "Git Re-Basin: Merging Models modulo Permutation Symmetries". In: *ICLR*. 2022 (cit. on p. 50).

[Aks03] Matti Aksela. "Comparison of classifier selection methods for improving committee performance". In: *MCS*. 2003 (cit. on pp. 21, 34, 40, 48, 49, 53, 58, 157).

[Ala+22] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. "Flamingo: a visual language model for few-shot learning". In: *NeurIPS* (2022) (cit. on p. 81).

[Ale+17] Alex Alemi, Ian Fischer, Josh Dillon, and Kevin Murphy. "Deep Variational Information Bottleneck". In: *ICLR*. 2017 (cit. on pp. 25, 27, 29, 30, 32, 33, 141).

[Amo+16] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. "Concrete problems in AI safety". In: *arXiv preprint* (2016) (cit. on pp. 63, 81).

[Ang+21] Sumegh Anglekar, Urvee Chaudhari, Atul Chitanvis, and Radha Shankarmani. "Machine Learning Based Risk Assessment Analysis for SMEs Loan Grant". In: *ICCICT*. 2021 (cit. on p. 4).

[Arj+17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks". In: *ICML*. 2017 (cit. on p. 79).

[Arj+19] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. "Invariant Risk Minimization". In: *arXiv preprint* (2019) (cit. on pp. 3, 13, 14, 19, 49, 161–163, 169).

[Arp+21] Devansh Arpit, Huan Wang, Yingbo Zhou, and Caiming Xiong. "Ensemble of Averages: Improving Model Selection and Boosting Performance in Domain Generalization". In: *NeurIPS*. 2021 (cit. on pp. 14, 24, 37, 39, 40, 42, 43, 45, 46, 49, 53, 57).

[Ash+20] Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. "Pitfalls of In-Domain Uncertainty Estimation and Ensembling in Deep Learning". In: *ICLR*. 2020 (cit. on pp. 19, 23, 155).

[Ask+21] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. "A General Language Assistant as a Laboratory for Alignment". In: *arXiv preprint* (2021) (cit. on pp. 63, 66).

[Bai+22a] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. "Training a helpful and harmless assistant with reinforcement learning from human feedback". In: *arXiv preprint* (2022) (cit. on pp. 63, 64, 69).

[Bai+22b] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. "Constitutional AI: Harmlessness from AI Feedback". In: *arXiv preprint* (2022) (cit. on pp. 63, 80).

[Bak+22] Michiel A. Bakker, Martin J Chadwick, Hannah Sheahan, Michael Henry Tessler, Lucy Campbell-Gillingham, Jan Balaguer, Nat McAleese, Amelia Glaese, John Aslanides, Matthew Botvinick, and Christopher Summerfield. "Fine-tuning language models to find agreement among humans with diverse preferences". In: *NeurIPS*. 2022 (cit. on p. 64).

[Ban+05] Satanjeev Banerjee and Alon Lavie. "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments". In: *ACL Workshop*. 2005 (cit. on p. 70).

[Bar+08] Leon Barrett and Srini Narayanan. "Learning all optimal policies with multiple criteria". In: *ICML*. 2008 (cit. on pp. 64, 66, 75).

[Bec+88] Sue Becker and Yann Le Cun. "Improving the Convergence of Back-Propagation Learning with Second Order Methods". In: *Connectionist models summer school*. 1988 (cit. on pp. 144, 169).

[Bee+18] Sara Beery, Grant Van Horn, and Pietro Perona. "Recognition in Terra Incognita". In: *ECCV*. 2018 (cit. on pp. 14, 45).

[Bee+23] Edward Beeching, Younes Belkada, Leandro von Werra, Sourab Mangrulkar, Lewis Tunstall, and Kashif Rasul. *Fine-tuning 20B LLMs with RLHF on a 24GB consumer GPU*. https://huggingface.co/blog/trl-peft. 2023 (cit. on p. 69).

[Bel+18] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. "Mutual information neural estimation". In: *ICML*. 2018 (cit. on pp. 26, 30).

[Ben+17] Hedi Ben-Younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. "Mutan: Multimodal tucker fusion for visual question answering". In: *ICCV*. 2017 (cit. on p. 9).

[Ben+21] Gregory Benton, Wesley Maddox, Sanae Lotfi, and Andrew Gordon Gordon Wilson. "Loss Surface Simplexes for Mode Connecting Volumes and Fast Ensembling". In: *ICML*. 2021 (cit. on p. 45).

[Bis06] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006 (cit. on p. 1).

[Bla+11] Gilles Blanchard, Gyemin Lee, and Clayton Scott. "Generalizing from several related classification tasks to a new unlabeled sample". In: *NeurIPS*. 2011 (cit. on p. 161).

[Bom+21] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. "On the opportunities and risks of foundation models". In: *arXiv preprint* (2021) (cit. on pp. 3, 37, 38, 51, 63, 78).

[Bow+22] Samuel R Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamile Lukosuite, Amanda Askell, Andy Jones, Anna Chen, et al. "Measuring progress on scalable oversight for large language models". In: *arXiv preprint* (2022) (cit. on p. 84).

[Bow13] Nicholas A Bowman. "How much diversity is enough? The curvilinear relationship between college diversity interactions and first-year student outcomes". In: *Research in Higher Education* (2013) (cit. on p. 132).

[Bra+99] Damien Brain and Geoffrey I Webb. "On the effect of data set size on bias and variance in classification learning". In: *AKAW*. 1999 (cit. on p. 135).

[Bre+20] Philip G Breen, Christopher N Foley, Tjarda Boekholt, and Simon Portegies Zwart. "Newton versus the machine: solving the chaotic three-body problem using deep neural networks". In: *Monthly Notices of the Royal Astronomical Society*. 2020 (cit. on p. 2).

[Bre96] Leo Breiman. "Bagging predictors". In: *Machine learning* (1996) (cit. on pp. 5, 19, 22, 153).

[Bro+05a] Gavin Brown, Jeremy Wyatt, and Ping Sun. "Between two extremes: Examining decompositions of the ensemble objective function". In: *MCS*. 2005 (cit. on pp. 20, 22, 41, 139).

[Bro+05b] Gavin Brown, Jeremy L Wyatt, and Peter Tiňo. "Managing diversity in regression ensembles". In: *JMLR* (2005) (cit. on p. 21).

[Bro+20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. "Language Models are Few-Shot Learners". In: *NeurIPS*. 2020 (cit. on p. 63).

[Bru15] Miles Brundage. "Taking superintelligence seriously: Superintelligence: Paths, dangers, strategies by Nick Bostrom (Oxford University Press, 2014)". In: *Futures* (2015) (cit. on p. 81).

[Cam+02] Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. "Deep blue". In: *Artificial intelligence* (2002) (cit. on p. 1).

[Car+20] Luigi Carratino, Moustapha Cissé, Rodolphe Jenatton, and Jean-Philippe Vert. "On Mixup Regularization". In: *arXiv preprint* (2020) (cit. on p. 155).

[Car+21] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. "Emerging Properties in Self-Supervised Vision Transformers". In: *ICCV*. 2021 (cit. on p. 63).

[Cas+13] Andrea Castelletti, Francesca Pianosi, and Marcello Restelli. "A multiobjective reinforcement learning approach to water resources systems operation: Pareto frontier approximation in a single run". In: *Water Resources Research* (2013) (cit. on p. 75).

[Cha+19] Guillaume Charpiat, Nicolas Girard, Loris Felardos, and Yuliya Tarabalka. "Input Similarity from the Neural Network Perspective". In: *NeurIPS*. 2019 (cit. on p. 161).

[Cha+20] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. "Invariant rationalization". In: *ICML*. 2020 (cit. on p. 163).

[Cha+21a] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. "SWAD: Domain Generalization by Seeking Flat Minima". In: *NeurIPS*. 2021 (cit. on pp. 14, 24, 37, 39, 40, 42, 43, 45, 46, 53, 57).

[Cha+21b] Ting-Yun Chang and Chi-Jen Lu. "Rethinking why intermediate-task finetuning works". In: *arXiv preprint* (2021) (cit. on pp. 53, 57).

[Che+16] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *SIGKDD*. 2016 (cit. on p. 19).

[Che+20] Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. "Online Knowledge Distillation with Diverse Peers." In: *AAAI*. 2020 (cit. on pp. 22, 32–34).

[Che+21] Sihao Chen, Fan Zhang, Kazoo Sone, and Dan Roth. "Improving Faithfulness in Abstractive Summarization with Contrast Candidate Generation and Selection". In: *NAACL*. 2021 (cit. on p. 69).

[Che+23] Yongqiang Chen, Kaiwen Zhou, Yatao Bian, Binghui Xie, Bingzhe Wu, Yonggang Zhang, MA KAILI, Han Yang, Peilin Zhao, Bo Han, and James Cheng. "Pareto Invariant Risk Minimization: Towards Mitigating the Optimization Dilemma in Out-of-Distribution Generalization". In: *ICLR*. 2023 (cit. on p. 79).

[Chi+20] Nadezhda Chirkova, Ekaterina Lobacheva, and Dmitry P. Vetrov. "Deep Ensembles on a Fixed Memory Budget: One Wide Network or Several Thinner Ones?" In: *arXiv preprint* (2020) (cit. on pp. 4, 23).

[Cho+22a] Leshem Choshen, Elad Venezian, Shachar Don-Yehia, Noam Slonim, and Yoav Katz. "Where to start? Analyzing the potential value of intermediate models". In: *arXiv preprint* (2022) (cit. on pp. 53, 55, 80).

[Cho+22b] Leshem Choshen, Elad Venezian, Noam Slonim, and Yoav Katz. "Fusing finetuned models for better pretraining". In: *arXiv preprint* (2022) (cit. on pp. 45, 51–55, 57).

[Chr+17a] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. "A Downsampled Variant of ImageNet as an Alternative to the CIFAR datasets". In: *arXiv preprint* (2017) (cit. on p. 154).

[Chr+17b] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. "Deep reinforcement learning from human preferences". In: *NeurIPS*. 2017 (cit. on pp. 4, 63, 69, 83).

[Chu+20] Inseop Chung, SeongUk Park, Jangho Kim, and Nojun Kwak. "Feature-map-level online adversarial knowledge distillation". In: *ACM*. 2020 (cit. on p. 22).

[Coe00] CA Coello. "Handling preferences in evolutionary multiobjective optimization: A survey". In: *CEC*. 2000 (cit. on p. 63).

[CR45] Rao C.R. "Information and accuracy attainable in the estimation of statistical parameters". In: *Bulletin of the Calcutta Mathematical Society*. 1945 (cit. on p. 168).

[Cro+23] Francesco Croce, Sylvestre-Alvise Rebuffi, Evan Shelhamer, and Sven Gowal. "Seasoning Model Soups for Robustness to Adversarial and Natural Distribution Shifts". In: *CVPR*. 2023 (cit. on p. 75).

[Cub+20] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. "RandAugment: Practical Automated Data Augmentation with a Reduced Search Space". In: *NeurIPS*. 2020 (cit. on p. 151).

[Cyb89] George Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of control, signals and systems* (1989) (cit. on p. 12).

[Dab+20] Ali Dabouei, Sobhan Soleymani, Fariborz Taherkhani, Jeremy Dawson, and Nasser M. Nasrabadi. "Exploiting Joint Robustness to Adversarial Perturbations". In: *CVPR*. 2020 (cit. on p. 22).

[Dag+05] Ido Dagan, Oren Glickman, and Bernardo Magnini. "The pascal recognising textual entailment challenge". In: *MLC Workshop*. 2005 (cit. on p. 80).

[DAm+20] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. "Underspecification presents challenges for credibility in modern machine learning". In: *JMLR* (2020) (cit. on pp. 18, 83).

[Dan+20] Felix Dangel, Frederik Kunstner, and Philipp Hennig. "BackPACK: Packing more into Backprop". In: *ICLR*. 2020 (cit. on p. 162).

[Dan+21] Felix Dangel, Lukas Tatzel, and Philipp Hennig. "ViViT: Curvature access through the generalized Gauss-Newton's low-rank structure". In: *arXiv preprint* (2021) (cit. on p. 168).

[Dan17] Amit Daniely. "SGD learns the conjugate kernel class of the network". In: *NeurIPS*. 2017 (cit. on p. 17).

[DAs+20] Stéphane D'Ascoli, Maria Refinetti, Giulio Biroli, and Florent Krzakala. "Double Trouble in Double Descent: Bias and Variance(s) in the Lazy Regime". In: *ICML*. 2020 (cit. on p. 16).

[DeG+21] Alex J DeGrave, Joseph D Janizek, and Su-In Lee. "AI for radiographic COVID-19 detection selects shortcuts over signal". In: *Nature Machine Intelligence* (2021) (cit. on pp. 37, 51, 161).

[Deg+22] Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de las Casas, Craig Donner, Leslie Fritz, Cristian Galperti, Andrea Huber, James Keeling, Maria Tsimpoukelli, Jackie Kay, Antoine Merle, Jean-Marc Moret, Seb Noury, Federico Pesamosca, David Pfau, Olivier Sauter, Cristian Sommariva, Stefano Coda, Basil Duval, Ambrogio Fasoli, Pushmeet Kohli, Koray Kavukcuoglu, Demis Hassabis, and Martin Riedmiller. “Magnetic control of tokamak plasmas through deep reinforcement learning”. In: *Nature* (2022) (cit. on p. 2).

[Deu11] D. Deutsch. “The beginning of infinity: Explanations that transform the world”. In: *Penguin UK* (2011) (cit. on p. 166).

[DeV+17a] Terrance DeVries and Graham W Taylor. “Dataset augmentation in feature space”. In: *arXiv preprint* (2017) (cit. on p. 151).

[DeV+17b] Terrance DeVries and Graham W Taylor. “Improved regularization of convolutional neural networks with cutout”. In: *arXiv preprint* (2017) (cit. on p. 151).

[Dev+19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *NAACL*. 2019 (cit. on p. 63).

[Dew14] Dan Dewey. “Reinforcement Learning and the Reward Engineering Principle”. In: *AAAI Spring Symposia*. 2014 (cit. on p. 74).

[Di +22] Lauro Langosco Di Langosco, Jack Koch, Lee D Sharkey, Jacob Pfau, and David Krueger. “Goal misgeneralization in deep reinforcement learning”. In: *ICML*. 2022 (cit. on p. 82).

[Die+22] Eustache Diemert, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Alexandre Ramé, and Ugo Tanielian. “PRINCE: PRomoting INvariance for Classification of browsing journeys across Environments”. In: *ECML PKDD* (2022) (cit. on p. 7).

[Die00] Thomas Dietterich. “Ensemble methods in machine learning”. In: *MCS*. 2000 (cit. on pp. 5, 19, 21).

[Dim+22] Nikolaos Dimitriadis, Pascal Frossard, and François Fleuret. “Pareto Manifold Learning: Tackling multiple tasks via ensembles of single-task models”. In: *arXiv preprint* (2022) (cit. on p. 75).

[Din+17a] Zhengming Ding and Yun Fu. “Deep domain generalization with structured low-rank constraint”. In: *TIP*. 2017 (cit. on p. 163).

[Din+17b] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. “Sharp Minima Can Generalize For Deep Nets”. In: *ICML*. 2017 (cit. on p. 39).

[Dom00] Pedro Domingos. “A unified bias-variance decomposition”. In: *ICML*. 2000 (cit. on p. 15).

[Don+23a] Shachar Don-Yehiya, Elad Venezian, Colin Raffel, Noam Slonim, Yoav Katz, and Leshem Choshen. "ColD Fusion: Collaborative Descent for Distributed Multitask Finetuning". In: *ACL*. 2023 (cit. on pp. 51, 53, 75, 80).

[Don+23b] Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. "RAFT: Reward rAnked FineTuning for Generative Foundation Model Alignment". In: *arXiv preprint* (2023) (cit. on p. 72).

[Don+75] Monroe D Donsker and SR Srinivasa Varadhan. "Asymptotic evaluation of certain Markov process expectations for large time". In: *Communications on Pure and Applied Mathematics* (1975) (cit. on p. 30).

[Dor+94] Marco Dorigo and Marco Colombetti. "Robot shaping: Developing autonomous agents through learning". In: *Artificial intelligence* (1994) (cit. on p. 74).

[Dos+21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *ICLR*. 2021 (cit. on pp. 3, 11).

[Dou+22] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. "DyTox: Transformers for Continual Learning with DYnamic TOken eXpansion". In: *CVPR*. 2022 (cit. on p. 7).

[Dra+18] Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred Hamprecht. "Essentially No Barriers in Neural Network Energy Landscape". In: *ICML*. 2018 (cit. on pp. 39, 53).

[Du+18] Yunshu Du, Wojciech M Czarnecki, Siddhant M Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. "Adapting auxiliary losses using gradient similarity". In: *arXiv preprint* (2018) (cit. on p. 164).

[Dua+16] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and P. Abbeel. "Benchmarking Deep Reinforcement Learning for Continuous Control". In: *ICML*. 2016 (cit. on p. 74).

[Dul+21] Gabriel Dulac-Arnold, Nir Levine, Daniel J Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. "Challenges of real-world reinforcement learning: definitions, benchmarks and analysis". In: *Machine Learning* (2021) (cit. on p. 63).

[Dur+20] Nikita Durasov, Timur Bagautdinov, Pierre Baque, and Pascal Fua. "Masksembles for Uncertainty Estimation". In: *arXiv preprint* (2020) (cit. on p. 23).

[Dur+23] Esin Durmus, Karina Nyugen, Thomas I. Liao, Nicholas Schiefer, Amanda Askell, Anton Bakhtin, Carol Chen, Zac Hatfield-Dodds, Danny Hernandez, Nicholas Joseph, Liane Lovitt, Sam McCandlish, Orowa Sikder, Alex Tamkin, Janel Thamkul, Jared Kaplan, Jack Clark, and Deep Ganguli. "Towards Measuring the Representation of Subjective Global Opinions in Language Models". In: *arXiv preprint* (2023) (cit. on pp. 64, 131).

[Dus+20] Michael Dusenberry, Ghassen Jerfel, Yeming Wen, Yian Ma, Jasper Snoek, Katherine Heller, Balaji Lakshminarayanan, and Dustin Tran. "Efficient and scalable bayesian neural nets with rank-1 factors". In: *ICML*. 2020 (cit. on p. 23).

[Dvo+19] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. "Diversity with cooperation: Ensemble methods for few-shot classification". In: *ICCV*. 2019 (cit. on p. 22).

[Eec+22] Steven Vander Eeckt et al. "Weight Averaging: A Simple Yet Effective Method to Overcome Catastrophic Forgetting in Automatic Speech Recognition". In: *arXiv preprint* (2022) (cit. on p. 75).

[Efr+94] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. 1994 (cit. on p. 22).

[Ent+22] Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. "The Role of Permutation Invariance in Linear Mode Connectivity of Neural Networks". In: *ICLR*. 2022 (cit. on p. 50).

[Eva+16] Richard Evans, Jim Gao, and Deepmind. *DeepMind AI Reduces Google Data Centre Cooling Bill by 40%*. https://deepmind.com/blog/article/deepmind-ai-reduces-google-data-centre-cooling-bill-40. 2016 (cit. on p. 2).

[Eve+18] Tom Everitt, Gary Lea, and Marcus Hutter. "AGI safety literature review". In: *arXiv preprint* (2018) (cit. on p. 81).

[Fag+20] Fartash Faghri, David Duvenaud, David J Fleet, and Jimmy Ba. "A Study of Gradient Variance in Deep Learning". In: *arXiv preprint* (2020) (cit. on p. 169).

[Fan+13] Chen Fang, Ye Xu, and Daniel N Rockmore. "Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias". In: *ICCV*. 2013 (cit. on pp. 14, 45).

[Fan+22] Alex Fang, Gabriel Ilharco, Mitchell Wortsman, Yuhao Wan, Vaishaal Shankar, Achal Dave, and Ludwig Schmidt. "Data Determines Distributional Robustness in Contrastive Language Image Pre-training (CLIP)". In: *ICML*. 2022 (cit. on p. 37).

[Far+20] Mojtaba Faramarzi, Mohammad Amini, Akilesh Badrinaaraayanan, Vikas Verma, and Sarath Chandar. "PatchUp: A Regularization Technique for Convolutional Neural Networks". In: *arXiv preprint* (2020) (cit. on p. 151).

[Fin+17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks”. In: *ICML*. 2017 (cit. on p. 164).

[Fis+20] Ian Fischer and Alexander A Alemi. “CEB Improves Model Robustness”. In: *arXiv preprint* (2020) (cit. on pp. 28, 29).

[Fis20] Ian Fischer. “The Conditional Entropy Bottleneck”. In: *arXiv preprint* (2020) (cit. on pp. 25, 27–29, 32, 33, 141).

[Fis22] Ronald A Fisher. “On the mathematical foundations of theoretical statistics”. In: *Philosophical Transactions of the Royal Society of London.* (1922) (cit. on pp. 147, 162, 168).

[For+19a] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. “Deep ensembles: A loss landscape perspective”. In: *arXiv preprint* (2019) (cit. on pp. 22, 38).

[For+19b] Stanislav Fort, Paweł Krzysztof Nowak, Stanislaw Jastrzebski, and Srini Narayanan. “Stiffness: A new perspective on generalization in neural networks”. In: *arXiv preprint* (2019) (cit. on p. 161).

[For+21] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. “Sharpness-aware Minimization for Efficiently Improving Generalization”. In: *ICLR*. 2021 (cit. on pp. 40, 167).

[Fra+19] Jonathan Frankle and Michael Carbin. “The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks”. In: *ICLR*. 2019 (cit. on pp. 23, 149).

[Fra+20] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. “Linear Mode Connectivity and the Lottery Ticket Hypothesis”. In: *ICML*. 2020 (cit. on pp. 38, 39, 42, 44, 52, 65).

[Fra+21] Elias Frantar, Eldar Kurtic, and Dan Alistarh. “Efficient Matrix-Free Approximations of Second-Order Information, with Applications to Pruning and Optimization”. In: *arXiv preprint* (2021) (cit. on p. 168).

[Fre+20] Geoff French, Avital Oliver, and Tim Salimans. “Milking CowMask for Semi-Supervised Image Classification”. In: *arXiv preprint* (2020) (cit. on p. 149).

[Fre+21] C Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. “Brax–A Differentiable Physics Engine for Large Scale Rigid Body Simulation”. In: *arXiv preprint* (2021) (cit. on p. 74).

[Fri01] Jerome H Friedman. “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics* (2001) (cit. on p. 22).

[Fuk80] Kunihiko Fukushima. “Neocognitron: A self-organizing neural network for a mechanism of pattern recognition unaffected by shift in position”. In: *Biological Cybernetics*. 1980 (cit. on pp. 3, 11).

[Gab+21] Iason Gabriel and Vafa Ghazavi. “The challenge of value alignment: From fairer algorithms to AI safety”. In: *arXiv preprint* (2021) (cit. on p. 131).

[Gal+16] Yarin Gal and Zoubin Ghahramani. “Dropout as a bayesian approximation: Representing model uncertainty in deep learning”. In: *ICML*. 2016 (cit. on pp. 3, 23).

[Gan+16] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. “Domain-adversarial training of neural networks”. In: *JMLR* (2016) (cit. on pp. 161, 163).

[Gan+22] Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. “Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned”. In: *arXiv preprint* (2022) (cit. on p. 63).

[Gao+18] Weihao Gao, Sewoong Oh, and Pramod Viswanath. “Demystifying Fixed k-Nearest Neighbor Information Estimators”. In: *Transactions on Information Theory* (2018) (cit. on p. 30).

[Gao+19] Yuan Gao, Zixiang Cai, and Lei Yu. “Intra-Ensemble in Neural Networks”. In: *arXiv preprint* (2019) (cit. on p. 23).

[Gao+22] Leo Gao, John Schulman, and Jacob Hilton. “Scaling Laws for Reward Model Overoptimization”. In: *arXiv preprint* (2022) (cit. on p. 82).

[Gay+22] Jean-Baptiste Gaya, Laure Soulier, and Ludovic Denoyer. “Learning a Subspace of Policies for Online Adaptation in Reinforcement Learning”. In: *ICLR*. 2022 (cit. on p. 75).

[Gei+20] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. “Shortcut learning in deep neural networks”. In: *Nature Machine Intelligence* (2020) (cit. on p. 161).

[Gho+19] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. “An Investigation into Neural Net Optimization via Hessian Eigenvalue Density”. In: *ICML*. 2019 (cit. on p. 166).

[Gho+21] Benyamin Ghojogh, Ali Ghodsi, Fakhri Karray, and Mark Crowley. “Reproducing Kernel Hilbert Space, Mercer’s Theorem, Eigenfunctions, Nystrom Method, and Use of Kernels in Machine Learning: Tutorial and Survey”. In: *arXiv preprint* (2021) (cit. on pp. 17, 134).

[Gla+22] Amelia Glaese, Nat McAleese, Maja Trebacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. “Improving alignment of dialogue agents via targeted human judgements”. In: *arXiv preprint* (2022) (cit. on p. 75).

[Gol23] Yoav Goldberg. *Reinforcement Learning for Language Models*. https://gist.github.com/yoavg/6bff0fecd65950898eba1bb321cfbd81. 2023 (cit. on p. 63).

[Gon+16] Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. "Domain Adaptation with Conditional Transferable Components". In: *ICML*. 2016 (cit. on p. 163).

[Gon+22] Raphael Gontijo-Lopes, Yann Dauphin, and Ekin Dogus Cubuk. "No One Representation to Rule Them All: Overlapping Features of Training Methods". In: *ICLR*. 2022 (cit. on pp. 38, 58).

[Goo+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets". In: *NeurIPS*. 2014 (cit. on pp. 26, 31).

[Goo+16] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016 (cit. on p. 2).

[Gou+23] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. "Critic: Large language models can self-correct with tool-interactive critiquing". In: *arXiv preprint* (2023) (cit. on p. 80).

[Gre+12] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. "A Kernel Two-Sample Test". In: *JMLR* (2012) (cit. on p. 135).

[Gro+19] Patrick Grother, Mei Ngan, and Kayee Hanaoka. *Face Recognition Vendor Test: Part 3: Demographic Effects*. 2019 (cit. on p. 4).

[Gul+21] Ishaan Gulrajani and David Lopez-Paz. "In Search of Lost Domain Generalization". In: *ICLR*. 2021 (cit. on pp. 3, 14, 34, 35, 37–39, 45, 46, 49, 51, 52, 56, 57, 161, 163, 171).

[Guo+17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. "On Calibration of Modern Neural Networks". In: *ICML*. 2017 (cit. on pp. 79, 151, 155).

[Guo+20] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. "Online Knowledge Distillation via Collaborative Learning". In: *CVPR*. 2020 (cit. on p. 22).

[Guo+21] Ruocheng Guo, Pengchuan Zhang, Hao Liu, and Emre Kiciman. "Out-of-distribution Prediction with Invariant Risk Minimization: The Limitation and An Effective Fix". In: *arXiv preprint* (2021) (cit. on p. 163).

[Gup+20] Vipul Gupta, Santiago Akle Serrano, and Dennis DeCoste. "Stochastic Weight Averaging in Parallel: Large-Batch Training That Generalizes Well". In: *ICLR*. 2020 (cit. on p. 45).

[Han+90] Lars Kai Hansen and Peter Salamon. "Neural network ensembles". In: *TPAMI* (1990) (cit. on pp. 5, 19, 71).

[Har+20] Ethan Harris, Antonia Marcu, Matthew Painter, Mahesan Niranjan, Adam Prügel-Bennett, and Jonathon Hare. "FMix: Enhancing Mixed Sample Data Augmentation". In: *arXiv preprint* (2020) (cit. on pp. 149, 151).

[Hav+21] Marton Havasi, Rodolphe Jenatton, Stanislav Fort, Jeremiah Liu, Jasper Roland Snoek, Balaji Lakshminarayanan, Andrew Mingbo Dai, and Dustin Tran. “Training independent subnetworks for robust prediction”. In: *ICLR*. 2021 (cit. on pp. 23, 149, 150, 152, 155, 157, 158).

[Hay+22] Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. “A practical guide to multi-objective reinforcement learning and planning”. In: *JAAMAS* (2022) (cit. on pp. 64, 65, 75).

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *CVPR*. 2016 (cit. on pp. 11, 14).

[He+19] Zhuoxun He, Lingxi Xie, Xin Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. “Data augmentation revisited: Rethinking the distribution gap between clean and augmented data”. In: *arXiv preprint* (2019) (cit. on p. 151).

[He+20] Hangfeng He and Weijie Su. “The Local Elasticity of Neural Networks”. In: *ICLR*. 2020 (cit. on pp. 17, 134).

[Hen+19a] Dan Hendrycks and Thomas Dietterich. “Benchmarking Neural Network Robustness to Common Corruptions and Perturbations”. In: *ICLR*. 2019 (cit. on pp. 37, 149, 158).

[Hen+19b] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. “AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty”. In: *ICLR*. 2019 (cit. on pp. 151, 158).

[Hen+21] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. “The many faces of robustness: A critical analysis of out-of-distribution generalization”. In: *ICCV*. 2021 (cit. on p. 37).

[Hen+22] Dan Hendrycks and Mantas Mazeika. “X-risk analysis for AI research”. In: *arXiv preprint* (2022) (cit. on pp. 4, 82).

[Hen+23] Dan Hendrycks, Mantas Mazeika, and Thomas Woodside. “An Overview of Catastrophic AI Risks”. In: *arXiv preprint* (2023) (cit. on p. 82).

[Hen23] Dan Hendrycks. “Natural Selection Favors AIs over Humans”. In: *arXiv preprint* (2023) (cit. on pp. 4, 82).

[Hin20] David Hin. “StackOverflow vs Kaggle: A Study of Developer Discussions About Data Science”. In: *arXiv preprint* (2020) (cit. on p. 5).

[Hje+19] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. “Learning deep representations by mutual information estimation and maximization”. In: *ICLR*. 2019 (cit. on p. 26).

[Ho+20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models”. In: *NeurIPS* (2020) (cit. on p. 72).

[Ho95] Tin Kam Ho. “Random decision forests”. In: *ICDAR*. 1995 (cit. on p. 19).

[Hoc+01] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. *Gradient flow in recurrent nets: the difficulty of learning long-term dependencies*. Ed. by IEEE Press. Kremer, S. C.; Kolen, J. F. (eds.), 2001 (cit. on p. 11).

[Hof+20] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. “Augment Your Batch: Improving Generalization Through Instance Repetition”. In: *CVPR*. 2020 (cit. on p. 155).

[Hof+22] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. “Training compute-optimal large language models”. In: *NeurIPS*. 2022 (cit. on p. 2).

[Hon+04] Lu Hong and Scott E Page. “Groups of diverse problem solvers can outperform groups of high-ability problem solvers”. In: *Proceedings of the National Academy of Sciences* (2004) (cit. on p. 132).

[Hu+22a] Hu, Roberto Cavicchioli, and Alessandro Capotondi. “ExpansionNet v2: Block Static Expansion in fast end to end training for Image Captioning”. In: *arXiv preprint* (2022) (cit. on pp. 71, 72).

[Hu+22b] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. “LoRA: Low-Rank Adaptation of Large Language Models”. In: *ICLR*. 2022 (cit. on p. 69).

[Hua+17] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q Weinberger. “Snapshot Ensembles: Train 1, get M for free”. In: *ICLR*. 2017 (cit. on p. 24).

[Hua+23] Pu Hua, Yubei Chen, and Huazhe Xu. “Simple Emergent Action Representations from Multi-Task Policy Training”. In: *ICLR*. 2023 (cit. on p. 75).

[Idn+20] Daksh Idnani and Jonathan C Kao. “Learning Robust Representations with Score Invariant Learning”. In: *ICML UDL Workshop*. 2020 (cit. on p. 163).

[Idr+22] Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. “Simple data balancing achieves competitive worst-group-accuracy”. In: *CCLR*. 2022 (cit. on p. 19).

[Ilh+22] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. “Patching open-vocabulary models by interpolating weights”. In: *NeurIPS*. 2022 (cit. on pp. 53, 75).

[Ilh+23] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. “Editing Models with Task Arithmetic”. In: *ICLR*. 2023 (cit. on pp. 53, 79).

[Ino18] Hiroshi Inoue. “Data augmentation by pairing samples for images classification”. In: *arXiv preprint* (2018) (cit. on p. 151).

[Iof+15] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *ICML*. 2015 (cit. on p. 11).

[Irv+18] Geoffrey Irving, Paul Christiano, and Dario Amodei. "AI safety via debate". In: *arXiv preprint* (2018) (cit. on p. 80).

[Irv+23] Robert Irvine, Douglas Boubert, Vyas Raina, Adian Liusie, Vineet Mudupalli, Aliaksei Korshuk, Zongyi Liu, Fritz Cremer, Valentin Assassi, Christie-Carol Beauchamp, et al. "Rewarding Chatbots for Real-World Engagement with Millions of Users". In: *arXiv preprint* (2023) (cit. on p. 64).

[Izm+18] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. "Averaging weights leads to wider optima and better generalization". In: *UAI*. 2018 (cit. on pp. 24, 37, 39, 41, 43, 45, 51–53, 60, 78, 79, 167).

[Izm+19] Pavel Izmailov, Wesley Maddox, Polina Kirichenko, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. "Subspace Inference for Bayesian Deep Learning". In: *UAI*. 2019 (cit. on p. 45).

[Jac+18] Arthur Jacot, Franck Gabriel, and Clement Hongler. "Neural Tangent Kernel: Convergence and Generalization in Neural Networks". In: *NeurIPS*. 2018 (cit. on pp. 17, 134).

[Jad+17] Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. "Population based training of neural networks". In: *arXiv preprint* (2017) (cit. on p. 75).

[Jad+19] Max Jaderberg, Wojciech M Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castaneda, Charles Beattie, Neil C Rabinowitz, Ari S Morcos, Avraham Ruderman, et al. "Human-level performance in 3D multiplayer games with population-based reinforcement learning". In: *Science* (2019) (cit. on p. 75).

[Jan+23] Joel Jang, Seungone Kim, Seonghyeon Ye, Doyoung Kim, Lajanugen Logeswaran, Moontae Lee, Kyungjae Lee, and Minjoon Seo. "Exploring the Benefits of Training Expert Language Models over Instruction Tuning". In: *arXiv preprint* (2023) (cit. on p. 79).

[Jan23] Ilya Sutskever Jan Leike. *Introducing Superalignment*. https://openai.com/blog/introducing-superalignment. 2023 (cit. on p. 81).

[Jas+18] Stanisław Jastrzebski, Zac Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Amos Storkey, and Yoshua Bengio. "Three factors influencing minima in SGD". In: *ICANN*. 2018 (cit. on p. 168).

[Jas+21] Stanislaw Jastrzebski, Devansh Arpit, Oliver Astrand, Giancarlo B Kerg, Huan Wang, Caiming Xiong, Richard Socher, Kyunghyun Cho, and Krzysztof J Geras. "Catastrophic fisher explosion: Early phase fisher matrix impacts generalization". In: *ICML*. 2021 (cit. on p. 169).

[Joh+19] Fredrik D Johansson, David Sontag, and Rajesh Ranganath. "Support and invertibility in domain-invariant representations". In: *AISTATS*. 2019 (cit. on pp. 163, 167).

[Joh+22] Rebecca L Johnson, Giada Pistilli, Natalia Menédez-González, Leslye Denisse Dias Duran, Enrico Panai, Julija Kalpokiene, and Donald Jay Bertulfo. "The Ghost in the Machine has an American accent: value conflict in GPT-3". In: *arXiv preprint* (2022) (cit. on p. 64).

[Ju+18] Cheng Ju, Aurélien Bibaut, and Mark van der Laan. "The relative performance of ensemble methods with deep convolutional neural networks for image classification". In: *Journal of Applied Statistics* (2018) (cit. on p. 19).

[Jum+21] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* (2021) (cit. on p. 2).

[Jun+23] Jeevesh Juneja, Rachit Bansal, Kyunghyun Cho, João Sedoc, and Naomi Saphra. "Linear Connectivity Reveals Generalization Strategies". In: *ICLR*. 2023 (cit. on p. 60).

[Kad+22] Jean Kaddour, Linqing Liu, Ricardo Silva, and Matt Kusner. "When Do Flat Minima Optimizers Work?" In: *NeurIPS*. 2022 (cit. on p. 40).

[Kad22] Jean Kaddour. "Stop Wasting My Time! Saving Days of ImageNet and BERT Training with Latest Weight Averaging". In: *NeurIPS Workshop*. 2022 (cit. on p. 39).

[Kam+21] Pritish Kamath, Akilesh Tangella, Danica Sutherland, and Nathan Srebro. "Does Invariant Risk Minimization Capture Invariance?" In: *AISTATS*. 2021 (cit. on p. 163).

[Kan+15] Kirthevasan Kandasamy, Akshay Krishnamurthy, Barnabas Poczos, Larry Wasserman, et al. "Nonparametric von mises estimators for entropies, divergences and mutual informations". In: *NeurIPS*. 2015 (cit. on p. 30).

[Kan+23] Nikhil Kandpal, Brian Lester, Mohammed Muqeeth, Anisha Mascarenhas, Monty Evans, Vishal Baskaran, Tenghao Huang, Haokun Liu, and Colin Raffel. "Git-Theta: A Git Extension for Collaborative Development of Machine Learning Models". In: 2023 (cit. on p. 80).

[Kap+20] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. "Scaling laws for neural language models". In: *arXiv preprint* (2020) (cit. on p. 2).

[Kar+15] Andrej Karpathy and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions". In: *CVPR*. 2015 (cit. on p. 72).

[Kar+19] Sanjay Kariyappa and Moinuddin K. Qureshi. "Improving Adversarial Robustness of Ensembles with Diversity Training". In: *arXiv preprint* (2019) (cit. on p. 22).

[Kar17] Andrej Karpathy. *Software 2.0*. https://karpathy.medium.com/software-2-0-a64152b37c35. 2017 (cit. on p. 80).

[Kaw+17] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. "Generalization in deep learning". In: *arXiv preprint* (2017) (cit. on pp. 3, 13).

[Kem+20] Mete Kemertas, Leila Pishdad, Konstantinos G. Derpanis, and Afsaneh Fazly. "RankMI: A Mutual Information Maximizing Ranking Loss". In: *CVPR*. 2020 (cit. on p. 26).

[Ken+21] Zachary Kenton, Tom Everitt, Laura Weidinger, Iason Gabriel, Vladimir Mikulik, and Geoffrey Irving. "Alignment of language agents". In: *arXiv preprint* (2021) (cit. on pp. 4, 82).

[Ker+21] Corentin Kervadec, Grigory Antipov, Moez Baccouche, and Christian Wolf. "Roses are red, violets are blue... but should vqa expect them to?" In: *CVPR*. 2021 (cit. on p. 161).

[Kim+18] Wonsik Kim, Bhavya Goyal, Kunal Chawla, Jungmin Lee, and Keunjoo Kwon. "Attention-based ensemble for deep metric learning". In: *ECCV*. 2018 (cit. on p. 32).

[Kim+19] Hyoungseok Kim, Jaekyeom Kim, Yeonwoo Jeong, Sergey Levine, and Hyun Oh Song. "EMI: Exploration with Mutual Information". In: *ICML*. 2019 (cit. on p. 26).

[Kim+20] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. "Puzzle mix: Exploiting saliency and local statistics for optimal mixup". In: *ICML*. 2020 (cit. on p. 149).

[Kin+15] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *ICLR*. 2015 (cit. on pp. 3, 12, 14, 144, 169).

[Kir+17] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. "Overcoming catastrophic forgetting in neural networks". In: *PNAS*. 2017 (cit. on p. 169).

[Kir+22] Andreas Christian Kirsch, Balaji Lakshminarayanan, Clara Huiyi Hu, D. Sculley, Du Phan, Dustin Tran, Jasper Roland Snoek, Jeremiah Liu, Jie Jessie Ren, Joost van Amersfoort, Kehang Han, Kelly Buchanan, Kevin Patrick Murphy, Mark Patrick Collier, Michael W. Dusenberry, Neil Band, Nithum Thain, Rodolphe Jenatton, Tim G. J. Rudner, Yarin Gal, Zachary Nado, Zelda Mariet, Zi Wang, and Zoubin Ghahramani. "Plex: Towards Reliability using Pretrained Large Model Extensions". In: *ICML Workshop*. 2022 (cit. on pp. 38, 53).

[Kir+23] Hannah Rose Kirk, Bertie Vidgen, Paul Röttger, and Scott A Hale. "Personalisation within bounds: A risk taxonomy and policy framework for the alignment of large language models with personalised feedback". In: *arXiv preprint* (2023) (cit. on pp. 64, 75, 76, 131).

[Koh+21] 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 Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. "WILDS: A Benchmark of in-the-Wild Distribution Shifts". In: *ICML*. 2021 (cit. on pp. 59, 163).

[Koh+96] Ron Kohavi, David H Wolpert, et al. "Bias plus variance decomposition for zero-one loss functions". In: *ICML*. 1996 (cit. on pp. 15, 21, 139).

[Kol+91] John F Kolen and Jordan B Pollack. "Back propagation is sensitive to initial conditions". In: *NeurIPS*. 1991 (cit. on p. 22).

[Köp+23] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. "OpenAssistant Conversations–Democratizing Large Language Model Alignment". In: *arXiv preprint* (2023) (cit. on p. 69).

[Kor+19] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. "Similarity of Neural Network Representations Revisited". In: *ICML*. 2019 (cit. on pp. 21, 48).

[Kov+23] Grgur Kovač, Masataka Sawayama, Rémy Portelas, Cédric Colas, Peter Ford Dominey, and Pierre-Yves Oudeyer. "Large Language Models as Superpositions of Cultural Perspectives". In: *arXiv preprint* (2023) (cit. on p. 64).

[Koy+20] Masanori Koyama and Shoichiro Yamaguchi. "Out-of-distribution generalization with maximal invariant predictor". In: *arXiv preprint* (2020) (cit. on p. 164).

[Kra+04] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. "Estimating mutual information". In: *Physical review E* (2004) (cit. on p. 30).

[Kri+09] Alex Krizhevsky and Geoffrey Hinton. *Learning multiple layers of features from tiny images*. Tech. rep. 2009 (cit. on pp. 14, 32, 154).

[Kri+12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *NeurIPS*. 2012 (cit. on pp. 2, 3, 44, 161).

[Kro+91] Anders Krogh and John Hertz. “A simple weight decay can improve generalization”. In: *NeurIPS*. 1991 (cit. on p. 3).

[Kro+95] Anders Krogh and Jesper Vedelsby. “Neural network ensembles, cross validation, and active learning”. In: *NeurIPS*. 1995 (cit. on p. 19).

[Kru+21] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. “Out-of-Distribution Generalization via Risk Extrapolation (REx)”. In: *ICML*. 2021 (cit. on pp. 161, 164, 166, 167, 169).

[Kul+20] Apoorv Kulshreshtha, Daniel De Freitas Adiwardana, David Richard So, Gaurav Nemade, Jamie Hall, Noah Fiedel, Quoc V Le, Romal Thoppilan, Thang Luong, Yifeng Lu, et al. “Towards a human-like open-domain chatbot”. In: (2020) (cit. on p. 83).

[Kul59] Solomon Kullback. *Information Theory and Statistics*. New York, 1959 (cit. on p. 141).

[Kum+22] Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. “Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution”. In: *ICLR*. 2022 (cit. on pp. 44, 46, 53, 55, 60).

[Kun+03] Ludmila I Kuncheva and Christopher J Whitaker. “Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy”. In: *Machine learning* (2003) (cit. on pp. 21, 53, 58, 61).

[Kun+19] Frederik Kunstner, Philipp Hennig, and Lukas Balles. “Limitations of the empirical Fisher approximation for natural gradient descent”. In: *NeurIPS*. 2019 (cit. on pp. 147, 169).

[Kwo+23] Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. “Reward Design with Language Models”. In: *ICLR*. 2023 (cit. on pp. 63, 80).

[Lak+17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. “Simple and scalable predictive uncertainty estimation using deep ensembles”. In: *NeurIPS*. 2017 (cit. on pp. 19, 22, 25, 32, 33, 45, 71, 155).

[Lam+22] Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. “Can language models learn from explanations in context?” In: *EMNLP*. 2022 (cit. on p. 3).

[Lam+23] Nathan Lambert, Lewis Tunstall, Nazneen Rajani, and Tristan Thrush. *HuggingFace H4 Stack Exchange Preference Dataset*. 2023. URL: https://huggingface.co/datasets/HuggingFaceH4/stack-exchange-preferences (cit. on p. 69).

[Lan+18] Xu Lan, Xiatian Zhu, and Shaogang Gong. “Knowledge distillation by on-the-fly native ensemble”. In: *NeurIPS*. 2018 (cit. on pp. 22, 23, 32, 33).

[Lar+20] Agostina J. Larrazabal, Nicolás Nieto, Victoria Peterson, Diego H. Milone, and Enzo Ferrante. "Gender imbalance in medical imaging datasets produces biased classifiers for computer-aided diagnosis". In: *PNAS*. 2020 (cit. on p. 4).

[Law+23] Daniel Lawson and Ahmed H Qureshi. "Merging Decision Transformers: Weight Averaging for Forming Multi-Task Policies". In: *ICLR RRL Workshop*. 2023 (cit. on p. 75).

[LeC+10] Yann LeCun, Corinna Cortes, and Chris Burges. *MNIST handwritten digit database*. 2010 (cit. on pp. 14, 49).

[LeC+12] Yann LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. "Efficient BackProp." In: *Neural Networks*. 2012 (cit. on pp. 144, 169).

[LeC+90] Yann LeCun, J. S. Denker, Sara A. Solla, R. E. Howard, and L.D. Jackel. "Optimal brain damage". In: *NeurIPS*. 1990 (cit. on pp. 144, 169).

[LeC+99] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning". In: *Shape, contour and grouping in computer vision*. 1999 (cit. on p. 12).

[LeC22] Yann LeCun. "A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27". In: *Open Review* (2022) (cit. on p. 83).

[Lee+15] Stefan Lee, Senthil Purushwalkam, Michael Cogswell, David J. Crandall, and Dhruv Batra. "Why M Heads are Better than One: Training a Diverse Ensemble of Deep Networks". In: *arXiv preprint* (2015) (cit. on pp. 5, 23, 33, 35).

[Lee+16] Stefan Lee, Senthil Purushwalkam Shiva Prakash, Michael Cogswell, Viresh Ranjan, David Crandall, and Dhruv Batra. "Stochastic multiple choice learning for training diverse deep ensembles". In: *NeurIPS*. 2016 (cit. on p. 22).

[Lee+17] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. "Deep neural networks as gaussian processes". In: *ICLR*. 2017 (cit. on pp. 17, 134).

[Lee+23] Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. "Aligning Text-to-Image Models using Human Feedback". In: *arXiv preprint* (2023) (cit. on pp. 63, 72).

[Li+17] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. "Deeper, broader and artier domain generalization". In: *ICCV*. 2017 (cit. on pp. 13, 14, 45).

[Li+18a] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. "Domain generalization with adversarial feature learning". In: *CVPR*. 2018 (cit. on p. 163).

[Li+18b] Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. "Domain generalization via conditional invariant representations". In: *AAAI*. 2018 (cit. on p. 163).

[Li+19] Qinbin Li, Zeyi Wen, and Bingsheng He. "Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection." In: *arXiv preprint* (2019) (cit. on p. 80).

[Li+20a] Boyi Li, Felix Wu, Ser-Nam Lim, Serge Belongie, and Kilian Q. Weinberger. "On Feature Normalization and Data Augmentation". In: *arXiv preprint* (2020) (cit. on p. 151).

[Li+20b] Kaiwen Li, Tao Zhang, and Rui Wang. "Deep reinforcement learning for multiobjective optimization". In: *IEEE-T-CYBERNETICS* (2020) (cit. on pp. 64, 66, 75).

[Li+20c] Xinyan Li, Qilong Gu, Yingxue Zhou, Tiancong Chen, and Arindam Banerjee. "Hessian based analysis of sgd for deep nets: Dynamics and generalization". In: *SIAM*. 2020 (cit. on p. 168).

[Li+21] Wei-Hong Li, Xialei Liu, and Hakan Bilen. "Universal Representation Learning From Multiple Domains for Few-Shot Classification". In: *ICCV*. 2021 (cit. on pp. 53, 58).

[Li+22] Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A Smith, and Luke Zettlemoyer. "Branch-Train-Merge: Embarrassingly Parallel Training of Expert Language Models". In: *arXiv preprint* (2022) (cit. on pp. 53, 60, 79, 80).

[Lia+18] Daojun Liang, Feng Yang, Tian Zhang, and Peter Yang. "Understanding mixup training methods". In: *IEEE Access* (2018) (cit. on p. 151).

[Lif23] Future of Life Institute. *Pause Giant AI Experiments: An Open Letter*. https://futureoflife.org/open-letter/pause-giant-ai-experiments/. 2023 (cit. on p. 81).

[Lig+23] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. "Let's Verify Step by Step". In: *arXiv preprint* (2023) (cit. on p. 80).

[Lin+03] Chin-Yew Lin and Eduard Hovy. "Automatic evaluation of summaries using n-gram co-occurrence statistics". In: *NAACL*. 2003 (cit. on pp. 63, 64, 70).

[Lin+14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft coco: Common objects in context". In: *ECCV*. 2014 (cit. on p. 71).

[Lin+17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal loss for dense object detection". In: *ICCV*. 2017 (cit. on p. 154).

[Lin+22] Xi Lin, Zhiyuan Yang, Xiaoyuan Zhang, and Qingfu Zhang. "Pareto Set Learning for Expensive Multi-Objective Optimization". In: *NeuriPS*. 2022 (cit. on p. 75).

[Liu+21] Liyang Liu, Shilong Zhang, Zhanghui Kuang, Aojun Zhou, Jing-Hao Xue, Xinjiang Wang, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. "Group Fisher Pruning for Practical Network Compression". In: *ICML*. 2021 (cit. on p. 168).

[Liu+22] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. "Swin Transformer V2: Scaling Up Capacity and Resolution". In: *CVPR*. 2022 (cit. on p. 71).

[Liu+23] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. "Llm+ p: Empowering large language models with optimal planning proficiency". In: *arXiv preprint* (2023) (cit. on p. 80).

[Liu+99a] Yong Liu and Xin Yao. "Ensemble learning via negative correlation". In: *Neural networks* (1999) (cit. on p. 22).

[Liu+99b] Yong Liu and Xin Yao. "Simultaneous training of negatively correlated neural networks in an ensemble". In: *IEEE Cybernetics* (1999) (cit. on p. 22).

[Lob+20] Ekaterina Lobacheva, Nadezhda Chirkova, Maxim Kodryan, and Dmitry P Vetrov. "On Power Laws in Deep Ensembles". In: *NeurIPS*. 2020 (cit. on p. 23).

[Lon+14] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. "Transfer joint matching for unsupervised domain adaptation". In: *CVPR*. 2014 (cit. on p. 163).

[Lop+17] David Lopez-Paz and Marc'Aurelio Ranzato. "Gradient Episodic Memory for Continual Learning". In: *NeurIPS*. 2017 (cit. on p. 164).

[Lop+20] Raphael Gontijo Lopes, Sylvia J. Smullin, Ekin D. Cubuk, and Ethan Dyer. "Affinity and Diversity: Quantifying Mechanisms of Data Augmentation". In: *CoRR*. 2020 (cit. on pp. 149, 155).

[Lop+22] David Lopez-Paz, Diane Bouchacourt, Levent Sagun, and Nicolas Usunier. "Measuring and signing fairness as performance under multiple stakeholder distributions". In: *arXiv preprint* (2022) (cit. on pp. 63, 79).

[Lub+22] Ekdeep Singh Lubana, Eric J Bigelow, Robert Dick, David Krueger, and Hidenori Tanaka. "Mechanistic Lens on Mode Connectivity". In: *NeurIPS Workshop*. 2022 (cit. on p. 60).

[Mad+19] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. "A simple baseline for bayesian uncertainty in deep learning". In: *NeurIPS*. 2019 (cit. on p. 45).

[Mad+23] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. "Self-refine: Iterative refinement with self-feedback". In: *arXiv preprint* (2023) (cit. on p. 80).

[Mag+19] Jan R Magnus and Heinz Neudecker. *Matrix differential calculus with applications in statistics and econometrics*. John Wiley & Sons, 2019 (cit. on p. 136).

[Man+18] Massimiliano Mancini, Samuel Rota Bulo, Barbara Caputo, and Elisa Ricci. "Best sources forward: domain generalization through source-specific nets". In: *ICIP*. 2018 (cit. on p. 163).

[Man+21a] Patrick Mannion, Fredrik Heintz, Thommen George Karimpanal, and Peter Vamplew. “Multi-objective decision making for trustworthy ai”. In: *MODeM Workshop*. 2021 (cit. on p. 65).

[Man+21b] Lucas Mansilla, Rodrigo Echeveste, Diego H. Milone, and Enzo Ferrante. “Domain Generalization via Gradient Surgery”. In: *ICCV*. 2021 (cit. on p. 164).

[Mar+10] Sébastien Marcel and Yann Rodriguez. “Torchvision the Machine-Vision Package of Torch”. In: *ACM*. 2010 (cit. on p. 51).

[Mar+23] Daniel Marta, Simon Holk, Christian Pek, Jana Tumova, and Iolanda Leite. “Aligning Human Preferences with Baseline Objectives in Reinforcement Learning”. In: *ICRA*. 2023 (cit. on pp. 64, 75).

[Mar14] James Martens. “New insights and perspectives on the natural gradient method”. In: *arXiv preprint* (2014) (cit. on pp. 168, 169).

[Mas20] Andres R. Masegosa. “Learning under Model Misspecification: Applications to Variational and Ensemble methods”. In: *NeurIPS*. 2020 (cit. on pp. 22, 25).

[Mat+22] Michael Matena and Colin Raffel. “Merging Models with Fisher-Weighted Averaging”. In: *NeurIPS*. 2022 (cit. on pp. 45, 53, 80, 146, 147).

[McM+17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. “Communication-efficient learning of deep networks from decentralized data”. In: *AISTATS*. 2017 (cit. on pp. 76, 131).

[Men+22] Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. “Teaching language models to support answers with verified quotes”. In: *arXiv preprint* (2022) (cit. on p. 80).

[Mic+20] Eric J Michaud, Adam Gleave, and Stuart Russell. “Understanding learned reward functions”. In: *arXiv preprint* (2020) (cit. on p. 63).

[Mil+21] John P Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. “Accuracy on the Line: on the Strong Correlation Between Out-of-Distribution and In-Distribution Generalization”. In: *ICML*. 2021 (cit. on p. 78).

[Mir+21] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Dilan Gorur, Razvan Pascanu, and Hassan Ghasemzadeh. “Linear mode connectivity in multitask and continual learning”. In: *ICLR*. 2021 (cit. on p. 52).

[Mol+17] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. “Pruning convolutional neural networks for resource efficient transfer learning”. In: *ICLR*. 2017 (cit. on p. 23).

[Mon20] Emily Monaco. *The right way to make ratatouille*. https://www.bbc.com/travel/article/20200812-the-right-way-to-make-ratatouille. 2020 (cit. on p. 51).

[Mor+15] Igor Mordatch, Kendall Lowrey, and Emanuel Todorov. "Ensemble-cio: Full-body dynamic motion planning that transfers to physical humanoids". In: *IROS*. 2015 (cit. on p. 75).

[Mos+16] Hossam Mossalam, Yannis M Assael, Diederik M Roijers, and Shimon Whiteson. "Multi-objective deep reinforcement learning". In: *arXiv preprint* (2016) (cit. on p. 75).

[Mua+13] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. "Domain generalization via invariant feature representation". In: *ICML*. 2013 (cit. on pp. 161, 163).

[Muk+20] Sudipto Mukherjee, Himanshu Asnani, and Sreeram Kannan. "CCMI: Classifier based conditional mutual information estimation". In: *UAI*. 2020 (cit. on p. 30).

[Mul16] Ryan Muldoon. *Social contract theory for a diverse world: Beyond tolerance*. 2016 (cit. on p. 132).

[Mur+12] Naila Murray, Luca Marchesotti, and Florent Perronnin. "AVA: A large-scale database for aesthetic visual analysis". In: *CVPR*. 2012 (cit. on p. 72).

[Nad+19] Marcos Nadal and Anjan Chatterjee. "Neuroaesthetics and art's diversity and universality". In: *Wiley Interdisciplinary Reviews: Cognitive Science* (2019) (cit. on p. 63).

[Nag+19] Vaishnavh Nagarajan and J Zico Kolter. "Uniform convergence may be unable to explain generalization in deep learning". In: *NeurIPS* (2019) (cit. on pp. 38, 44).

[Nak+19] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. "Deep Double Descent: Where Bigger Models and More Data Hurt". In: *ICLR*. 2019 (cit. on pp. 16, 22).

[Nak+21] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. "Webgpt: Browser-assisted question-answering with human feedback". In: *arXiv preprint* (2021) (cit. on p. 80).

[Nan+23] Neel Nanda, Lawrence Chan, Tom Liberum, Jess Smith, and Jacob Steinhardt. "Progress measures for grokking via mechanistic interpretability". In: *arXiv preprint* (2023) (cit. on p. 84).

[Ney+20] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. "What is being transferred in transfer learning?" In: *NeurIPS*. 2020 (cit. on pp. 38, 44, 53, 65, 68).

[Ng+99] Andrew Y Ng, Daishi Harada, and Stuart Russell. "Policy invariance under reward transformations: Theory and application to reward shaping". In: *ICML*. 1999 (cit. on p. 74).

[Ngo+22] Richard Ngo, Lawrence Chan, and Soren Mindermann. "The alignment problem from a deep learning perspective". In: *arXiv preprint* (2022) (cit. on pp. 4, 63, 82).

[Ngu+20] Thanh Thi Nguyen, Ngoc Duy Nguyen, Peter Vamplew, Saeid Nahavandi, Richard Dazeley, and Chee Peng Lim. "A multi-objective deep reinforcement learning framework". In: *EAAI* (2020) (cit. on p. 75).

[Ngu+22] Thao Nguyen, Gabriel Ilharco, Mitchell Wortsman, Sewoong Oh, and Ludwig Schmidt. "Quality Not Quantity: On the Interaction between Dataset Design and Robustness of CLIP". In: *NeurIPS*. 2022 (cit. on p. 37).

[Nil65] Nils J. Nilsson. "Learning Machines: Foundations of Trainable Pattern-Classifying Systems". In: 1965 (cit. on pp. 4, 19).

[Nix+20] Jeremy Nixon, Balaji Lakshminarayanan, and Dustin Tran. "Why Are Bootstrapped Deep Ensembles Not Better?" In: *NeurIPS Workshop*. 2020 (cit. on p. 22).

[Ope23] OpenAI. "GPT-4 Technical Report". In: *arXiv preprint* (2023) (cit. on pp. 2, 63, 80).

[Oqu+14] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. "Learning and transferring mid-level image representations using convolutional neural networks". In: *CVPR*. 2014 (cit. on pp. 3, 37, 38, 43, 52, 53, 63, 78).

[Ort+23] Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. "Task Arithmetic in the Tangent Space: Improved Editing of Pre-Trained Models". In: *arXiv preprint* (2023) (cit. on p. 79).

[Osa+22] Takayuki Osa, Voot Tangkaratt, and Masashi Sugiyama. "Discovering diverse solutions in deep reinforcement learning by maximizing state–action-based mutual information". In: *Neural Networks* (2022) (cit. on p. 75).

[Ouy+22] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. "Training language models to follow instructions with human feedback". In: *NeurIPS* (2022) (cit. on pp. 63, 64).

[Ova+19] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift". In: *NeurIPS*. 2019 (cit. on p. 149).

[Ova23] Aviv Ovadya. "Generative CI through Collective Response Systems". In: *arXiv preprint* (2023) (cit. on p. 64).

[Pan+19] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. "Improving Adversarial Robustness via Promoting Ensemble Diversity". In: *ICML*. 2019 (cit. on pp. 22, 25, 32, 33).

[Pan+22] Alexander Pan, Kush Bhatia, and Jacob Steinhardt. "The Effects of Reward Misspecification: Mapping and Mitigating Misaligned Models". In: *ICLR*. 2022 (cit. on p. 63).

[Pan+23] Alexander Pan, Chan Jun Shern, Andy Zou, Nathaniel Li, Steven Basart, Thomas Woodside, Jonathan Ng, Hanlin Zhang, Scott Emmons, and Dan Hendrycks. "Do the Rewards Justify the Means? Measuring Trade-Offs Between Rewards and Ethical Behavior in the MACHIAVELLI Benchmark". In: *ICML*. 2023 (cit. on pp. 64, 81).

[Pap+02] Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. "Bleu: a method for automatic evaluation of machine translation". In: *ACL*. 2002 (cit. on pp. 63, 64, 70).

[Par+20] Jack Parker-Holder, Aldo Pacchiano, Krzysztof M Choromanski, and Stephen J Roberts. "Effective Diversity in Population Based Reinforcement Learning". In: *NeurIPS*. 2020 (cit. on p. 75).

[Par+21] Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. "Learning explanations that are hard to vary". In: *ICLR*. 2021 (cit. on pp. 162, 164–167, 169).

[Par64] Vilfredo Pareto. *Cours d'économie politique*. Librairie Droz, 1964 (cit. on p. 65).

[Pas+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *NeurIPS*. 2019 (cit. on p. 2).

[Pea09] Judea Pearl. *Causality*. Cambridge university press, 2009 (cit. on p. 163).

[Pen+19] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. "Moment matching for multi-source domain adaptation". In: *ICCV*. 2019 (cit. on pp. 14, 45, 49).

[Pér+13] Fernando Pérez-Cruz, Steven Van Vaerenbergh, Juan José Murillo-Fuentes, Miguel Lázaro-Gredilla, and Ignacio Santamaria. "Gaussian processes for nonlinear signal processing: An overview of recent advances". In: *EEE Signal Process. Mag.* (2013) (cit. on p. 16).

[Pes+21] Markus Peschl, Arkady Zgonnikov, Frans A Oliehoek, and Luciano C Siebert. "MORAL: Aligning AI with human norms through multi-objective reinforced active learning". In: *arXiv preprint* (2021) (cit. on p. 75).

[Pet+16] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. "Causal inference by using invariant prediction: identification and confidence intervals". In: *JSTOR* (2016) (cit. on pp. 161, 163).

[Pet+21] Henning Petzka, Michael Kamp, Linara Adilova, Cristian Sminchisescu, and Mario Boley. "Relative Flatness and Generalization". In: *NeurIPS*. 2021 (cit. on p. 39).

[Pha+18] Jason Phang, Thibault Févry, and Samuel R Bowman. "Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks". In: *arXiv preprint* (2018) (cit. on pp. 51–53, 55, 57, 78).

[Pin+23] André Susano Pinto, Alexander Kolesnikov, Yuge Shi, Lucas Beyer, and Xiaohua Zhai. "Tuning computer vision models with task rewards". In: *arXiv preprint* (2023) (cit. on pp. 63, 70).

[Pru+20] Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel Bowman. "Intermediate-Task Transfer Learning with Pretrained Language Models: When and Why Does It Work?" In: *ACL*. 2020 (cit. on pp. 51, 53).

[Qin+20] Jie Qin, Jiemin Fang, Qian Zhang, Wenyu Liu, Xingang Wang, and Xinggang Wang. "ResizeMix: Mixing Data with Preserved Object Information and True Labels". In: *arXiv preprint* (2020) (cit. on p. 159).

[Qin+22] Yujia Qin, Cheng Qian, Jing Yi, Weize Chen, Yankai Lin, Xu Han, Zhiyuan Liu, Maosong Sun, and Jie Zhou. "Exploring Mode Connectivity for Pre-trained Language Models". In: *EMNLP*. 2022 (cit. on p. 60).

[Rad+18] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. "Improving Language Understanding by Generative Pre-Training". In: (2018) (cit. on p. 63).

[Răd+20] Roxana Rădulescu, Patrick Mannion, Diederik M Roijers, and Ann Nowé. "Multi-objective multi-agent decision making: a utility-based analysis and survey". In: *AAMAS* (2020) (cit. on pp. 64, 68, 75).

[Rad+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. "Learning transferable visual models from natural language supervision". In: *ICML*. 2021 (cit. on p. 63).

[Raf23] Colin Raffel. "Building Machine Learning Models Like Open Source Software". In: *ACM* (2023) (cit. on pp. 52, 80, 131).

[Raj+17] Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. "EPOpt: Learning Robust Neural Network Policies Using Model Ensembles". In: *ICLR*. 2017 (cit. on p. 75).

[Ram+21a] Alexandre Ramé and Matthieu Cord. "DICE: Diversity in Deep Ensembles via Conditional Redundancy Adversarial Estimation". In: *ICLR*. 2021 (cit. on pp. 5–7, 14, 26, 27, 31, 34, 35).

[Ram+21b] Alexandre Ramé, Remy Sun, and Matthieu Cord. "MixMo: Mixing Multiple Inputs for Multiple Outputs via Deep Subnetworks". In: *ICCV*. 2021 (cit. on pp. 6, 7, 14, 23, 150).

[Ram+22a] Alexandre Ramé, Corentin Dancette, and Matthieu Cord. "Fishr: Invariant Gradient Variances for Out-of-Distribution Generalization". In: *ICML*. 2022 (cit. on pp. 6, 7, 19, 49, 144, 163, 166, 169, 171).

[Ram+22b] Alexandre Ramé, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. "Diverse Weight Averaging for Out-of-Distribution Generalization". In: *NeurIPS*. 2022 (cit. on pp. 5–7, 9, 38, 43, 49, 51–53, 55, 56, 58, 78).

[Ram+22c] Alexandre Ramé, Jianyu Zhang, Léon Bottou, and David Lopez-Paz. "Pretrain, fine-tune, interpolate: a three-stage strategy for domain generalization". In: *NeurIPS Workshop*. 2022 (cit. on pp. 7, 79).

[Ram+23a] Alexandre Ramé, Kartik Ahuja, Jianyu Zhang, Matthieu Cord, Léon Bottou, and David Lopez-Paz. "Model Ratatouille: Recycling Diverse Models for Out-of-Distribution Generalization". In: *ICML*. 2023 (cit. on pp. 5–7, 52, 56, 57, 75, 78, 131).

[Ram+23b] Alexandre Ramé, Guillaume Couairon, Mustafa Shukor, Corentin Dancette, Jean-Baptiste Gaya, Laure Soulier, and Matthieu Cord. "Rewarded soups: towards Pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards". In: *arXiv preprint* (2023) (cit. on pp. 5–7, 50, 65, 78, 81).

[Ras03] Carl Edward Rasmussen. "Gaussian processes in machine learning". In: *Summer school on machine learning*. 2003 (cit. on p. 134).

[Reb+17] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. "iCaRL: Incremental classifier and representation learning". In: *CVPR*. 2017 (cit. on p. 53).

[Ren+17] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. "Self-critical sequence training for image captioning". In: *CVPR*. 2017 (cit. on pp. 9, 63, 70, 71).

[Ren05] Jason Rennie. "How to normalize a kernel matrix". In: *MIT Computer Science - Artificial Intelligence Lab Tech Rep* (2005) (cit. on pp. 17, 134).

[Ro+23] Yeonju Ro, Zhangyang Wang, Vijay Chidambaram, and Aditya Akella. "Lowering the Pre-training Tax for Gradient-based Subset Training: A Lightweight Distributed Pre-Training Toolkit". In: *arXiv preprint* (2023) (cit. on p. 79).

[Rob+21] Michael Roberts, Derek Driggs, Matthew Thorpe, Julian Gilbey, Michael Yeung, Stephan Ursprung, Angelica I Aviles-Rivero, Christian Etmann, Cathal McCague, Lucian Beer, et al. "Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans". In: *Nature Machine Intelligence* (2021) (cit. on p. 161).

[Roi+13] Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. "A survey of multi-objective sequential decision-making". In: *JAIR* (2013) (cit. on pp. 64, 75).

[Roi+23] Paul Roit, Johan Ferret, Lior Shani, Roee Aharoni, Geoffrey Cideron, Robert Dadashi, Matthieu Geist, Sertan Girgin, Léonard Hussenot, Orgad Keller, et al. "Factually Consistent Summarization via Reinforcement Learning with Textual Entailment Feedback". In: *ACL*. 2023 (cit. on p. 80).

[Roj+18] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. "Invariant models for causal transfer learning". In: *JMLR* (2018) (cit. on p. 163).

[Rok10] Lior Rokach. "Ensemble-based classifiers". In: *Artificial intelligence review* (2010) (cit. on p. 19).

[Rom+22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models". In: *CVPR*. 2022 (cit. on pp. 2, 9, 72).

[Ros+21] Elan Rosenfeld, Pradeep Kumar Ravikumar, and Andrej Risteski. "The Risks of Invariant Risk Minimization". In: *ICLR*. 2021 (cit. on p. 163).

[Rua+22] Yangjun Ruan, Yann Dubois, and Chris J. Maddison. "Optimal Representations for Covariate Shift". In: *ICLR*. 2022 (cit. on pp. 19, 138).

[Rui+20] Adrià Ruiz and Jakob Verbeek. "Distilled Hierarchical Neural Ensembles with Adaptive Inference Cost". 2020 (cit. on p. 22).

[Rum+85] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. *Learning internal representations by error propagation*. 1985 (cit. on p. 3).

[Rus+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. "ImageNet large scale visual recognition challenge". In: *IJCV*. 2015 (cit. on pp. 2, 38).

[Rus+16] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, 2016 (cit. on p. 9).

[Sag+18] Levent Sagun, Utku Evci, V. Ugur Guney, Yann Dauphin, and Leon Bottou. *Empirical Analysis of the Hessian of Over-Parametrized Neural Networks*. 2018 (cit. on p. 166).

[Sag+20a] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. "Distributionally Robust Neural Networks". In: *ICLR*. 2020 (cit. on pp. 19, 163).

[Sag+20b] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. "An Investigation of Why Overparameterization Exacerbates Spurious Correlations". In: *ICML*. 2020 (cit. on p. 163).

[Sal+23] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. "LaMP: When Large Language Models Meet Personalization". In: *arXiv preprint* (2023) (cit. on pp. 76, 131).

[San+20] Karthik Abinav Sankararaman, Soham De, Zheng Xu, W Ronny Huang, and Tom Goldstein. "The impact of neural network overparameterization on gradient confusion and stochastic gradient descent". In: *ICML*. 2020 (cit. on pp. 161, 164).

[San+23] Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. "Whose opinions do language models reflect?" In: *arXiv preprint* (2023) (cit. on p. 64).

[Sch+12] Shalom H Schwartz et al. "An overview of the Schwartz theory of basic values". In: *Online readings in Psychology and Culture* (2012) (cit. on p. 63).

[Sch+17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. “Proximal policy optimization algorithms”. In: *arXiv preprint* (2017) (cit. on p. 69).

[Sch+23] Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. “Toolformer: Language models can teach themselves to use tools”. In: *arXiv preprint* (2023) (cit. on p. 80).

[Sch02] Nicol N Schraudolph. “Fast curvature matrix-vector products for second-order gradient descent”. In: *Neural computation*. 2002 (cit. on pp. 147, 168).

[Sek+20] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. “Planning to Explore via Self-Supervised World Models”. In: *ICML*. 2020 (cit. on p. 68).

[Sel+22] Mariia Seleznova and Gitta Kutyniok. “Neural Tangent Kernel Beyond the Infinite-Width Limit: Effects of Depth and Initialization”. In: *ICML* (2022) (cit. on pp. 17, 134).

[Sha+20] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. “The Pitfalls of Simplicity Bias in Neural Networks”. In: *NeurIPS*. 2020 (cit. on pp. 4, 161).

[Sha+21] Soroosh Shahtalebi, Jean-Christophe Gagnon-Audet, Touraj Laleh, Mojtaba Faramarzi, Kartik Ahuja, and Irina Rish. “SAND-mask: An Enhanced Gradient Masking Strategy for the Discovery of Invariances in Domain Generalization”. In: *ICML Workshop*. 2021 (cit. on pp. 164, 167).

[Sha+22] Rohin Shah, Vikrant Varma, Ramana Kumar, Mary Phuong, Victoria Krakovna, Jonathan Uesato, and Zac Kenton. “Goal misgeneralization: Why correct specifications aren’t enough for correct goals”. In: *arXiv preprint* (2022) (cit. on p. 82).

[She+23] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. “Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface”. In: *arXiv preprint* (2023) (cit. on p. 80).

[Shi+21] Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. “Gradient Matching for Domain Generalization”. In: *arXiv preprint* (2021) (cit. on pp. 162, 164, 169).

[Shu+18] Changjian Shui, Azadeh Sadat Mozafari, Jonathan Marek, Ihsen Hedhli, and Christian Gagné. “Diversity Regularization in Deep Ensembles”. In: (2018) (cit. on p. 22).

[Shu+23] Mustafa Shukor, Corentin Dancette, Alexandre Ramé, and Matthieu Cord. “UnIVAL: Unified Model for Image, Video, Audio and Language”. In: *arXiv preprint* (2023) (cit. on pp. 7, 81).

[Sie+18] Demetrio Sierra-Mercado and Gabriel Lázaro-Muñoz. “Enhance diversity among researchers to promote participant trust in precision medicine research”. In: *The American Journal of Bioethics* (2018) (cit. on p. 132).

[Sil+16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. "Mastering the game of Go with deep neural networks and tree search". In: *Nature* (2016) (cit. on p. 2).

[Sin+03] Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. "Nearest neighbor estimates of entropy". In: *American journal of mathematical and management sciences* (2003) (cit. on p. 30).

[Sin+16] Saurabh Singh, Derek Hoiem, and David Forsyth. "Swapout: Learning an ensemble of deep architectures". In: *NeurIPS*. 2016 (cit. on p. 22).

[Sin+20a] Sidak Pal Singh and Dan Alistarh. "WoodFisher: Efficient Second-Order Approximation for Neural Network Compression". In: *NeurIPS*. 2020 (cit. on p. 168).

[Sin+20b] Samarth Sinha, Homanga Bharadhwaj, Anirudh Goyal, Hugo Larochelle, Animesh Garg, and Florian Shkurti. "DIBS: Diversity inducing Information Bottleneck in Model Ensembles". In: *AAAI*. 2020 (cit. on p. 32).

[Ska+22] Joar Max Viktor Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. "Defining and Characterizing Reward Gaming". In: *NeurIPS*. 2022 (cit. on p. 82).

[Smi21] Ben Smith. *A brief review of the reasons multi-objective RL could be important in AI Safety Research*. `https://www.alignmentforum.org/posts/i5dLfi6m6FCexReK9/a-brief-review-of-the-reasons-multi-objective-rl-could-be`. 2021 (cit. on p. 82).

[Sof+20] Masoumeh Soflaei, Hongyu Guo, Ali Al-Bashabsheh, Yongyi Mao, and Richong Zhang. "Aggregated Learning: A Vector-Quantization Approach to Learning Neural Network Classifiers". In: *AAAI*. 2020 (cit. on pp. 23, 149).

[Son+18] Guocong Song and Wei Chai. "Collaborative learning for deep neural networks". In: *NeurIPS*. 2018 (cit. on pp. 22, 23).

[Sti+20a] Asa Cooper Stickland and Iain Murray. "Diverse Ensembles Improve Calibration". In: *arXiv preprint* (2020) (cit. on p. 22).

[Sti+20b] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. "Learning to summarize with human feedback". In: *NeurIPS* (2020) (cit. on pp. 63, 64, 69).

[Sto+22] Zafir Stojanovski, Karsten Roth, and Zeynep Akata. "Momentum-based Weight Interpolation of Strong Zero-Shot Models for Continual Learning". In: *NeurIPS Workshop*. 2022 (cit. on p. 75).

[Sum+19] Cecilia Summers and Michael J Dinneen. "Improved Mixed-Example Data Augmentation". In: *WACV*. 2019 (cit. on p. 151).

[Sun+16] Baochen Sun, Jiashi Feng, and Kate Saenko. "Return of Frustratingly Easy Domain Adaptation". In: *AAAI*. 2016 (cit. on pp. 45, 46, 49, 50, 57, 161, 163, 164).

[Sun+20] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. "Scalability in Perception for Autonomous Driving: Waymo Open Dataset". In: *CVPR*. 2020 (cit. on p. 2).

[Sun+22] Rémy Sun, Alexandre Ramé, Clément Masson, Nicolas Thome, and Matthieu Cord. "Towards efficient feature sharing in MIMO architectures". In: *CVPR Workshop*. 2022 (cit. on pp. 6, 7, 159).

[Sun+23] Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. "Principle-Driven Self-Alignment of Language Models from Scratch with Minimal Human Supervision". In: *arXiv preprint* (2023) (cit. on p. 80).

[Sut19] Richard Sutton. "The bitter lesson". In: *Incomplete Ideas blog* (2019) (cit. on p. 78).

[Swa+19] Talia H Swartz, Ann-Gel S Palermo, Sandra K Masur, and Judith A Aberg. "The science and value of diversity: Closing the gaps in our understanding of inclusion and diversity". In: *The Journal of infectious diseases* (2019) (cit. on p. 132).

[Sze+15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions". In: *CVPR*. 2015 (cit. on p. 33).

[Sze+16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision". In: *CVPR*. 2016 (cit. on pp. 39, 53).

[Tak+20] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. "Data augmentation using random image cropping and patching for deep cnns". In: *IEEE Transactions on Circuits and Systems for Video Technology* (2020) (cit. on p. 151).

[Tan+03] Fumihide Tanaka and Masayuki Yamamura. "Multitask reinforcement learning on the distribution of MDPs". In: *CIRA*. 2003 (cit. on pp. 64, 75).

[Tao+20] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. "Measuring Robustness to Natural Distribution Shifts in Image Classification". In: *NeurIPS*. 2020 (cit. on p. 37).

[Tao+23] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. *Stanford Alpaca: An Instruction-following LLaMA model*. `https://github.com/tatsu-lab/stanford_alpaca`. 2023 (cit. on pp. 63, 64, 69).

[Tay+16] Jessica Taylor, Eliezer Yudkowsky, Patrick LaVictoire, and Andrew Critch. "Alignment for advanced machine learning systems". In: *Ethics of AI* (2016) (cit. on pp. 4, 37, 63, 82).

[Tay+19] J Taylor, B Earnshaw, B Mabey, M Victors, and J Yosinski. “RxRx1: An image set for cellular morphological variation across many experimental batches”. In: *ICLR Workshop*. 2019 (cit. on p. 59).

[Ten+21] Damien Teney, Ehsan Abbasnejad, and Anton van den Hengel. “Unshuffling Data for Improved Generalization”. In: *ICCV*. 2021 (cit. on p. 163).

[Ten+22] Damien Teney, Yong Lin, Seong Joon Oh, and Ehsan Abbasnejad. “ID and OOD performance are sometimes inversely correlated on real-world datasets”. In: *arXiv preprint* (2022) (cit. on p. 61).

[Ten18] Josh Tenenbaum. “Building machines that learn and think like people”. In: *AAMAS*. 2018 (cit. on p. 161).

[The+18] L. Theis, I. Korshunova, A. Tejani, and F. Huszár. “Faster gaze prediction with dense networks and Fisher pruning”. In: *arXiv preprint* (2018) (cit. on p. 169).

[Tho+20] Valentin Thomas, Fabian Pedregosa, Bart van Merriënboer, Pierre-Antoine Manzagol, Yoshua Bengio, and Nicolas Le Roux. “On the interplay between noise and curvature and its effect on optimization and generalization”. In: *AISTATS*. 2020 (cit. on pp. 147, 168, 169).

[Tis01] Naftali Tishby. “The information bottleneck method”. In: *CCCSP*. 2001 (cit. on p. 25).

[Tod+12] Emanuel Todorov, Tom Erez, and Yuval Tassa. “MuJoCo: A physics engine for model-based control”. In: *IROS*. 2012 (cit. on pp. 74, 75).

[Tok+18a] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. “Between-Class Learning for Image Classification”. In: *CVPR*. 2018 (cit. on p. 151).

[Tok+18b] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. “Learning from Between-class Examples for Deep Sound Recognition”. In: *ICLR*. 2018 (cit. on p. 151).

[Tol+23] Jamie Tolan, Hung-I Yang, Ben Nosarzewski, Guillaume Couairon, Huy Vo, John Brandt, Justine Spore, Sayantan Majumdar, Daniel Haziza, Janaki Vamaraju, et al. “Sub-meter resolution canopy height maps using self-supervised learning and a vision transformer trained on Aerial and GEDI Lidar”. In: *arXiv preprint* (2023) (cit. on p. 2).

[Tös+09] Andreas Töscher and Michael Jahrer. “The BigChaos Solution to the Netflix Grand Prize”. In: (2009) (cit. on p. 1).

[Tou+23a] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. “LLaMA: Open and Efficient Foundation Language Models”. In: *arXiv preprint* (2023) (cit. on pp. 64, 69, 70).

[Tou+23b] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. "LLaMA 2: Open Foundation and Fine-Tuned Chat Models". In: *arXiv preprint* (2023) (cit. on pp. 81, 83).

[Tze+14] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. "Deep domain confusion: Maximizing for domain invariance". In: *CoRR*. 2014 (cit. on p. 163).

[Ued+96] Naonori Ueda and Ryohei Nakano. "Generalization error of ensemble estimators". In: *ICNN*. 1996 (cit. on pp. 20, 41, 139).

[Val+19] Guillermo Valle-Perez, Chico Q. Camargo, and Ard A. Louis. "Deep learning generalizes because the parameter-function map is biased towards simple functions". In: *ICLR*. 2019 (cit. on p. 161).

[Vam+08] Peter Vamplew, John Yearwood, Richard Dazeley, and Adam Berry. "On the limitations of scalarisation for multi-objective reinforcement learning of pareto fronts". In: *AJCAIA*. 2008 (cit. on pp. 66, 68, 81).

[Vam+11] Peter Vamplew, Richard Dazeley, Adam Berry, Rustam Issabekov, and Evan Dekker. "Empirical evaluation methods for multiobjective reinforcement learning algorithms". In: *Deakin University* (2011) (cit. on p. 75).

[Vam+18] Peter Vamplew, Richard Dazeley, Cameron Foale, Sally Firmin, and Jane Mummery. "Human-aligned artificial intelligence is a multiobjective problem". In: *Ethics and Information Technology* (2018) (cit. on pp. 64, 81).

[Van+14] Kristof Van Moffaert and Ann Nowé. "Multi-objective reinforcement learning using sets of pareto dominating policies". In: *JMLR* (2014) (cit. on pp. 64, 75).

[Vap92] V. Vapnik. "Principles of Risk Minimization for Learning Theory". In: *NeurIPS*. 1992 (cit. on pp. 11, 38, 51, 53).

[Vap99] Vladimir N Vapnik. "An overview of statistical learning theory". In: *TNN*. 1999 (cit. on pp. 11, 63, 163).

[Vas+15] Bogdan Vasilescu, Daryl Posnett, Baishakhi Ray, Mark GJ van den Brand, Alexander Serebrenik, Premkumar Devanbu, and Vladimir Filkov. "Gender and tenure diversity in GitHub teams". In: *ACM*. 2015 (cit. on p. 132).

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is All you Need". In: *NeurIPS*. 2017 (cit. on pp. 2, 3, 11).

[Ved+15] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. "Consensus-based image description evaluation". In: *ICCV*. 2015 (cit. on pp. 63, 70).

[Ven+17] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. "Deep hashing network for unsupervised domain adaptation". In: *CVPR*. 2017 (cit. on pp. 14, 39, 45, 46).

[Ver+19] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. "Manifold Mixup: Better Representations by Interpolating Hidden States". In: *ICML*. 2019 (cit. on pp. 151, 155).

[Völ+17] Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. "Tl; dr: Mining reddit to learn automatic summarization". In: *ACL Workshop*. 2017 (cit. on p. 69).

[Wan+10] Jack M Wang, David J Fleet, and Aaron Hertzmann. "Optimizing walking controllers for uncertain inputs and environments". In: *ACM* (2010) (cit. on p. 75).

[Wan+20a] Xiaofang Wang, Dan Kondratyuk, Kris M. Kitani, Yair Movshovitz-Attias, and Elad Eban. "Multiple Networks are More Efficient than One: Fast and Accurate Models via Ensembles and Cascades". In: *arXiv preprint* (2020) (cit. on p. 23).

[Wan+20b] Yufei Wang, Haoliang Li, and Alex C Kot. "Heterogeneous domain generalization via domain mixup". In: *ICASSP*. 2020 (cit. on p. 163).

[Wan+22a] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. "OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework". In: *ICML*. 2022 (cit. on pp. 73, 74, 81).

[Wan+22b] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. "Self-Instruct: Aligning Language Model with Self Generated Instructions". In: *arXiv preprint* (2022) (cit. on p. 69).

[Wan+22c] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit

Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. "Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks". In: *ACL*. 2022 (cit. on p. 63).

[Was+20] Abdul Wasay, Brian Hentschel, Yuze Liao, Sanyuan Chen, and Stratos Idreos. "MotherNets: Rapid Deep Ensemble Learning". In: *MLSys*. 2020 (cit. on p. 24).

[Wei+21] Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. "Ethical and social risks of harm from language models". In: *arXiv preprint* (2021) (cit. on p. 64).

[Wei+22a] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. "Finetuned Language Models are Zero-Shot Learners". In: *ICLR*. 2022 (cit. on pp. 63, 69).

[Wei+22b] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. "Emergent Abilities of Large Language Models". In: *TMLR* (2022) (cit. on p. 3).

[Wei+22c] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. "Chain of thought prompting elicits reasoning in large language models". In: *NeurIPS*. 2022 (cit. on pp. 3, 80).

[Wei+22d] Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, et al. "Taxonomy of risks posed by language models". In: *ACM*. 2022 (cit. on p. 81).

[Wen+19] Yeming Wen, Dustin Tran, and Jimmy Ba. "BatchEnsemble: an Alternative Approach to Efficient Ensemble and Lifelong Learning". In: *ICLR*. 2019 (cit. on p. 23).

[Wen+20] Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. "Hyperparameter Ensembles for Robustness and Uncertainty Quantification". In: *NeurIPS*. 2020 (cit. on p. 22).

[Wen+21] Yeming Wen, Ghassen Jerfel, Rafael Muller, Michael W Dusenberry, Jasper Snoek, Balaji Lakshminarayanan, and Dustin Tran. "Combining Ensembles and Data Augmentation Can Harm Your Calibration". In: *ICLR*. 2021 (cit. on pp. 22, 152).

[Wen+22] Florian Wenzel, Andrea Dittadi, Peter Vincent Gehler, Carl-Johann Simon-Gabriel, Max Horn, Dominik Zietlow, David Kernert, Chris Russell, Thomas Brox, Bernt Schiele, Bernhard Schölkopf, and Francesco Locatello. "Assaying Out-Of-Distribution Generalization in Transfer Learning". In: *NeurIPS*. 2022 (cit. on p. 78).

[Wer+20] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, and Nathan Lambert. *TRL: Transformer Reinforcement Learning*. https://github.com/lvwerra/trl. 2020 (cit. on p. 69).

[Wig+21] Ross Wightman, Hugo Touvron, and Hervé Jégou. "ResNet strikes back: An improved training procedure in timm". In: *NeurIPS Workshop*. 2021 (cit. on p. 12).

[Wig19] Ross Wightman. *PyTorch Image Models*. https://github.com/rwightman/pytorch-image-models. 2019 (cit. on p. 51).

[Wil+07] Aaron Wilson, Alan Fern, Soumya Ray, and Prasad Tadepalli. "Multi-task reinforcement learning: a hierarchical bayesian approach". In: *ICML*. 2007 (cit. on p. 75).

[Wil+20] Andrew Gordon Wilson and Pavel Izmailov. "Bayesian deep learning and a probabilistic perspective of generalization". In: *NeurIPS* (2020) (cit. on p. 22).

[Wil87] Aaron Wildavsky. "Choosing preferences by constructing institutions: A cultural theory of preference formation". In: *American political science review* (1987) (cit. on p. 63).

[Wil92] Ronald J Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Reinforcement learning* (1992) (cit. on p. 71).

[Wol+20] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. "Transformers: State-of-the-Art Natural Language Processing". In: *EMNLP*. 2020 (cit. on pp. 2, 51, 69, 70).

[Wol+23] Yotam Wolf, Noam Wies, Yoav Levine, and Amnon Shashua. "Fundamental Limitations of Alignment in Large Language Models". In: *arXiv preprint* (2023) (cit. on p. 82).

[Wol92] David H Wolpert. "Stacked generalization". In: *Neural networks* (1992) (cit. on p. 19).

[Won+20] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. "A scalable approach to control diverse behaviors for physically simulated characters". In: *TOG* (2020) (cit. on p. 75).

[Woo+23] Danny Wood, Tingting Mu, Andrew Webb, Henry Reeve, Mikel Lujan, and Gavin Brown. "A Unified Theory of Diversity in Ensemble Learning". In: *arXiv preprint* (2023) (cit. on pp. 15, 21).

[Wor+21] Mitchell Wortsman, Maxwell Horton, Carlos Guestrin, Ali Farhadi, and Mohammad Rastegari. "Learning Neural Network Subspaces". In: *ICML* (2021) (cit. on p. 45).

[Wor+22a] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time". In: *ICML*. 2022 (cit. on pp. 41, 42, 44, 45, 52, 53, 56, 58, 60, 65, 79).

[Wor+22b] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Hanna Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. "Robust fine-tuning of zero-shot models". In: *CVPR*. 2022 (cit. on pp. 39, 43, 45, 51–53, 55).

[Wor+23] Mitchell Wortsman, Suchin Gururangan, Shen Li, Ali Farhadi, Ludwig Schmidt, Michael Rabbat, and Ari S. Morcos. "lo-fi: distributed fine-tuning without communication". In: *TMLR* (2023) (cit. on pp. 35, 79, 80).

[Wor23] Mitchell Wortsman. *Reaching 80% zero-shot accuracy with OpenCLIP: VIT-G/14 trained on LAION-2b*. https://laion.ai/blog/giant-openclip/. 2023 (cit. on p. 79).

[Wu+20a] Guile Wu and Shaogang Gong. "Peer Collaborative Learning for Online Knowledge Distillation". In: *AAAI*. 2020 (cit. on p. 22).

[Wu+20b] Yuan Wu, Diana Inkpen, and Ahmed El-Roby. "Dual mixup regularized learning for adversarial domain adaptation". In: *ECCV*. 2020 (cit. on p. 163).

[Wu+21] Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. "Recursively summarizing books with human feedback". In: *arXiv preprint* (2021) (cit. on pp. 63, 69).

[Wu+23a] Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. "Better Aligning Text-to-Image Models with Human Preference". In: *arXiv preprint* (2023) (cit. on pp. 63, 72).

[Wu+23b] Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. "Fine-Grained Human Feedback Gives Better Rewards for Language Model Training". In: *arXiv preprint* (2023) (cit. on pp. 80, 81).

[Wu+23c] Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. "Fine-Grained Human Feedback Gives Better Rewards for Language Model Training". In: *arXiv preprint* (2023) (cit. on pp. 64, 66, 75).

[Xie+22] Zhihui Xie, Zichuan Lin, Junyou Li, Shuai Li, and Deheng Ye. "Pretraining in Deep Reinforcement Learning: A Survey". In: *arXiv preprint* (2022) (cit. on p. 68).

[Xie+23] Enze Xie, Lewei Yao, Han Shi, Zhili Liu, Daquan Zhou, Zhaoqiang Liu, Jiawei Li, and Zhenguo Li. "DiffFit: Unlocking Transferability of Large Diffusion Models via Simple Parameter-Efficient Fine-Tuning". In: *arXiv preprint arXiv:2304.06648* (2023) (cit. on p. 72).

[Xu+23] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. “ImageReward: Learning and Evaluating Human Preferences for Text-to-Image Generation”. In: *arXiv preprint* (2023) (cit. on pp. 63, 72).

[Yag+19] Yoichi Yaguchi, Fumiyuki Shiratani, and Hidekazu Iwaki. “MixFeat: Mix Feature in Latent Space Learns Discriminative Space”. In: *Openreview preprint* (2019) (cit. on p. 151).

[Yan+19a] Greg Yang and Hadi Salman. “A Fine-Grained Spectral Perspective on Neural Networks”. In: *arXiv preprint* (2019) (cit. on p. 134).

[Yan+19b] Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. “A Generalized Algorithm for Multi-Objective Reinforcement Learning and Policy Adaptation”. In: *NeurIPS*. 2019 (cit. on p. 75).

[Yan+20a] Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. “Improve unsupervised domain adaptation with mixup training”. In: *arXiv preprint* (2020) (cit. on p. 50).

[Yan+20b] Chuanyu Yang, Kai Yuan, Qiuguo Zhu, Wanming Yu, and Zhibin Li. “Multi-expert learning of adaptive legged locomotion”. In: *Science Robotics* (2020) (cit. on p. 75).

[Yan+20c] Taojiannan Yang, Sijie Zhu, and Chen Chen. “GradAug: A New Regularization Method for Deep Neural Networks”. In: *NeurIPS* (2020) (cit. on pp. 23, 155, 159).

[Yan+20d] Yanchao Yang and Stefano Soatto. “Fda: Fourier domain adaptation for semantic segmentation”. In: *CVPR*. 2020 (cit. on p. 22).

[Yan+23a] Rui Yang, Lin Song, Yanwei Li, Sijie Zhao, Yixiao Ge, Xiu Li, and Ying Shan. “GPT4Tools: Teaching Large Language Model to Use Tools via Self-instruction”. In: *arXiv preprint* (2023) (cit. on p. 80).

[Yan+23b] Sherry Yang, Ofir Nachum, Yilun Du, Jason Wei, Pieter Abbeel, and Dale Schuurmans. “Foundation Models for Decision Making: Problems, Methods, and Opportunities”. In: *arXiv preprint* (2023) (cit. on p. 68).

[Yao+20] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. “Pyhessian: Neural networks through the lens of the hessian”. In: *Big Data*. 2020 (cit. on p. 39).

[Yao+23] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. “Tree of thoughts: Deliberate problem solving with large language models”. In: *arXiv preprint* (2023) (cit. on p. 80).

[Ye+22] Nanyang Ye, Kaican Li, Lanqing Hong, Haoyue Bai, Yiting Chen, Fengwei Zhou, and Zhenguo Li. “OoD-Bench: Benchmarking and Understanding Out-of-Distribution Generalization Datasets and Algorithms”. In: *CVPR* (2022) (cit. on pp. 13, 17, 18, 37, 38, 45, 49, 53, 137, 161).

[Yen+13] Gary G Yen and Zhenan He. “Performance metric ensemble for multiobjective evolutionary algorithms”. In: *TEVC* (2013) (cit. on pp. 69, 72).

[Yin+18] Dong Yin, Ashwin Pananjady, Max Lam, Dimitris Papailiopoulos, Kannan Ramchandran, and Peter Bartlett. "Gradient diversity: a key ingredient for scalable distributed learning". In: *AISTATS*. 2018 (cit. on pp. 161, 164).

[Yos+14] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. "How transferable are features in deep neural networks?" In: *NeuriPS*. 2014 (cit. on pp. 3, 63).

[Yu+16] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. "Modeling context in referring expressions". In: *ECCV*. 2016 (cit. on p. 73).

[Yu+20] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. "Gradient Surgery for Multi-Task Learning". In: *NeurIPS*. 2020 (cit. on p. 164).

[Yu+23] Han Yu, Xingxuan Zhang, Renzhe Xu, Jiashuo Liu, Yue He, and Peng Cui. "Rethinking the Evaluation Protocol of Domain Generalization". In: *arXiv preprint* (2023) (cit. on pp. 19, 171).

[Yul00] George Udny Yule. "On the association of attributes in statistics". In: *Philosophical Transactions of the Royal Society of London.* (1900) (cit. on p. 21).

[Yun+19] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. "Cutmix: Regularization strategy to train strong classifiers with localizable features". In: *ICCV*. 2019 (cit. on pp. 12, 23, 149, 151, 155, 159).

[Zag+16] Sergey Zagoruyko and Nikos Komodakis. "Wide Residual Networks". In: *BMVC*. 2016 (cit. on p. 155).

[Zec+18] John R. Zech, Marcus A. Badgeley, Manway Liu, Anthony B. Costa, Joseph J. Titano, and Eric Karl Oermann. "Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study". In: *PLOS Medicine* (2018) (cit. on pp. 37, 51).

[Zha+17] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. "Understanding deep learning requires rethinking generalization". In: *ICLR* (2017) (cit. on pp. 3, 13, 151).

[Zha+18a] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. "mixup: Beyond Empirical Risk Minimization". In: *ICLR*. 2018 (cit. on pp. 12, 149, 151, 155).

[Zha+18b] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. "Deep mutual learning". In: *CVPR*. 2018 (cit. on p. 22).

[Zha+19a] Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. "Lookahead optimizer: k steps forward, 1 step back". In: *NeurIPS* 32 (2019) (cit. on p. 24).

[Zha+19b] Yunbo Zhang, Wenhao Yu, and Greg Turk. "Learning novel policies for tasks". In: *ICML*. 2019 (cit. on p. 164).

[Zha+19c] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. "On learning invariant representations for domain adaptation". In: *ICML*. 2019 (cit. on pp. 163, 167).

[Zha+20a] Marvin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. "Adaptive Risk Minimization: A Meta-Learning Approach for Tackling Group Distribution Shift". In: *arXiv preprint* (2020) (cit. on p. 164).

[Zha+20b] Shuai Zhao, Liguang Zhou, Wenxiao Wang, Deng Cai, Tin Lun Lam, and Yangsheng Xu. "SplitNet: Divide and Co-training". In: *arXiv preprint* (2020) (cit. on pp. 23, 159).

[Zha+21] Xingxuan Zhang, Peng Cui, Renzhe Xu, Linjun Zhou, Yue He, and Zheyan Shen. "Deep Stable Learning for Out-Of-Distribution Generalization". In: *CVPR*. 2021 (cit. on p. 163).

[Zha+22] Jianyu Zhang, David Lopez-Paz, and Léon Bottou. "Rich Feature Construction for the Optimization-Generalization Dilemma". In: *ICML*. 2022 (cit. on p. 53).

[Zha+23a] Jianyu Zhang and Léon Bottou. "Learning useful representations for shifting tasks and distributions". In: *ICML*. 2023 (cit. on p. 53).

[Zha+23b] Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, et al. "HIVE: Harnessing Human Feedback for Instructional Visual Editing". In: *arXiv preprint* (2023) (cit. on p. 63).

[Zho+02] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. "Ensembling neural networks: many could be better than all". In: *Artificial intelligence* (2002) (cit. on p. 19).

[Zhu+23] Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, et al. "Ghost in the Minecraft: Generally Capable Agents for Open-World Enviroments via Large Language Models with Text-based Knowledge and Memory". In: *arXiv preprint* (2023) (cit. on p. 81).

[Zie+19] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. "Fine-tuning language models from human preferences". In: *arXiv preprint* (2019) (cit. on p. 63).

Appendix A

ACRONYMS AND NOTATIONS

Contents

A.1 Acronyms

AI	artificial intelligence
BN	batch normalization
CEB	conditional entropy bottleneck
CKAC	centered kernel alignment complement
CNN	convolutional neural network
CV	computer vision
DA	data augmentation
DE	deep ensembling
DL	deep learning
DNN	deep neural network
DRL	deep reinforcement learning
ECE	expected calibration error
ENS	ensembling
ERM	empirical risk minimization
FIM	Fisher information matrix
GPU	graphics processing unit
GP	Gaussian process
GPT	generative pre-trained transformers
IB	information bottleneck
ID	in-distribution
IoU	intersection over union
KL	Kullback-Leibler divergence

LLM	large language model
LMC	linear mode connectivity
LP	linear probing
MA	moving average
MI	mutual information
MLP	multi-layer perceptron
ML	machine learning
MMD	maximum mean discrepancy
MNI	minimum necessary information
MORL	multi-objective reinforcement learning
MSA	memory split advantage
MSDA	mixed sample data augmentation
MSE	mean-squared error
NLL_c	calibrated negative log-likelihood
NLP	natural language processing
NTK	neural tangent kernel
OOD	out-of-distribution
PCS	Pareto coverage set
PF	Pareto front
QA	question answering
RGB	red green blue
RKHS	reproducing kernel Hilbert space
RLAIF	reinforcement learning from AI feedback
RLHF	reinforcement learning from human feedback
RL	reinforcement learning
ReLU	rectified linear unit
SAM	sharpness-aware minimization
SGD	stochastic gradient descent
sota	state-of-the-art
TS	temperature scaling
VG	visual grounding
VQA	visual question answering
WA	weight averaging
WI	weight interpolation

A.2 Notations

DNN model	f parameterized with weights θ
Featurizer (encoder)	Φ (or e) parameterized with weights ϕ
Classifier (dense layer)	w (or d) parameterized with weights ω
Input	$x \in \mathcal{X}$
Embedding	z
Label	$y \in \mathcal{Y}$
Prediction	$\hat{y} \in \mathcal{Y}$
Number of classes	K
Number of models in ensembling	M
Probability density function and measure	p and P
Random variables	X, Y, Z
Ratios	λ, μ, κ
Train (source) domain	S with distribution p_S
Test (target) domain	T with distribution p_T
Training dataset	$\mathcal{D}_S = \{x_n, y_n\}_{n=1}^{n_S}$
Training configuration	c
Learning procedure	$l_S = \{\mathcal{D}_S, c\}$
Loss function	$\ell : \mathcal{Y}^2 \to \mathbb{R}_+$

Appendix B

SOCIETAL IMPACTS

This thesis was crafted within a given social and political context, and we now discuss its potential societal impact.

Sustainability. Proposing WA as an alternative to ensembling is a step towards more sustainable DL. By reducing the computational cost at inference, we may diminish the carbon footprint of large DNNs. However, the potential issue is that this might lead to a "rebound effect" where the efficiency gain enables the use of even larger models.

Decentralization. We also promote the *updatable machine learning* paradigm [Raf23], which facilitates "the collaborative creation of increasingly sophisticated AI systems" [Ram+23a]. As discussed in Section 7.2.3, this paradigm may further encourage the development of open-source models, potentially leading to the creation of more responsible and reliable AI systems that can adapt and learn in a constantly changing environment. This could help networks represent a more diverse range of opinions (as discussed in Chapter 6) by mitigating the "tyranny of the crowdworker" [Kir+23], where models are "tailored to meet the expectations of [...] a small number of crowdworkers primarily based in the US, with little to no representation of broader human cultures, geographies, or languages" [Kir+23], and more generally the cultural hegemony of a few individuals [Dur+23].

Privacy and fairness. The parallelization of ensembling approaches is compatible with *federated learning* scenarios [McM+17], where data must remain private. Additionally, by improving generalization, our methods can mitigate the impact of subpopulation shift.

Transparency. In DL, decisions made during the training of models shift the responsibility from the problem stakeholders to the system engineers, who need to anticipate the impacts of their choices. We demonstrated that combining multiple models offers an additional level of flexibility: by selecting the weighting coefficients a posteriori, we can improve the *transparency* [Gab+21] of AI systems, and facilitate regulation by an external non-technical authority. This may prove crucial to ensure the development of fair, unbiased, and inclusive [Aba+16] AIs. This could facilitate tailored generations to minorities by *model personalization* [Sal+23], but also pause risks of echo chambers and bias reinforcement, as discussed in [Kir+23] and Section 6.4.

Sociological analogy. More generally, we showed that the optimal diversity-accuracy trade-off was obtained with large diversity in DL. By analogy, this thesis promotes more diversity in our society [Mul16]: ideas should be shared and debated among members reflecting the diversity of the society's various components and backgrounds. Actually, in social science, this insight is known as the the Hong-Page theorem [Hon+04]: a group of low-ability, cognitively diverse people can outperform a more uniform group of high-ability experts to solve problems. Academia especially needs this diversity to promote trust in research [Sie+18], to improve quality of the findings [Swa+19], productivity of the teams [Vas+15] and even schooling's impact [Bow13].

APPENDIX C

PROOFS

Contents

C.1 Proofs for ENSEMBLING

We detail the proofs for the theoretical arguments in Chapter 2.

- Appendix C.1.1 proves the link between variance and diversity shift (Proposition 2.1).
- Appendix C.1.2 proves the link between bias and correlation shift (Proposition 2.2).
- Appendix C.1.3 proves the bias-variance-covariance decomposition for ensembling (Proposition 2.3).

C.1.1 Variance and diversity shift Proposition 2.1

We prove the link between variance and diversity shift. Our proof builds upon the similarity between DNNs and GPs in the interpolating regime, detailed in Appendix C.1.1.1. We discuss our simplifying Assumption 2.2 in Appendix C.1.1.2. We present our final proof in Appendix C.1.1.3.

C.1.1.1 Deep neural networks as GPs Assumption 2.1

We fix $\mathcal{D}_S, \mathcal{D}_T$ and denote $X_{\mathcal{D}_S} = \{x_S\}_{(x_S,y_S)\in\mathcal{D}_S}$, $X_{\mathcal{D}_T} = \{x_T\}_{(x_T,y_T)\in\mathcal{D}_T}$ their respective input supports. We fix the initialization of the network. l_S encapsulates all other sources of randomness.

Lemma C.1 (Inspired from [Ras03]). *Given a NN $f(\cdot, \theta(l_S))$ under Assumption 2.1, we denote K its NTK and $K(X_{\mathcal{D}_S}, X_{\mathcal{D}_S}) = (K(x_S, x'_S))_{x_S, x'_S \in X^2_{\mathcal{D}_S}} \in \mathbb{R}^{n_S \times n_S}$. Given $x \in \mathcal{X}$, we denote $K(x, X_{\mathcal{D}_S}) = [K(x, x_S)]_{x_S \in X_{\mathcal{D}_S}} \in \mathbb{R}^{n_S}$. Then:*

$$\mathrm{var}(x) = K(x,x) - K(x, X_{\mathcal{D}_S}) K(X_{\mathcal{D}_S}, X_{\mathcal{D}_S})^{-1} K(x, X_{\mathcal{D}_S})^{\intercal}. \tag{C.1}$$

Proof. Under Assumption 2.1, DNNs are equivalent to GPs. $\mathrm{var}(x)$ is the formula of the variance of the GP posterior given by Eq. (2.26) in [Ras03], when conditioned on $\mathcal{D}_S$. This formula thus also applies to the variance $f(\cdot, \theta(l_S))$ when l_S varies (at fixed $\mathcal{D}_S$ and initialization). □

C.1.1.2 Discussion of the same norm and low similarity Assumption 2.2

Lemma C.1 shows that the variance only depends on the input distributions without involving the label distributions. This formula highlights that the variance is related to shifts in input similarities (measured by K) between $X_{\mathcal{D}_S}$ and $X_{\mathcal{D}_T}$. Yet, a more refined analysis of the variance requires additional assumptions, in particular to obtain a closed-form expression of $K(X_{\mathcal{D}_S}, X_{\mathcal{D}_S})^{-1}$. Assumption 2.2 is useful because then $K(X_{\mathcal{D}_S}, X_{\mathcal{D}_S})$ is diagonally dominant and can be approximately inverted (see full proof in Appendix C.1.1.3).

The first part of Assumption 2.2 assumes that $\exists \lambda_S$ such that all training inputs $x_S \in X_{\mathcal{D}_S}$ verify $K(x_S, x_S) = \lambda_S$. Note that this equality is standard in some kernel machine algorithms [Ah-10; Gho+21; Ren05] and is usually achieved by replacing $K(x, x')$ by $\lambda_S \frac{K(x,x')}{\sqrt{K(x,x)}\sqrt{K(x',x')}}, \forall (x, x') \in (X_{\mathcal{D}_S} \cup X_{\mathcal{D}_T})^2$. In the NTK literature, this equality is achieved without changing the kernel by normalizing the samples of $X_{\mathcal{D}_S}$ such that they lie on the hypersphere; this input preprocessing was used in [Lee+17]. This is theoretically based: for example, the NTK $K(x, x')$ for an architecture with an initial fully connected layer only depends on $\|x\|, \|x'\|, \langle x, x' \rangle$ [Yan+19a]. Thus in the case where all samples from $X_{\mathcal{D}_S}$ are preprocessed to have the same norm, the value of $K(x_S, x_S)$ does not depend on $x_S \in X_{\mathcal{D}_S}$; we denote λ_S the corresponding value.

The second part of Assumption 2.2 states that $\exists 0 \leq \epsilon \ll \lambda_S$, s.t. $\forall x_S, x'_S \in X^2_{\mathcal{D}_S}, x_S \neq x'_S \Rightarrow |K(x_S, x'_S)| \leq \epsilon$, *i.e.*, that training samples are dissimilar and do not interact. This diagonal structure of the NTK [Jac+18], with diagonal values larger than non-diagonal ones, is consistent with empirical observations from [Sel+22] at initialization. Theoretically, this is reasonable if K is close to the RBF kernel $K_h(x, x') = \exp(-\|x - x'\|_2^2 / h)$ where h would be the bandwidth: in this case, Assumption 2.2 is satisfied when training inputs are distant in pixel space.

We now provide an analysis of the variance where the diagonal assumption is relaxed. Specifically, we provide the sketch for proving an upper-bound of the variance when the NTK has a block-diagonal structure. This is indeed closer to the empirical observations in [Sel+22] at the end of training, consistently with the local elasticity property of NNs [He+20]. We then consider the dataset $d_{S'} \subset \mathcal{D}_S$ made of one sample per block, to which Assumption 2.2 applies. As decreasing the size of a training dataset empirically reduces variance [Bra+99], the variance of f trained on $\mathcal{D}_S$ is upper-bounded by the variance of f trained on $d_{S'}$; the latter is given by applying Proposition 2.1 to $d_{S'}$. We believe that the proper formulation of this idea is beyond the scope of this proof and best left for future theoretical work.

C.1.1.3 Expression of OOD variance

We now prove Proposition 2.1 under Assumptions 2.1 and 2.2.

Proof. Our proof is original and is based on the posterior form of GPs in Lemma C.1. Given $\mathcal{D}_S$, we recall Equation (C.1) that states $\forall x \in \mathcal{X}$:

$$\mathrm{var}(x) = K(x,x) - K(x, X_{\mathcal{D}_S}) K(X_{\mathcal{D}_S}, X_{\mathcal{D}_S})^{-1} K(x, X_{\mathcal{D}_S})^{\intercal}.$$

Denoting $B = K(X_{\mathcal{D}_S}, X_{\mathcal{D}_S})^{-1}$ with symmetric coefficients $b_{i,j} = b_{j,i}$, then

$$\mathrm{var}(x) = K(x,x) - \sum_{\substack{1 \leq i \leq n_S \\ 1 \leq j \leq n_S}} b_{i,j} K(x, x_S^i) K(x, x_S^j). \tag{C.2}$$

Assumption 2.2 states that $K(X_{\mathcal{D}_S}, X_{\mathcal{D}_S}) = A + H$ where $A = \lambda_S \mathbb{I}_{n_S}$ and $H = (h_{ij})_{\substack{1 \leq i \leq n_S \\ 1 \leq j \leq n_S}}$ with $h_{i,i} = 0$ and $\max_{i,j} |h_{i,j}| \leq \epsilon$.

We fix $x_T \in X_{\mathcal{D}_T}$ and determine the form of B^{-1} in two cases: $\epsilon = 0$ and $\epsilon \neq 0$.

Case when $\epsilon = 0$ We first derive a simplified result, when $\epsilon = 0$.

Then, $b_{i,i} = \frac{1}{\lambda_S}$ and $b_{i,j} = 0$ s.t.

$$\mathrm{var}(x_T) = K(x_T, x_T) - \sum_{x_S \in X_{\mathcal{D}_S}} \frac{K(x_T, x_S)^2}{\lambda_S} = K(x,x) - \frac{n_S}{\lambda_S} \mathbb{E}_{x_S \in X_{\mathcal{D}_S}} [K^2(x, x_S)]$$

We can then write:

$$\begin{aligned}
\mathbb{E}_{x_T \in X_{\mathcal{D}_T}}[\mathrm{var}(x_T)] &= \mathbb{E}_{x_T \in X_{\mathcal{D}_T}}[K(x_T, x_T)] - \frac{n_S}{\lambda_S} \mathbb{E}_{x_T \in X_{\mathcal{D}_T}}[\mathbb{E}_{x_S \in X_{\mathcal{D}_S}}[K^2(x_T, x_S)]] \\
\mathbb{E}_{x_T \in X_{\mathcal{D}_T}}[\mathrm{var}(x_T)] &= \lambda_T - \frac{n_S}{\lambda_S} \mathbb{E}_{x_S \in X_{\mathcal{D}_S}, x_T \in X_{\mathcal{D}_T}}[K^2(x_T, x_S)].
\end{aligned}$$

We now relate the second term on the r.h.s. to a MMD distance. As K is a kernel, K^2 is a kernel and its MMD between $X_{\mathcal{D}_S}$ and $X_{\mathcal{D}_T}$ is per [Gre+12]:

$$\begin{aligned}
\mathrm{MMD}^2(X_{\mathcal{D}_S}, X_{\mathcal{D}_T}) =& \mathbb{E}_{x_S \neq x'_S \in X^2_{\mathcal{D}_S}}[K^2(x_S, x'_S)] + \mathbb{E}_{x_T \neq x'_T \in X^2_{\mathcal{D}_T}}[K^2(x_T, x'_T)] \\
& - 2\mathbb{E}_{x_S \in X_{\mathcal{D}_S}, x_T \in X_{\mathcal{D}_T}}[K^2(x_T, x_S)].
\end{aligned}$$

Finally, because $\epsilon = 0$, $\mathbb{E}_{x_S \neq x'_S \in X^2_{\mathcal{D}_S}} K^2\left(x_S, x'_S\right) = 0$ s.t.

$$\begin{aligned}
\mathbb{E}_{x_T \in X_{\mathcal{D}_T}}[\mathrm{var}(x_T)] &= \frac{n_S}{2\lambda_S}\mathrm{MMD}^2(X_{\mathcal{D}_S}, X_{\mathcal{D}_T}) + \lambda_T \\
&\quad - \frac{n_S}{2\lambda_S}\left(\mathbb{E}_{x_T \neq x'_T \in X^2_{\mathcal{D}_T}} K^2\left(x_T, x'_T\right) + \mathbb{E}_{x_S \neq x'_S \in X^2_{\mathcal{D}_S}} K^2\left(x_S, x'_S\right)\right) \\
&= \frac{n_S}{2\lambda_S}\mathrm{MMD}^2(X_{\mathcal{D}_S}, X_{\mathcal{D}_T}) + \lambda_T - \frac{n_S}{2\lambda_S}\mathbb{E}_{x_T \neq x'_T \in X^2_{\mathcal{D}_T}} K^2\left(x_T, x'_T\right) \\
&= \frac{n_S}{2\lambda_S}\mathrm{MMD}^2(X_{\mathcal{D}_S}, X_{\mathcal{D}_T}) + \lambda_T - \frac{n_S}{2\lambda_S}\beta_T.
\end{aligned}$$

We recover the same expression with a $\mathcal{O}(\epsilon)$ in the general setting where $\epsilon \neq 0$.

Case when $\epsilon \neq 0$ We denote $I : \begin{cases} \mathrm{GL}_{n_S}(\mathbb{R}) & \to \mathrm{GL}_{n_S}(\mathbb{R}) \\ A & \mapsto A^{-1} \end{cases}$ the inversion function defined on $\mathrm{GL}_{n_S}(\mathbb{R})$, the set of invertible matrices of $\mathcal{M}_{n_S}(\mathbb{R})$.

The function I is differentiable [Mag+19] in all $A \in \mathrm{GL}_{n_S}(\mathbb{R})$ with its differentiate given by the linear application $dI_A : \begin{cases} \mathcal{M}_{n_S}(\mathbb{R}) & \to \mathcal{M}_{n_S}(\mathbb{R}) \\ H & \mapsto -A^{-1}HA^{-1} \end{cases}$. Therefore, we can perform a Taylor expansion of I at the first order at A:

$$\begin{aligned}
I(A+H) &= I(A) + dI_A(H) + o(\|H\|), \\
(A+H)^{-1} &= A^{-1} - A^{-1}HA^{-1} + o(\|H\|).
\end{aligned}$$

where $\|H\| \leq n_S\epsilon = \mathcal{O}(\epsilon)$. Thus,

$$\begin{aligned}
(\lambda_S \mathbb{I}_{n_S} + H)^{-1} &= (\lambda_S \mathbb{I}_{n_S})^{-1} - (\lambda_S \mathbb{I}_{n_S})^{-1} H (\lambda_S \mathbb{I}_{n_S})^{-1} + \mathcal{O}(\epsilon) = \frac{1}{\lambda_S}\mathbb{I}_{n_S} - \frac{1}{\lambda_S^2}H + \mathcal{O}(\epsilon), \\
\forall i \in [\![1, n_S]\!], b_{ii} &= \frac{1}{\lambda_S} - \frac{1}{\lambda_S^2}h_{i,i} + o(\epsilon) = \frac{1}{\lambda_S} + \mathcal{O}(\epsilon), \\
\forall i \neq j \in [\![1, n_S]\!], b_{ij} &= -\frac{1}{\lambda_S^2}h_{i,j} + o(\epsilon) = \mathcal{O}(\epsilon).
\end{aligned}$$

Therefore, when ϵ is small, Equation (C.2) can be developed into:

$$\begin{aligned}
\mathrm{var}(x_T) &= K(x_T, x_T) - \sum_{x_S \in X_{\mathcal{D}_S}} (\frac{1}{\lambda_S} + \mathcal{O}(\epsilon)) K(x_T, x_S)^2 + \mathcal{O}(\epsilon) \\
&= K(x_T, x_T) - \frac{n_S}{\lambda_S}\mathbb{E}_{x_S \in X_{\mathcal{D}_S}}[K(x_T, x_S)^2] + \mathcal{O}(\epsilon)
\end{aligned}$$

Following the derivation for the case $\epsilon = 0$, and remarking that under Assumption 2.2 we have $\mathbb{E}_{x_S \neq x'_S \in X^2_{\mathcal{D}_S}} K^2\left(x_S, x'_S\right) = \mathcal{O}(\epsilon^2)$, yields:

$$\mathbb{E}_{x_T \in X_{\mathcal{D}_T}}[\mathrm{var}(x_T)] = \frac{n_S}{2\lambda_S}\mathrm{MMD}^2(X_{\mathcal{D}_S}, X_{\mathcal{D}_T}) + \lambda_T - \frac{n_S}{2\lambda_S}\beta_T + \mathcal{O}(\epsilon).$$

□

C.1.2 Bias and correlation shift (and support mismatch) Proposition 2.2

We first present in Appendix C.1.2.1 a decomposition of the bias without any assumptions, and then prove our results with the simplifying assumption Assumption 2.3.

C.1.2.1 OOD bias

Proposition (OOD bias)**.** *Denoting* $\bar{f}_S(x) = \mathbb{E}_{l_S}[f(x, \theta(l_S))]$, *the bias is:*

$$\mathbb{E}_{(x,y)\sim p_T}[\text{bias}^2(x,y)] = \int_{\mathcal{X}_T \cap \mathcal{X}_S} (f_T(x) - f_S(x))^2 p_T(x)dx \qquad \textit{(Correlation shift)}$$

$$+ \int_{\mathcal{X}_T \cap \mathcal{X}_S} \left(f_S(x) - \bar{f}_S(x)\right)^2 p_T(x)dx \qquad \textit{(Weighted ID bias)}$$

$$+ \int_{\mathcal{X}_T \cap \mathcal{X}_S} 2\left(f_T(x) - f_S(x)\right)\left(f_S(x) - \bar{f}_S(x)\right) p_T(x)dx \qquad \textit{(Interaction ID bias and corr. shift)}$$

$$+ \int_{\mathcal{X}_T \setminus \mathcal{X}_S} \left(f_T(x) - \bar{f}_S(x)\right)^2 p_T(x)dx. \qquad \textit{(Support mismatch)}$$

Proof. This proof is original and based on splitting the OOD bias in and out of $\mathcal{X}_S$:

$$\begin{aligned}
&\mathbb{E}_{(x,y)\sim p_T}[\text{bias}^2(x,y)] = \mathbb{E}_{(x,y)\sim p_T}\left(y - \bar{f}_S(x)\right)^2 \\
&= \int_{\mathcal{X}_T} \left(f_T(x) - \bar{f}_S(x)\right)^2 p_T(x)dx \\
&= \int_{\mathcal{X}_T \cap \mathcal{X}_S} \left(f_T(x) - \bar{f}_S(x)\right)^2 p_T(x)dx + \int_{\mathcal{X}_T \setminus \mathcal{X}_S} \left(f_T(x) - \bar{f}_S(x)\right)^2 p_T(x)dx.
\end{aligned}$$

To decompose the first term, we write $\forall x \in \mathcal{X}_S$, $-\bar{f}_S(x) = -f_S(x) + \left(f_S(x) - \bar{f}_S(x)\right)$.

$$\begin{aligned}
&\int_{\mathcal{X}_T \cap \mathcal{X}_S} \left(f_T(x) - \bar{f}_S(x)\right)^2 p_T(x)dx = \int_{\mathcal{X}_T \cap \mathcal{X}_S} \left((f_T(x) - f_S(x)) + \left(f_S(x) - \bar{f}_S(x)\right)\right)^2 p_T(x)dx \\
&= \int_{\mathcal{X}_T \cap \mathcal{X}_S} (f_T(x) - f_S(x))^2 p_T(x)dx + \int_{\mathcal{X}_T \cap \mathcal{X}_S} \left(f_S(x) - \bar{f}_S(x)\right)^2 p_T(x)dx \\
&+ \int_{\mathcal{X}_T \cap \mathcal{X}_S} 2\left(f_T(x) - f_S(x)\right)\left(f_S(x) - \bar{f}_S(x)\right) p_T(x)dx.
\end{aligned}$$

□

The four terms can be qualitatively analyzed:

- The first term measures differences between train and test labelling function. By rewriting $\forall x \in \mathcal{X}_T \cap \mathcal{X}_S$, $f_T(x) = \mathbb{E}_{p_T}[Y|X=x]$ and $f_S(x) = \mathbb{E}_{p_S}[Y|X=x]$, this term measures whether conditional distributions differ. This recovers a similar expression to the correlation shift formula from [Ye+22].

- The second term is exactly the ID bias, but weighted by the marginal distribution $p_T(X)$.

- The third term $\int_{\mathcal{X}_T \cap \mathcal{X}_S} 2\left(f_T(x) - f_S(x)\right)\left(f_S(x) - \bar{f}_S(x)\right) p_T(x) dx$ measures to what extent the ID bias compensates the correlation shift. It can be negative if (by chance) the ID bias goes in opposite direction to the correlation shift.

- The last term measures support mismatch between test and train marginal distributions. It lead to the "No free lunch for learning representations for DG" in [Rua+22]. The error is irreducible because "outside of the source domain, the label distribution is unconstrained": "for any domain which gives some probability mass on an example that has not been seen during training, then all [...] labels for that example" are possible.

C.1.2.2 OOD bias when small ID bias

We now prove Proposition 2.2 under Assumption 2.3.

Proof. We simplify the second and third terms from Appendix C.1.2.1 under Assumption 2.3.

The second term is $\int_{\mathcal{X}_T \cap \mathcal{X}_S} \left(f_S(x) - \bar{f}_S(x)\right)^2 p_T(x) dx$. Under Assumption 2.3, $|f_S(x) - \bar{f}_S(x)| \leq \epsilon$. Thus the second term is $\mathcal{O}(\epsilon^2)$.

The third term is $\int_{\mathcal{X}_T \cap \mathcal{X}_S} 2\left(f_T(x) - f_S(x)\right)\left(f_S(x) - \bar{f}_S(x)\right) p_T(x) dx$. As $f_T - f_S$ is bounded on $\mathcal{X}_S \cap \mathcal{X}_T$, $\exists K \geq 0$ such that $\forall x \in \mathcal{X}_S$,

$$|\left(f_T(x) - f_S(x)\right)\left(f_S(x) - \bar{f}_S(x)\right) p_T(x)| \leq K \left|f_S(x) - \bar{f}_S(x)\right| p_T(x) = \mathcal{O}(\epsilon) p_T(x).$$

Thus the third term is $\mathcal{O}(\epsilon)$.

Finally, note that we cannot say anything about $\bar{f}_S(x)$ when $x \in \mathcal{X}_T \setminus \mathcal{X}_S$. □

To prove the previous equality, we needed a bounded difference between labeling functions $f_T - f_S$ on $\mathcal{X}_T \cap \mathcal{X}_S$. We relax this bounded assumption to obtain an inequality in the following Proposition C.1.

Proposition C.1 (OOD bias when small ID bias without bounded difference between labeling functions)**.** *Under Assumption 2.3,*

$$\mathbb{E}_{(x,y)\sim p_T}[\mathrm{bias}^2(x,y)] \leq 2 \times \textit{Correlation shift} + \textit{Support mismatch} + \mathcal{O}(\epsilon^2) \quad \text{(C.3)}$$

Proof. We follow the same proof as in Appendix C.1.2.1, except that we now use: $(a+b)^2 \leq 2(a^2+b^2)$. Then,

$$\begin{aligned}
&\int_{\mathcal{X}_T\cap\mathcal{X}_S} \left(f_T(x) - \bar{f}_S(x)\right)^2 p_T(x)dx = \int_{\mathcal{X}_T\cap\mathcal{X}_S} \left((f_T(x) - f_S(x)) + \left(f_S(x) - \bar{f}_S(x)\right)\right)^2 p_T(x)dx \\
&\leq 2 \times \int_{\mathcal{X}_T\cap\mathcal{X}_S} (f_T(x) - f_S(x))^2 + \left(f_S(x) - \bar{f}_S(x)\right)^2 p_T(x)dx \\
&\leq 2 \times \int_{\mathcal{X}_T\cap\mathcal{X}_S} (f_T(x) - f_S(x))^2 p_T(x)dx + 2 \times \int_{\mathcal{X}_T\cap\mathcal{X}_S} \epsilon^2 p_T(x)dx \\
&\leq 2 \times \int_{\mathcal{X}_T\cap\mathcal{X}_S} (f_T(x) - f_S(x))^2 p_T(x)dx + \mathcal{O}(\epsilon^2)
\end{aligned}$$

□

C.1.3 Proof of the bias-variance-covariance decomposition Proposition 2.3

Proof. This proof recovers the bias-variance-covariance decomposition from [Ued+96; Bro+05a] of ensembling, with *i.d.* learning procedures.

With $\bar{f}_S(x) = \mathbb{E}_{l_S}[f(x, \theta(l_S))]$, we recall the bias-variance decomposition [Koh+96] (Equation (BV)):

$$\begin{aligned}
\mathbb{E}_{l_S}\mathcal{E}_T(\theta(l_S)) &= \mathbb{E}_{(x,y)\sim p_T}[\text{bias}^2(x,y) + \text{var}(x)], \\
\text{where } \text{bias}(x,y) &= \text{Bias}\{f|(x,y)\} = y - \bar{f}_S(x), \\
\text{and } \text{var}(x) &= \text{Var}\{f|x\} = \mathbb{E}_{l_S}\left[\left(f(x,\theta(l_S)) - \bar{f}_S(x)\right)^2\right].
\end{aligned}$$

Using $f_{\text{ENS}} = f_{\text{ENS}}(\cdot, \{\theta(l_S^{(i)})\}_{i=1}^M) = \frac{1}{M}\sum_{i=1}^M f(\cdot, \theta(l_S^{(i)}))$ in this decomposition yields,

$$\mathbb{E}_{L_S^M}\mathcal{E}_T(\{\theta(l_S^{(i)})\}_{i=1}^M) = \mathbb{E}_{x\sim p_T}\left[\text{Bias}\{f_{\text{ENS}} \mid (x,y)\}^2 + \text{Var}\{f_{\text{ENS}} \mid x\}\right]. \tag{C.4}$$

As f_{ENS} depends on L_S^M, we extend the bias into:

$$\text{Bias}\{f_{\text{ENS}} \mid (x,y)\} = y - \mathbb{E}_{L_S^M}\left[\frac{1}{M}\sum_{i=1}^M f(x, \theta(l_S^{(i)}))\right] = y - \frac{1}{M}\sum_{i=1}^M \mathbb{E}_{l_S^{(i)}}\left[f(x,\theta(l_S^{(i)}))\right]$$

Under *i.d.* $L_S^M = \{l_S^{(i)}\}_{i=1}^M$,

$$\frac{1}{M}\sum_{i=1}^M \mathbb{E}_{l_S^{(i)}}\left[y - f(x,\theta(l_S^{(i)}))\right] = \mathbb{E}_{l_S}[y - f(x,\theta(l_S))] = \text{Bias}\{f|(x,y)\}.$$

Thus the bias of ENS is the same as for a single member of the ensemble.

Regarding the variance:

$$\mathrm{Var}\left\{f_{\mathrm{ENS}} \mid x\right\} = \mathbb{E}_{L_S^M}\left[\left(\frac{1}{M}\sum_{i=1}^{M} f(x,\theta(l_S^{(i)})) - \mathbb{E}_{L_S^M}\left[\frac{1}{M}\sum_{i=1}^{M} f(x,\theta(l_S^{(i)}))\right]\right)^2\right].$$

Under *i.d.* $L_S^M = \{l_S^{(i)}\}_{i=1}^M$,

$$\begin{aligned}
\mathrm{Var}\left\{f_{\mathrm{ENS}} \mid x\right\} = &\frac{1}{M^2}\sum_{i=1}^{M}\mathbb{E}_{l_S}\left[\left(f(x,\theta(l_S)) - \mathbb{E}_{l_S}\left[f(x,\theta(l_S))\right]\right)^2\right] + \\
&\frac{1}{M^2}\sum_{i}\sum_{i'\neq i}\mathbb{E}_{l_S,l'_S}\left[\left(f(x,\theta(l_S)) - \mathbb{E}_{l_S}\left[f(x,\theta(l_S))\right]\right)\left(f(x,\theta(l'_S)) - \mathbb{E}_{l'_S}\left[f(x,\theta(l'_S))\right]\right)\right] \\
= &\frac{1}{M}\mathbb{E}_{l_S}\left[\left(f(x,\theta(l_S)) - \mathbb{E}_{l_S}\left[f(x,\theta(l_S))\right]\right)^2\right] + \\
&\frac{M-1}{M}\mathbb{E}_{l_S,l'_S}\left[\left(f(x,\theta(l_S)) - \mathbb{E}_{l_S}\left[f(x,\theta(l_S))\right]\right)\left(f(x,\theta(l'_S)) - \mathbb{E}_{l'_S}\left[f(x,\theta(l'_S))\right]\right)\right] \\
= &\frac{1}{M}\mathrm{var}\left(x\right) + \left(1 - \frac{1}{M}\right)\mathrm{cov}\left(x\right).
\end{aligned}$$

The variance is split into the variance of a single member divided by M and a new covariance. □

C.2 Theoretical insights for DICE

C.2.1 KL between Gaussians

The Kullback-Leibler divergence (KL) divergence [Kul59] between two Gaussian distributions takes a particularly simple form:

$$\begin{aligned} D_{\mathrm{KL}}(e(z|x)\|b(z|y)) &= \log\frac{b^{\sigma}(y)}{e^{\sigma}(x)} + \frac{e^{\sigma}(x)^2 + (e^{\mu}(x) - b^{\mu}(y))^2}{2b^{\sigma}(y)^2} - \frac{1}{2} && \text{(Gaussian param.)} \\ &= \frac{1}{2}[\underbrace{(1 + e^{\sigma}(x)^2 - \log(e^{\sigma}(x)^2))}_{\text{Variance}} + \underbrace{(e^{\mu}(x) - b^{\mu}(y))^2}_{\text{Mean}}]. && (b^{\sigma}(y) = \mathbb{1}) \end{aligned}$$

The variance component forces the predicted variance $e^{\sigma}(x)$ to be close to $b^{\sigma}(y) = \mathbb{1}$. The mean component forces the class-embedding $b^{\mu}(y)$ to converge to the average of the different elements in its class.

C.2.2 Difference between VCEB and VIB

In [Fis20], CEB is variationally upper bounded by VCEB. We detail the computations:

$$\begin{aligned} \mathrm{CEB}_{\beta_{ceb}}(Z) &= \frac{1}{\beta_{ceb}} I(X;Z|Y) - I(Y;Z) && \text{(Definition)} \\ &= \frac{1}{\beta_{ceb}}[I(X,Y;Z) - I(Y;Z)] - I(Y;Z) && \text{(Chain rule)} \\ &= \frac{1}{\beta_{ceb}}[I(X;Z) - I(Y;Z)] - I(Y;Z) && \text{(Markov assumptions)} \\ &= \frac{1}{\beta_{ceb}}[-H(Z|X) + H(Z|Y)] - [H(Y) - H(Y|Z)] && \text{(MI as diff. of 2 ent.)} \\ &\leq \frac{1}{\beta_{ceb}}[-H(Z|X) + H(Z|Y)] - [-H(Y|Z)] && \text{(Non-negativity of ent.)} \\ &= \int \{\frac{1}{\beta_{ceb}} \log\frac{e(z|x)}{p(z|y)} - \log p(y|z)\} p(x,y,z) \partial x \partial y \partial z && \text{(Definition of ent.)} \\ &\leq \int \{\frac{1}{\beta_{ceb}} \log\frac{e(z|x)}{b(z|y)} - \log d(y|z)\} p(x,y) e(z|x) \partial x \partial y \partial z && \text{(Variational approx.)} \\ &\approx \frac{1}{n_s}\sum_{n=1}^{n_s} \int \{\frac{1}{\beta_{ceb}} \log\frac{e(z|x_n)}{b(z|y_n)} - \log d(y_n|z)\} e(z|x_n) \partial z && \text{(Empirical data distrib.)} \\ &\approx \mathrm{VCEB}_{\beta_{ceb}}(\theta = \{e, b, d\}), && \text{(Reparameterization trick)} \end{aligned}$$

where, by introducing ϵ such as $z = e(x, \epsilon)$,

$$\mathrm{VCEB}_{\beta_{ceb}}(\theta = \{e,b,d\}) = \frac{1}{n_s}\sum_{n=1}^{n_s}\{\frac{1}{\beta_{ceb}} D_{\mathrm{KL}}(e(z|x_n)\|b(z|y_n)) - \mathbb{E}_{\epsilon} \log d(y_n|e(x_n,\epsilon)\}.$$

As a reminder, [Ale+17] upper bounded: $\mathrm{IB}_{\beta_{ib}}(Z) = \frac{1}{\beta_{ib}} I(X;Z) - I(Y;Z)$ by:

$$\mathrm{VIB}_{\beta_{ib}}(\theta = \{e,b,d\}) = \frac{1}{n_s}\sum_{n=1}^{n_s}\{\frac{1}{\beta_{ib}} D_{\mathrm{KL}}(e(z|x_n)\|b(z)) - \mathbb{E}_{\epsilon} \log d(y_n|e(x_n,\epsilon)\}. \tag{C.5}$$

In VIB, all features distribution $e(z|x)$ are moved towards the same class-agnostic distribution $b(z) \sim N(\mu, \sigma)$, independently of y. In VCEB, $e(z|x)$ are moved towards the class conditional marginal $b^{\mu}(y) \sim N(b^{\mu}(y), b^{\sigma}(y))$. This is the **unique difference between VIB and VCEB**. VIB leads to a looser approximation with more bias than VCEB.

C.3 Proof for DIVERSE WEIGHT AVERAGING

Below we detail the (simple) proof of Lemma 4.1, validating the similarity between weight averaging and functional ensembling.

Proof. **Functional approximation.** With a Taylor expansion at the first order of the models' predictions w.r.t. parameters θ:

$$\begin{aligned} f_{\theta_i} &= f_{\text{WA}} + \nabla f_{\text{WA}}^{\intercal} \Delta_i + O\left(\|\Delta_i\|_2^2\right) \\ f_{\text{ENS}} - f_{\text{WA}} &= \frac{1}{M} \sum_{i=1}^{M} \left(\nabla f_{\text{WA}}^{\intercal} \Delta_i + O\left(\|\Delta_i\|_2^2\right)\right) \end{aligned}$$

Therefore, because $\sum_{i=1}^{M} \Delta_i = 0$,

$$f_{\text{ENS}} - f_{\text{WA}} = O\left(\Delta^2\right) \text{ where } \Delta = \max_{i=1}^{M} \|\Delta_i\|_2 . \tag{C.6}$$

Loss approximation. With a Taylor expansion at the zeroth order of the loss w.r.t. its first input and injecting Equation (C.6):

$$\begin{aligned} \ell\left(f_{\text{ENS}}(x); y\right) &= \ell\left(f_{\text{WA}}(x); y\right) + O\left(\|f_{\text{ENS}}(x) - f_{\text{WA}}(x)\|_2\right) \\ \ell\left(f_{\text{ENS}}(x); y\right) &= \ell\left(f_{\text{WA}}(x); y\right) + O\left(\Delta^2\right) . \end{aligned}$$

□

C.4 Proofs for REWARDED SOUPS

In Chapter 6, we proved the Pareto optimality for Hessians proportional to the identity. We now consider the more complex case with the relaxed Assumption C.1. For simplicity, we only consider $N = 2$ rewards R_1 and R_2.

Assumption C.1 (Diagonal Hessians)**.** *The rewards are quadratic, with Hessians diagonal negative definite. Specifically, we can write for $i \in \{1, 2\}$:*

$$\forall \theta = (\theta^1, \ldots, \theta^d) \in \Theta, \quad R_i(\theta) = R_i(\theta_i) - \sum_{j=1}^{d} \eta_i^j (\theta^j - \theta_i^j)^2, \tag{C.7}$$

where $(\eta_i^1, \ldots \eta_i^d) \in \{\mathbb{R}_+^*\}^d$ and $\theta_i = (\theta_i^1, \ldots, \theta_i^d)$ is the global maximum for reward R_i.

Remark C.1. *This diagonal Assumption C.1 of the Hessian is common: for example in optimization [LeC+12; Kin+15], to prune networks [LeC+90] or in out-of-distribution generalization [Ram+22a]. This strong assumption is supported by the empirical observation [Bec+88] that Hessians are diagonally dominant, in particular at the end of training. Also, we note that our findings remain valid assuming only that the Hessians are co-diagonalizable.*

Lemma C.2. *We consider the user's reward $R_{\hat{\mu}} = (1 - \hat{\mu}) \times R_1 + \hat{\mu} \times R_2$ with $\hat{\mu} \in [0, 1]$, and*

$$\Delta R_{\hat{\mu}} = \max_{\theta \in \Theta} R_{\hat{\mu}}(\theta) - \max_{\lambda \in [0,1]} R_{\hat{\mu}} \left((1 - \lambda) \cdot \theta_1 + \lambda \cdot \theta_2 \right). \tag{C.8}$$

$\Delta R_{\hat{\mu}}$ corresponds to the difference in terms of $R_{\hat{\mu}}$ between the global maximum and the maximum reachable by weight interpolation through rewarded soups (with a single interpolating coefficient for all dimensions). Then, under Assumption C.1, we have:

$$\Delta R_{\hat{\mu}} \leq \frac{\hat{\mu}^2 (1 - \hat{\mu})^2 (M \Delta_1 - \Delta_2)(M \Delta_2 - \Delta_1)}{\left(\hat{\mu}(1 - \hat{\mu})(M - 1)^2 + M\right) \left((1 - \hat{\mu}) \Delta_1 + \hat{\mu} \Delta_2 \right)}, \tag{C.9}$$

where $M = \max_{j \in \{1, \ldots, d\}} \max \left(\frac{\eta_1^j}{\eta_2^j}, \frac{\eta_2^j}{\eta_1^j} \right)$ is the maximum of eigenvalues ratio, $\Delta_1 = R_1(\theta_1) - R_1(\theta_2)$ and $\Delta_2 = R_2(\theta_2) - R_2(\theta_1)$.

When $\Delta_1 = \Delta_2$, the bound simplifies into:

$$\Delta R_{\hat{\mu}} \leq \frac{\hat{\mu}^2 (1 - \hat{\mu})^2 (M - 1)^2}{\hat{\mu}(1 - \hat{\mu})(M - 1)^2 + M} \Delta_1 \tag{C.10}$$

Furthermore, when the Hessians are equal, then $M = 1$ and $\Delta R_{\hat{\mu}} = 0$: RS is optimal .

Proof. This novel proof is in three steps. First, we find $\hat{\theta}$ maximizing $R_{\hat{\mu}}(\theta)$ for θ on the full set of weights Θ. Second, we find $\bar{\lambda}$ maximizing $R_{\hat{\mu}} \left((1 - \lambda) \cdot \theta_1 + \lambda \cdot \theta_2 \right)$ for $\lambda \in [0, 1]$ and thus defining the best interpolation between the expert weights. Finally, we bound $\Delta R_{\hat{\mu}}$, the differences between their rewards, by applying the Bhatia-Davis inequality.

First step. Let's first find the maximum of $R_{\hat{\mu}}$ on Θ. Denoting $S = (1-\hat{\mu}) \times R_1(\theta_1) + \hat{\mu} \times R_2(\theta_2)$, we have for all $\theta \in \Theta$:

$$R_{\hat{\mu}}(\theta) = S - \sum_{j=1}^{d} \left((1-\hat{\mu})\eta_1^j \left(\theta^j - \theta_1^j\right)^2 + \hat{\mu}\eta_2^j \left(\theta^j - \theta_2^j\right)^2 \right) \tag{C.11}$$

Since $R_{\hat{\mu}}$ is a sum of concave quadratic functions, it has a unique global maximum reached at a point we note $\hat{\theta} = \left(\hat{\theta}^1, \ldots, \hat{\theta}^d\right)$. The global maximum can be computed by differentiating $R_{\hat{\mu}}$ with respect to each variable θ^j, which gives:

$$\hat{\theta}^j = \left(1 - \hat{\lambda}^j\right) \cdot \theta_1^j + \hat{\lambda}^j \cdot \theta_2^j$$

where the interpolating coefficients per dimension $\hat{\lambda}^j$ are defined for $j \in \{1, \ldots, d\}$ as:

$$\hat{\lambda}^j = \frac{\hat{\mu}\eta_2^j}{(1-\hat{\mu})\eta_1^j + \hat{\mu}\eta_2^j} \in [0, 1]. \tag{C.12}$$

Second step. With $\lambda \in [0,1]$ and $\theta = (1-\lambda) \cdot \theta_1 + \lambda \cdot \theta_2$, we can write $R_{\hat{\mu}}(\theta)$ as a function of λ:

$$\begin{aligned} R_{\hat{\mu}}(\theta) &= S - \sum_{j=1}^{d} \left(\left((1-\hat{\mu})\,\eta_1^j + \hat{\mu}\eta_2^j \right) \left(\lambda - \hat{\lambda}^j\right)^2 + \frac{\hat{\mu}(1-\hat{\mu})\eta_1^j\eta_2^j}{(1-\hat{\mu})\eta_1^j + \hat{\mu}\eta_2^j} \right) \left(\theta_1^j - \theta_2^j\right)^2 \\ &= R_{\hat{\mu}}(\hat{\theta}) - \sum_{j=1}^{d} p_j \left(\lambda - \hat{\lambda}^j\right)^2 \end{aligned} \tag{C.13}$$

where p_j is defined as $p_j = \left((1-\hat{\mu})\,\eta_1^j + \hat{\mu}\eta_2^j \right) \left(\theta_1^j - \theta_2^j\right)^2$.

From Equation (C.13), we can compute the maximum reward obtainable for weight averaging $\max_{\lambda \in [0,1]} R_{\hat{\mu}}\left((1-\lambda) \cdot \theta_1 + \lambda \cdot \theta_2\right)$. Since the function $\lambda \mapsto R_{\hat{\mu}}\left((1-\lambda) \cdot \theta_1 + \lambda \cdot \theta_2\right)$ is a concave quadratic function, there is a unique value $\bar{\lambda}$ maximizing $R_{\hat{\mu}}$ equal to

$$\bar{\lambda} = \frac{\sum_{j=1}^{d} p_j \hat{\lambda}^j}{\sum_{j=1}^{d} p_j}. \tag{C.14}$$

Since all p_j are positive and all $\hat{\lambda}^j$ are between 0 and 1, $\bar{\lambda}$ is also between 0 and 1. Therefore, $R_{\hat{\mu}}\left((1-\bar{\lambda}) \cdot \theta_1 + \bar{\lambda} \cdot \theta_2\right)$ is indeed the maximum reward for rewarded soups.

Third step. Applying Equation (C.13) to $\bar{\lambda}$ gives:

$$\Delta R_{\hat{\mu}} = R_{\hat{\mu}}(\hat{\theta}) - R_{\hat{\mu}}\left((1-\bar{\lambda})\cdot\theta_1 + \bar{\lambda}\cdot\theta_2\right) \tag{C.15}$$

$$= \sum_{j=1}^{d} p_j \left(\bar{\lambda} - \hat{\lambda}^j\right)^2 \tag{C.16}$$

$$= \left(\sum_{j=1}^{d} \frac{p_j}{\sum_{i=1}^{n} p_i}\left(\bar{\lambda} - \hat{\lambda}^j\right)^2\right)\left(\sum_{j=1}^{n} p_j\right) \tag{C.17}$$

The second term in Equation (C.17) can be simplified as:

$$\sum_{j=1}^{d} p_j = (1-\hat{\mu})\Delta_1 + \hat{\mu}\Delta_2. \tag{C.18}$$

The core component of this proof is the upper bounding of the first term in Equation (C.17). The key idea is to recognize the variance of a discrete random variable Λ with $\mathbb{P}(\Lambda = \hat{\lambda}_i) = \frac{p_i}{\sum_{j=1}^{n} p_j}$; then, $\bar{\lambda}$ from Equation (C.14) is actually the expectation of Λ. Then, we can apply the **Bhatia-Davis inequality**, as recalled in Equation (C.19), on the variance of a bounded random variable $a \leq \Lambda \leq b$:

$$Var(\Lambda) \leq (b - \mathbb{E}(\Lambda))\,(\mathbb{E}(\Lambda) - a) \tag{C.19}$$

Therefore Equation (C.17) is bounded by:

$$\Delta R_{\hat{\mu}} \leq \left(\max_{1\leq j\leq d} \hat{\lambda}^j - \bar{\lambda}\right)\left(\bar{\lambda} - \min_{1\leq j\leq d} \hat{\lambda}^j\right)\left((1-\hat{\mu})\,\Delta_1 + \hat{\mu}\Delta_2\right). \tag{C.20}$$

Now, we bound the variables $\hat{\lambda}^j$, since $1/M \leq \eta_1^j/\eta_2^j \leq M$. Then for all j we have:

$$\frac{\hat{\mu}}{(1-\hat{\mu})M + \hat{\mu}} \leq \hat{\lambda}^j \leq \frac{\hat{\mu}M}{(1-\hat{\mu}) + \hat{\mu}M}, \tag{C.21}$$

and thus:

$$\Delta R_{\hat{\mu}} \leq \left(\frac{\hat{\mu}M}{1+\hat{\mu}(M-1)} - \bar{\lambda}\right)\left(\bar{\lambda} - \frac{\hat{\mu}}{M - \hat{\mu}(M-1)}\right)\left((1-\hat{\mu})\,\Delta_1 + \hat{\mu}\Delta_2\right). \tag{C.22}$$

Finally, noting that $\Delta_i = \sum_{j=1}^{d} \eta_i^j \left(\theta_2^j - \theta_1^j\right)^2$, we deduce from Equation (C.14) that $\bar{\lambda} = \frac{\hat{\mu}\Delta_2}{(1-\hat{\mu})\Delta_1 + \hat{\mu}\Delta_2}$. Replacing this in the previous Equation (C.22) gives the final Equation (C.9), concluding the proof. □

Remark C.2. *As a final remark, please note that the suboptimality of RS comes from the need of having one single interpolating coefficient $\bar{\lambda}$ for all d parameters $(\theta^1, \dots, \theta^d)$ of the network. Yet, the advanced merging operations in [Mat+22] remove this constraint, with interpolating coeffi-*

cients proportional to the eigenvalues of the Fisher matrices [Fis22], which actually approximate the eigenvalues of the Hessian [Sch02; Tho+20]. Combining [Mat+22] and our RS is a promising research direction, the key issue being the computation of the Fisher matrices [Kun+19] for networks with billions of parameters.

We visualize in Figure C.1 the bound given by Lemma C.2. We show that for small values of M like $M = 2$, the value of $R_{\hat{\mu}}$ for RS is quite close to the global optimum. Also, recall that RS theoretically matches this upper bound when $M = 1$. For larger values like $M = 10$, the bound is less tight, and we note that the maximum value of $R_{\hat{\mu}}$ approaches the constant function 1 as $M \to \infty$.

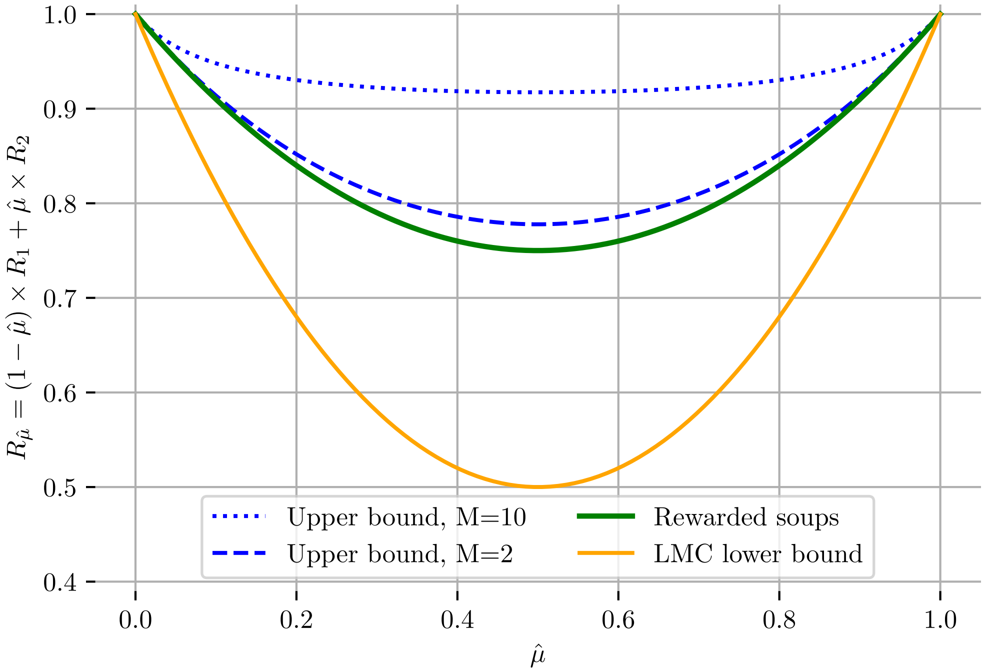


Figure C.1. – Illustration of the bound given by Lemma C.2 under Assumption C.1. For simplicity, we showcase the case where $R_1(\theta_1) = R_2(\theta_2) = 1$, $R_1(\theta_2) = R_2(\theta_1) = 0$, thus $\Delta_1 = \Delta_2 = 1$. In green, we plot the rewards obtained with rewarded soups for the optimal $\bar{\lambda}$, *i.e.*, $R_{\hat{\mu}}\big((1-\bar{\lambda}) \cdot \theta_1 + \bar{\lambda} \cdot \theta_2\big)$, whose value is independent of M in this case. In blues, we plot the maximum value of $\mathcal{R}_{\hat{\mu}}$ given by Equation (C.10) in Lemma C.2, for $M = 2$ and $M = 10$. For reference, we also plot the values for the lower bound in the LMC Hypothesis 6.1, *i.e.*, equal to $(1-\hat{\mu})(1-\bar{\lambda})R_1(\theta_1) + \hat{\mu}\bar{\lambda}R_2(\theta_2)$. As RS outperforms this lower bound, it validates Hypothesis 6.1 in this case.

Appendix D

MIXMO: MIXING MULTIPLE INPUTS FOR MULTIPLE OUTPUTS VIA DEEP SUBNETWORKS

D.1 Introduction

In deep learning (DL), to improve robustness [Hen+19a; Ova+19] or to win Kaggle competitions, DNNs usually pair up with two practical strategies.

The first strategy is data augmentation (DA), reducing overfitting by diversifying the training samples [Lop+20]: in particular, recent mixed sample data augmentation (MSDA) create artificial samples by mixing multiple inputs and their labels proportionally to a ratio λ. The seminal Mixup [Zha+18a] linearly interpolates pixels: binary masking MSDAs [Fre+20; Har+20; Kim+20] such as CutMix [Yun+19] have since diversified mixed samples by pasting patches from one image onto another.

The second strategy is ensembling (ENS), the main topic of this thesis. This chapter follows the recent multi-input multi-output (MIMO) [Hav+21; Sof+20], that approximates traditional functional ensembling of models by fitting M independent subnetworks inside a single base network: this is possible as large networks only leverage a subset of their weights [Fra+19]. Specifically, we introduce MixMo, a new generalized framework for learning deep subnetworks. As in MIMO, we consider M (input, label) pairs at the same time in training: $\{(x_i, y_i)\}_{0\leq i<M}$, as shown on Figure D.1 with $M = 2$. The M inputs are encoded by M separate convolutional layers $\{c_i\}_{0\leq i<M}$ into a shared latent space before being mixed. The representation is then fed to the core network, which finally branches out into M dense layers $\{d_i\}_{0\leq i<M}$. The main idea during training to prevent homogenization is that each subnetwork learns to classify only one of the multiple inputs simultaneously provided. Diverse subnetworks naturally emerge as d_i learns to classify y_i from input x_i. At inference, the same image is repeated M times: we obtain ensembling "for free" by averaging M predictions.

Our key originality lies in the *multi-input mixing block*; indeed, the question of how to best mix these multiple inputs has not been studied so far. Should the merging be a basic summation, we would recover MIMO [Hav+21]. Our main intuition is simple: we see summing as a balanced and restrictive form of Mixup [Zha+18a] where $\lambda = \frac{1}{M}$. Then by analogy, we leverage the literature in MSDA to propose novel mixing strategies. In particular, we show that binary masking methods—particularly with rectangular patches from CutMix[Yun+19]—enhances results by making subnetworks stronger and more diverse. We thus create a new Cut-MixMo variant inspired by CutMix, and illustrated in Figure D.1: a patch of features from the first input is pasted into the features from

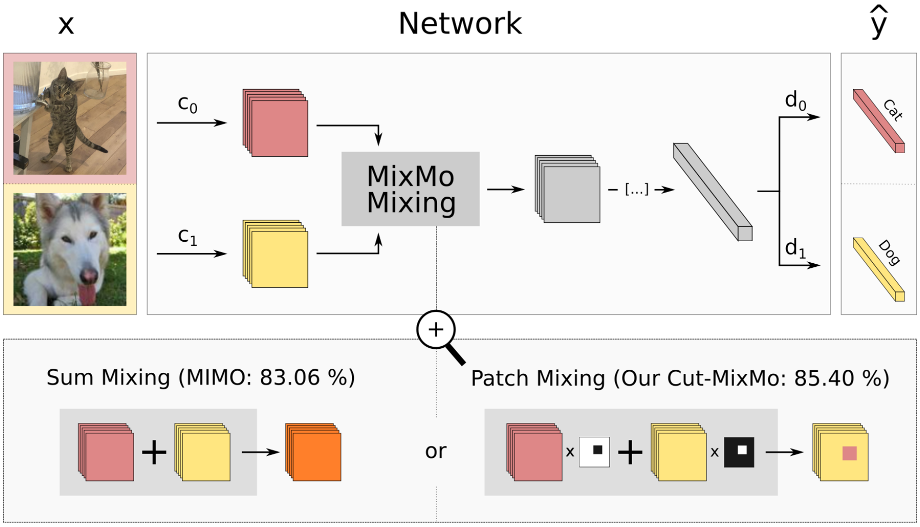

x
Network
ŷ
c0
c1
MixMo
Mixing
– [...] →
d0
d1
Cat
Dog
Sum Mixing (MIMO: 83.06 %)
or
Patch Mixing (Our Cut-MixMo: 85.40 %)
x
x

Figure D.1. – MixMo framework. We embed $M = 2$ inputs into a shared space with convolutional layers (c_1, c_2), mix them, pass the embedding through further layers and output 2 predictions via dense layers (d_1, d_2). The key point of our MixMo is the mixing block. Mixing with patches performs better than basic summing: 85.40% vs. 83.06% (MIMO [Hav+21]) on CIFAR-100 with WRN-28-10.

the second input. Overall, we replace the suboptimal summing operation hidden in MIMO with an improved mixing block based on patching. Our asymmetrical mixing also raises new questions regarding information flow in the network's features; we tackle the imbalance between the multiple classification training tasks via a new weighting scheme.

1. We propose a general framework, MixMo, connecting two successful fields: mixing samples data augmentations & multi-input multi-output ensembling (Appendix D.3.1).

2. We identify the appropriate mixing block to best tackle the diversity/individual accuracy trade-off in subnetworks: our easy to implement Cut-MixMo benefits from the synergy between CutMix and ensembling (Appendix D.3.2).

3. We design a new weighting of the loss components to properly leverage the asymmetrical inputs mixing (Appendix D.3.3).

4. We improve performances for image classification on CIFAR-100 and Tiny ImageNet datasets (Appendix D.4). As exhibited by Figure D.2, Cut-MixMo outperforms CutMix, MIMO and deep ensembling (DE), at (almost) the same inference cost as a single network.

The work in this chapter has led to the publication of the following paper: Alexandre Ramé, Remy Sun, and Matthieu Cord. "MixMo: Mixing Multiple Inputs for Multiple Outputs via Deep Subnetworks". In: *ICCV*. 2021.

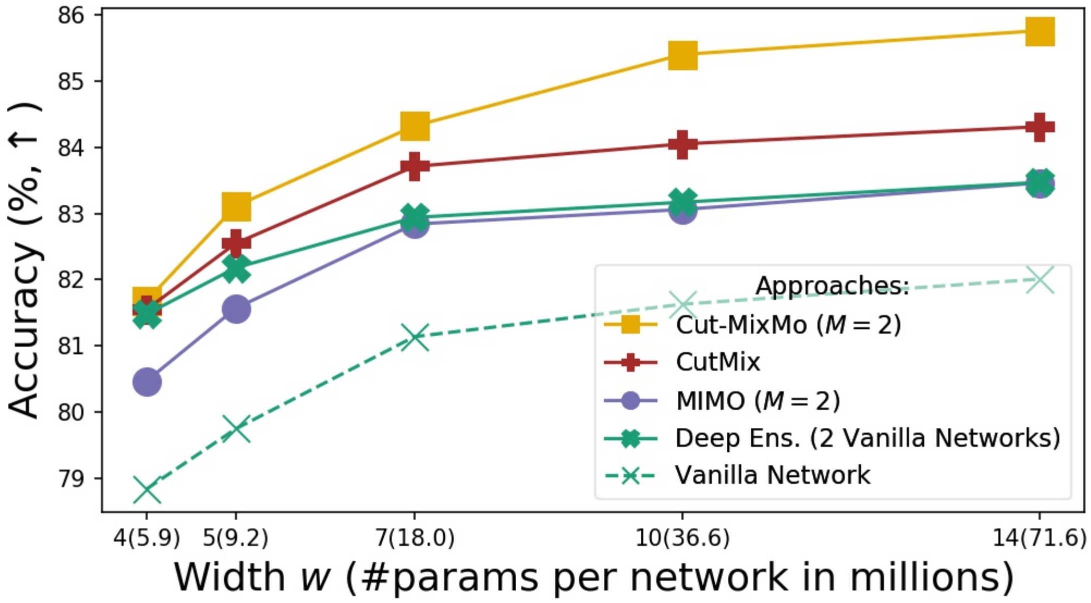


Figure D.2. – Main MixMo results on CIFAR-100 with WRN-28-w. Our Cut-MixMo variant (patch mixing and $M = 2$) surpasses CutMix and deep ensembles (with half the parameters) by leveraging over-parameterization in wide networks.

D.2 Related work

Data augmentation. DNNs are known to memorize the training data [Zha+17] and make overconfident predictions [Guo+17] to the detriment of generalization on new test examples. Data augmentations (DA) inflate the training dataset's size by creating artificial samples from available labeled data. Beyond slight perturbations (*e.g.*, rotation), recent works [Cub+20; Hen+19b; DeV+17b] apply stronger transformations [He+19]. Mixed sample data augmentation (MSDA) recently expanded the notion of DA. From pairs of labeled samples $\{(x_i, y_i), (x_k, y_k)\}$, they create virtual samples: $(m_x(x_i, x_k, \lambda), \lambda y_i + (1 - \lambda) y_k)$ where $\lambda \sim \text{Beta}(\alpha, \alpha)$. [Lia+18] shows that mixing the targets differently than this linear interpolation may cause underfitting and unstable learning. Then, approaches mainly focus on developing the most effective input mixing m_x. In [Ino18; Tok+18a; Tok+18b; Zha+18a], m_x performs a simple linear interpolation between pixels: *e.g* in Mixup [Zha+18a], $m_x(x_i, x_k, \lambda) = \lambda x_i + (1 - \lambda) x_k$. Then, CutMix [Yun+19] draws from Mixup and CutOut [DeV+17b] by pasting a patch from x_k onto x_i: $m_x(x_i, x_k, \lambda) = \mathbb{1}_m \odot x_i + (\mathbb{1} - \mathbb{1}_m) \odot x_k$, where $\odot$ represents the element-wise product and $\mathbb{1}_m$ a binary mask with average value λ. CutMix randomly samples squares, which often leads to rectangular masks due to boundary effects. Such *non-linear binary masking* improves generalization [Sum+19; Tak+20] by creating new images with usually disjoint patches [Har+20]. Finally, in addition to Manifold Mixup [Ver+19], only a few works [DeV+17a; Far+20; Li+20a; Yag+19; Yun+19] have tried to mix intermediate *latent features* as we do.

Ensembling. Like [Wen+21], we explore combining DA with another standard technique in machine learning: ensembling, whose fundamental drawback is the inherent *computational and memory overhead*. This chapter follows the multi-input multi-output MIMO paper [Hav+21], which achieves *ensemble almost "for free"*: all of the layers except the first convolutional and last dense layers are shared ($\approx +1\%$ #parameters).

D.3 MixMo

We first introduce the main components of our MixMo strategy, summarized in Figure D.3: we mix multiple inputs to obtain multiple outputs via subnetworks. We highlight the key mixing block combining information from inputs, and our training loss based on a dedicated weighting scheme. We study $M = 2$ subnetworks in this thesis, both for clarity and as it empirically performs best in standard parameterization regimes.

D.3.1 General overview

We leverage a training classification dataset $\mathcal{D}_S$ of *i.i.d.* pairs of associated image/label $\{x_i, y_i\}_{i=1}^{n_S}$. We randomly sample a subset of b samples $\{x_i, y_i\}_{i\in B}$ that we randomly shuffle via permutation π. Our training batch is $\{(x_i, x_j), (y_i, y_j)\}_{i\in B, j=\pi(i)}$. The loss $\mathcal{L}_{\text{MixMo}}$ is averaged over these b samples: the networks' weights are updated through backpropagation and gradient descent.

Let's focus on the training sample $\{(x_0, x_1), (y_0, y_1)\}$. In MixMo, both inputs are *separately encoded* (see Figure D.1) into the shared latent space with two different convolutional layers (with 3 input channels each and no bias term): x_0 via c_0 and x_1 via c_1. To recover a strictly equivalent formulation to MIMO [Hav+21], we simply sum the two encodings: $c_0(x_0) + c_1(x_1)$. Indeed, MIMO merges inputs through channel-wise concatenation in pixels: MIMO's first convolutional layer (with 6 input channels and no bias term) hides the summing operation in the output channels.

Explicitly highlighting the underlying mixing leads us to consider a *generalized multi-input mixing block* $\mathcal{M}$. This manifold mixing presents a unique opportunity to tackle the ensemble diversity/individual accuracy trade-off and to improve overall ensemble results (see Appendix D.3.2). The shared representation $\mathcal{M}\left(c_0(x_0), c_1(x_1)\right)$ feeds the next convolutional layers. We note κ the *mixing ratio* between inputs.

The core network $\mathcal{C}$ handles features that represent both inputs simultaneously. The dense layer d_0 predicts $\hat{y}_0 = d_0\left[\mathcal{C}\left(\mathcal{M}\left\{c_0\left(x_0\right), c_1\left(x_1\right)\right\}\right)\right]$ and targets y_0, while d_1 targets y_1. Thus, the *training loss* is the sum of two cross-entropies $\mathcal{L}_{\text{CE}}$ weighted by parametrized function w_r (defined in Appendix D.3.3) to balance the asymmetry when $\kappa \neq 0.5$:

$$\mathcal{L}_{\text{MixMo}} = w_r(\kappa) \times \mathcal{L}_{CE}\left(y_0, \hat{y}_0\right) + w_r(1-\kappa) \times \mathcal{L}_{CE}\left(y_1, \hat{y}_1\right). \tag{D.1}$$

At inference, the same input x is repeated twice: the core network $\mathcal{C}$ is fed the sum $c_0(x) + c_1(x)$ that preserves maximum information from both encodings. Then, the diverse predictions are averaged: $\frac{1}{2}(\hat{y}_0 + \hat{y}_1)$. This allows us to benefit from functional ensembling in a single forward pass.

D.3.2 Mixing block

The mixing block $\mathcal{M}$, combining both inputs into a shared representation, is the cornerstone of MixMo. Our main intuition was to analyze MIMO as a simplified Mixup variant where the mixing ratio κ is fixed to 0.5. MixMo generalized framework encompasses a wider range of variants inspired by MSDA mixing methods. Our first main variant, named Linear-MixMo, fully extends Mixup with the following mixing block:

$$\mathcal{M}_{\text{Linear-MixMo}}\left(l_0, l_1\right) = 2\left[\kappa l_0 + (1-\kappa) l_1\right], \tag{D.2}$$

where $l_0 = c_0(x_0)$, $l_1 = c_1(x_1)$ and $\kappa \sim \text{Beta}(\alpha, \alpha)$ and α the concentration parameter. The second and more effective *Cut-MixMo* variant adapts the patch mixing from CutMix:

$$\mathcal{M}_{\text{Cut-MixMo}}\left(l_0, l_1\right) = 2\left[\mathbb{1}_{\mathcal{M}} \odot l_0 + (\mathbb{1} - \mathbb{1}_{\mathcal{M}}) \odot l_1\right], \tag{D.3}$$

where $\mathbb{1}_{\mathcal{M}}$ is a binary mask with area ratio $\kappa \sim \text{Beta}(\alpha, \alpha)$, valued at 1 either on a rectangle or on the complementary of a rectangle. In brief, a patch from $c_0(x_0)$ is pasted onto $c_1(x_1)$, or vice versa. This binary mixing in Cut-MixMo advantageously replaces the linear interpolation in MIMO and Linear-MixMo: subnetworks are more accurate and more diverse, as shown empirically in Figure D.5.

First, binary mixing in $\mathcal{M}$ trains stronger *individual* subnetworks for the same reasons why CutMix improves over Mixup. By masking features, we simulate common object occlusion problems. This spreads subnetworks' focus across different locations: the two classifiers are forced to find information relevant to their assigned input at disjoint locations. This occlusion remains effective as the receptive field in this first shallow latent space remains small.

Secondly, linear interpolation is fundamentally ill-suited to induce diversity as full information is preserved from both inputs. CutMix on the other hand explicitly increases dataset diversity by presenting patches of images that do not normally appear together. Such benefits can be directly transposed to $\mathcal{M}_{\text{Cut-MixMo}}$: binary mixing with patches increases randomness and *diversity between the subnetworks*. Indeed, in a similar spirit to bagging [Bre96], different samples are given to the subnetworks. By deleting asymmetrical complementary locations from the two inputs, subnetworks will not rely on the same region and information. Overall, they are less likely to collapse on close solutions.

D.3.3 Loss weighting

Asymmetries in the mixing mechanism can cause one input to overshadow the other. Notably when $\kappa \neq 0.5$, the predominant input may be easier to predict. We seek a weighting function w_r to *balance the relative importance* of the two $\mathcal{L}_{\text{CE}}$ in $\mathcal{L}_{\text{MixMo}}$. This

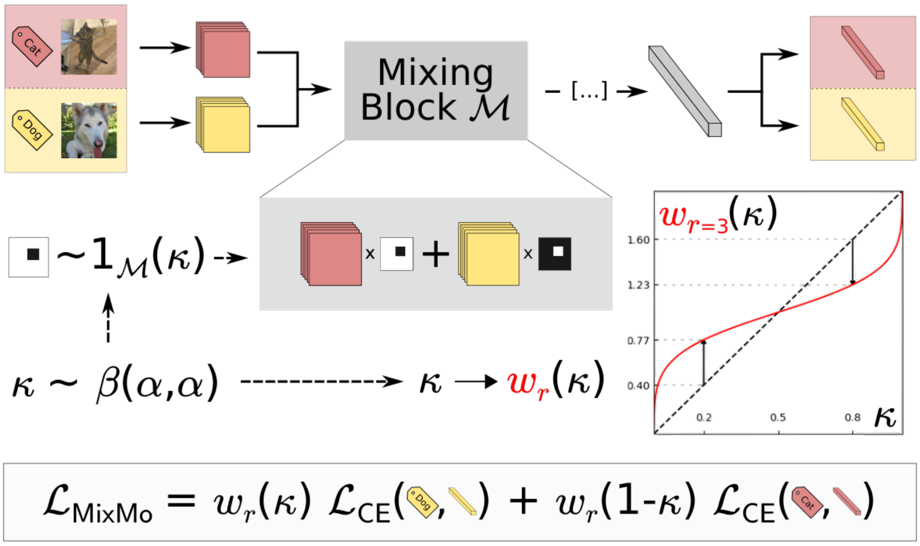

Cat
Dog
Mixing Block $\mathcal{M}$
[...]
$\sim 1_{\mathcal{M}}(\kappa)$
$\kappa \sim \beta(\alpha,\alpha)$
$\kappa \rightarrow w_r(\kappa)$
$w_{r=3}(\kappa)$
1.60
1.23
0.77
0.40
0.2
0.5
0.8
κ
$\mathcal{L}_{\text{MixMo}} = w_r(\kappa)\, \mathcal{L}_{\text{CE}}(\ ,\ ) + w_r(1-\kappa)\, \mathcal{L}_{\text{CE}}(\ ,\ )$

Figure D.3. – **Cut-MixMo training**. We sample a mixing mask given κ, and balance the losses with $w_r(\kappa)$ from Eq. D.4.

weighting modifies the effective learning rate, how gradients flow in the network and overall how mixed information is represented in features. We then propose to weight via the parametrized:

$$w_r(\kappa) = 2\frac{\kappa^{1/r}}{\kappa^{1/r} + (1-\kappa)^{1/r}}. \tag{D.4}$$

This defines a family of functions indexed by the parameter r, visualized for $r = 3$ in red on Figure D.3. This power law provides a natural relaxation between *two extreme configurations*. The first extreme, $r = 1$, $w_1(\kappa) = 2\kappa$, is in line with linear label interpolation in MSDA. The resulting imbalance in each subnetwork's contribution to $\mathcal{L}_{\text{MixMo}}$ causes lopsided updates. While it promotes diversity, it also reduces regularization: the overshadowed input has a reduced impact on the loss. The opposite extreme, $r \rightarrow \infty$, $w_\infty(\kappa) \rightarrow 1$, removes reweighting. Consequently, w_r inflates the importance of hard under-represented inputs, *à la* Focal Loss [Lin+17]. However, minimizing the role of the predominant inputs destabilizes training. Overall, we empirically observe that moderate values of r perform best as they trade off pros and cons from both extremes.

D.4 Experiments

D.4.1 Implementation details

We evaluate MixMo efficiency on CIFAR-{10,100} [Kri+09], and also consider Tiny ImageNet [Chr+17a] in the paper. We mostly study the Linear-MixMo and Cut-MixMo variants with $M = 2$, with the following hyperparameters values: $r = 3$ and $\alpha = 2$.

MIMO [Hav+21] refers to linear summing, like Linear-MixMo, but with $\kappa = 0.5$ instead of $\kappa \sim \text{Beta}(\alpha, \alpha)$. Other hyperparameters are taken directly from MIMO [Hav+21].

Different mixing methods create a strong *train-test distribution gap* [Car+20; Lop+20]. Thus, in Cut-MixMo we actually substitute $\mathcal{M}_{\text{Cut-MixMo}}$ for $\mathcal{M}_{\text{Linear-MixMo}}$ with probability $1 - p$ to accommodate for the summing in $\mathcal{M}$ at inference. We set the probability of patch mixing during training to $p = 0.5$, with linear descent to 0 over the last twelfth of training epochs. When MixMo is combined with CutMix, the pixels inputs are: $(m_x(x_i, x_k, \lambda), m_x(x_j, x_{k'}, \lambda'))$ with interpolated targets $(\lambda y_i + (1-\lambda) y_k, \lambda' y_j + (1-\lambda') y_{k'}))$, where k, k' are randomly sampled and $\lambda, \lambda' \sim \text{Beta}(1, 1)$. MIMO duplicates samples b times via *batch repetition*: x_i will be associated with $x_{\pi(i)}$ and $x_{\pi'(i)}$ in the same batch if $b = 2$. As the batch size remains fixed, the count of unique samples per batch and the learning rate is divided by b. Conversely, the number of steps is multiplied by b. Overall, this stabilizes training but multiplies its cost by b. We thus indicate an estimated (training/inference) overhead (w.r.t. vanilla training) in the *time* column of our tables. Note that some concurrent approaches also lengthen training: *e.g.*, GradAug [Yan+20c] via multiple subnetworks predictions ($\approx \times 3$). We equally track accuracies (Top{1,5}, ↑) and the calibrated negative log-likelihood (NLL$_c$) (↓). Indeed, [Ash+20] shows that we should compare in-domain uncertainty estimations after temperature scaling (TS) [Guo+17]: we thus split the test set in two and calibrate (after averaging in ensembles) with the temperature optimized on the other half.

D.4.2 Main results on CIFAR

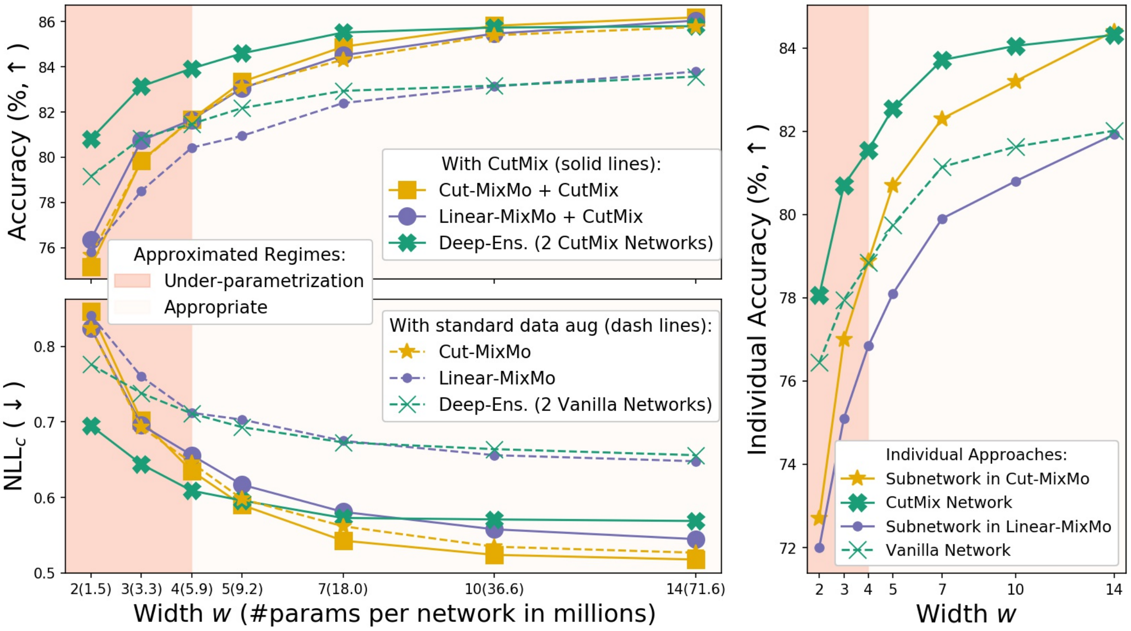
Table D.1 reports averaged scores over 3 runs for our main experiment on CIFAR with WRN-28-10 [Zag+16]. **Bold** highlights best scores, † marks approaches not reimplemented. Cut-MixMo reaches (85.40% Top1, 0.535 NLL$_c$) on CIFAR-100 with $b = 4$: it surpasses our Linear-MixMo (83.08%, 0.656) and MIMO (83.06%, 0.661). Cut-MixMo is further improved when combined with CutMix (85.77%, 0.524). Results remain strong when $b = 2$: Cut-MixMo (84.38%, 0.563) proves better on its own than DE [Lak+17], and MSDAs like Mixup [Zha+18a; Ver+19] or CutMix [Yun+19]. We see similar trends on CIFAR-10: Cut-MixMo reaches 0.081 in NLL$_c$, 0.079 with CutMix. Yet, the costlier batch augmented Mixup BA [Hof+20] edges it out in Top1.

Table D.1. – Main MixMo results: WRN-28-10 on CIFAR.

Dataset		CIFAR-100			CIFAR-10	
Approach	Time Tr./Inf.	Top1 %,↑	Top5 %,↑	NLL$_c$ 10^{-2},↓	Top1 %,↑	NLL$_c$ 10^{-2},↓
Vanilla	1/1	81.63	95.49	73.9	96.34	12.6
Mixup		83.44	95.92	65.7	97.07	11.2
Manifold Mixup†		81.96	95.51	73.4	97.45	12.2
CutMix		84.05	96.09	64.8	97.23	9.9
ResizeMix†		84.31	-	-	97.60	-
Puzzle-Mix†	2/1	84.31	96.46	66.8	-	-
GradAug†	3/1	84.14	96.43	-	-	-
+ CutMix†		85.51	96.86	-	-	-
Mixup BA†	7/1	84.30	-	-	**97.80**	-
DE (2 Nets)	2/2	83.17	96.37	66.4	96.67	11.1
+ CutMix		85.74	96.82	57.1	97.52	8.6
MIMO	2/1	82.40	95.78	68.8	96.38	12.1
Linear-MixMo		82.54	95.99	67.6	96.56	11.4
+ CutMix		84.69	97.12	57.2	97.32	9.4
Cut-MixMo		84.38	96.94	56.3	97.31	8.9
+ CutMix		85.18	97.20	54.5	97.45	8.4
MIMO	4/1	83.06	96.23	66.1	96.74	11.4
Linear-MixMo		83.08	96.26	65.6	96.91	10.8
+ CutMix		85.47	97.04	55.8	97.68	8.7
Cut-MixMo		85.40	97.22	53.5	97.51	8.1
+ CutMix		**85.77**	**97.42**	**52.4**	97.73	**7.9**

D.4.3 MixMo efficiency

Figure D.4 shows how MixMo grows stronger than DE (green curves) as width w in WRN-28-w increases. The parameterization becomes appropriate at $w = 4$: Cut-MixMo (yellow curves) then matches DE (with half the parameters) in Figure D.4(a) and its subnetworks match a vanilla network in Figure D.4(b). Beyond, MixMo better uses over-parameterization: Cut-MixMo + CutMix surpasses DE + CutMix in NLL$_c$ for $w{\geq}5$, and this is true in Top1 for $w{\geq}10$. Compared to our strong Linear-MixMo + CutMix (purple curves), Cut-MixMo performs similarly in Top1, and better with CutMix for $w{\geq}4$. While Linear-MixMo and DE learn from occlusion, Cut-MixMo also benefits from CutMix, notably from the induced label smoothing. Overall, Cut-MixMo, even without CutMix, significantly better estimates uncertainty.



(a) Ensemble Top1 and NLL$_c$.

(b) Individual Top1.

Figure D.4. – Parameters efficiency (metrics/#params). CIFAR-100 with WRN-28-w, b=4. Comparisons between (a) ensemble and some of their (b) individual counterparts.

D.4.4 MixMo analysis on CIFAR-100 w/ WRN-28-10

D.4.4.1 The mixing block $\mathcal{M}$

We study the impact of patch mixing through the lens of the *ensemble diversity/individual accuracy trade-off*. We measure diversity via the pairwise ratio-error [Aks03] (d_{re}, ↑), averaged over the last 10 epochs. As argued in Appendix D.3.2, patch mixing increases diversity compared to linear mixing in Figure D.5. As the probability p of patch mixing grows, so does diversity: from $d_{re}(p = 0.0) \approx 0.78$ (Linear-MixMo) to $d_{re}(p = 0.5) \approx 0.85$ (Cut-MixMo). In contrast, DE has $d_{re} \approx 0.76$ while MIMO has $d_{re} \approx 0.77$ on the same setup. Increasing p past 0.6 boosts diversity even more at the cost of subnetworks' accuracies: this is due to underfitting and an increased test-train distribution gap. $p \in [0.5, 0.6]$ is thus the best trade off.

D.4.4.2 Generalization to $M \geq 2$ subnetworks

We try to generalize MixMo to more than $M = 2$ subnetworks in Figure D.6. Cut-MixMo's subnetworks perform at 82.3% when $M = 2$ vs. 79.5% when $M = 3$. In MIMO, it's 79.8% vs. 77.7%. Because subnetworks do not share features, higher M degrades their results: only two can fit seamlessly. Ensemble Top1 overall decreases in spite of the additional predictions, as already noticed in MIMO [Hav+21]. This reflects MixMo's strength in over-parametrized regimes, but also its limitations with fewer parameters

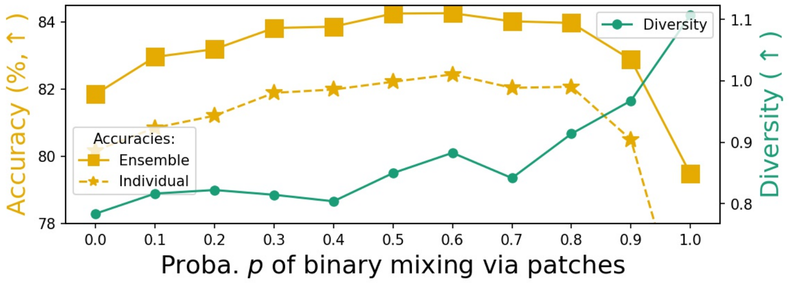


Figure D.5. – Diversity/accuracy as function of p with r=3.

when subnetworks underfit (recall previous Figure D.4). Facing similar findings, MIMO [Hav+21] introduced input repetition so that subnetworks share their features, at the cost of drastically reducing diversity.

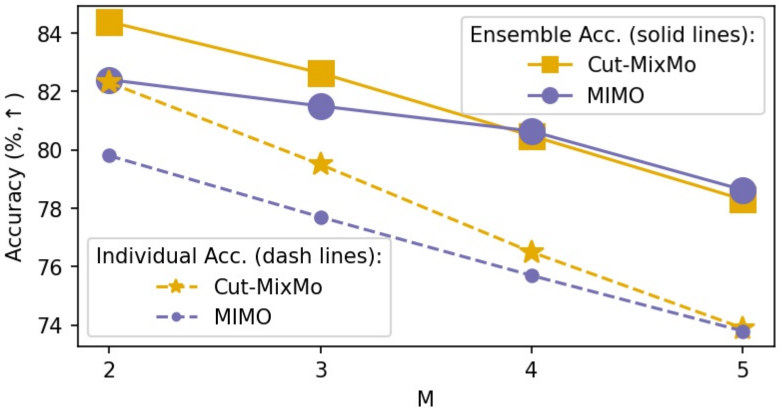


Figure D.6. – Ensemble/individual accuracies for $M \geq 2$.

D.4.4.3 Robustness to image corruptions

To measure MixMo effectiveness under distribution shifts, we now consider CIFAR-100-c [Hen+19a], a version CIFAR-100 where test images have been corrupted. We report WRN-28-10 results with and without AugMix [Hen+19b], a pixels data augmentation technique specialized on this task. Table D.2 shows that Cut-MixMo ($b = 4$) best complements AugMix and reaches 71.1% Top1.

Table D.2. – Robustness comparison on CIFAR-100-c.

Approach	1 Net.		CutMix	Puzzle-Mix†		DE (2 Nets)		MIMO	Linear-MixMo		Cut-MixMo	
AugMix	-	✓	-	-	✓	-	✓	-	-	✓	-	✓
Top1 ↑	52.2	67.8	51.93	58.09	70.46	53.8	69.9	53.6	55.6	70.4	57.0	**71.1**
Top5 ↑	73.7	87.5	72.03	77.3	87.7	74.9	88.9	74.9	76.1	89.4	77.4	**89.5**
NLL ↓	2.50	1.38	2.13	1.96	1.34	2.27	1.24	2.66	2.33	1.22	2.04	**1.16**

D.4.5 Ensemble of MixMo

When combined with CutMix [Yun+19], Cut-MixMo previously set a new state of the art of 85.77% with $N = 1$ WRN-28-10. Since MixMo adds very little parameters ($\approx +1\%$), we can combine independently trained MixMo like in traditional functional ensembling. This ensembling of ensemble of subnetworks leads in practice to the averaging of $M \times N = 2 \times N$ predictions. Final Table D.3 shows it further reaches 86.63% with $N = 2$ and even 86.81% with $N = 3$.

Table D.3. – Best results for WRN-28-10 on CIFAR-100 via Cut-MixMo + CutMix [Yun+19] + N-ensembling and $b = 4$. Previous Top1 sotas: 85.23 [Qin+20], 85.51 [Yan+20c], 85.74 [Zha+20b].

N	# params	Average			Best run		
		Top1 ↑	Top5 ↑	NLL_c ↓	Top1 ↑	Top5 ↑	NLL_c ↓
1	36.6M	85.77 ± 0.14	97.36 ± 0.02	0.524 ± 0.005	85.92	97.36	0.518
2	73.2M	86.63 ± 0.19	97.73 ± 0.05	0.479 ± 0.003	86.75	97.80	0.475
3	109.8M	86.81 ± 0.17	97.85 ± 0.04	0.464 ± 0.002	86.94	97.83	0.464

D.5 Conclusion

We introduce the MixMo framework that generalizes the multi-input multi-output paradigm. MixMo can be analyzed as either an ensembling method or a mixed samples data augmentation, while remaining complementary to works from both lines of research. MixMo improved the state of the art on CIFAR-100, CIFAR-100-c and Tiny ImageNet.

Limitations. Despite some relative success on medium-scale datasets, the MixMo framework has certain constraints. First, it is wasteful in its use of parameterization. Indeed, as we latter highlighted in MixShare [Sun+22], the learned subnetworks fail to share even generic features. Specifically, each channel or feature is almost exclusively used by one subnetwork; thus each additional subnetwork significantly reduces the effective size of the other subnetworks. This explains why MixMo requires (very) wide base models for larger datasets, and MixMo's failure with $M > 2$ subnetworks in Appendix D.4.4.2. Though we tried to mitigate this failure in [Sun+22], results remained inconclusive. Overall, this limits MixMo's applicability. Moreover, MixMo requires training from scratch; and thus cannot benefit from transfer learning from foundation models. This limitation renders MixMo less relevant at the end of this thesis, thus was relegated in Appendix for the sake of brevity.

Appendix E

FISHR: INVARIANT GRADIENT VARIANCES FOR GENERALIZATION UNDER CORRELATION SHIFTS

E.1 Introduction

The success of DNNs in supervised learning [Kri+12] relies on the crucial assumption that the train and test data distributions are identical. In particular, the tendency of networks to rely on simple features [Val+19; Gei+20] is generally a desirable behavior reflecting Occam's razor. However, in case of spurious features, this simplicity bias deteriorates performance when more complex features are needed [Ten18; Sha+20; Ker+21]. For example, in the recent fight against Covid-19, most of the DL methods developed to detect coronavirus from chest scans were shown useless for clinical use [DeG+21; Rob+21]: models actually exploited simple bias in the training datasets such as patients' age or body position rather than *truly* analyzing medical pathologies.

To better generalize under correlation shift, ensembling strategies are useless, as verified empirically in Section 4.6.3.4: theoretically, they do not reduce the bias, as they all learn the same predictive mechanism based on the same spurious features. Indeed, when the train-test shifts occur in the posterior covariate distributions, additional information is required to differentiate between the relevant and the spurious features [Bla+11; Mua+13]. The standard *invariance* strategy [Arj+19] is to learn simultaneously from multiple training domains, across which we assume an underlying invariant causal mechanism [Pet+16]. To remove the domain-dependent explanations, different *invariance criteria across those training domains* have been proposed. [Gan+16; Sun+16] enforce similar feature distributions, others [Arj+19; Kru+21] force the classifier to be simultaneously optimal across all domains. Yet, despite the popularity of this research topic, none of these methods perform significantly better than the classical empirical risk minimization (ERM) when applied with controlled model selection and restricted hyperparameter search [Gul+21; Ye+22]. These failures motivate the need for new ideas.

To foster the emergence of a shared mechanism with consistent generalization properties, our intuition is that learning should progress consistently and similarly across domains. Besides, the learning procedure of DNNs is dictated by the distribution of the gradients with respect to the network weights [Yin+18; San+20], usually backpropagated in the network during gradient descent. Additionally, individual gradients are expressive representations of the input [For+19b; Cha+19]. Thus, we seek distributional invariance across domains in the gradient space: *domain-level gradients should be similar*, not only in average direction, but most importantly in statistics such as variance and disagreements.

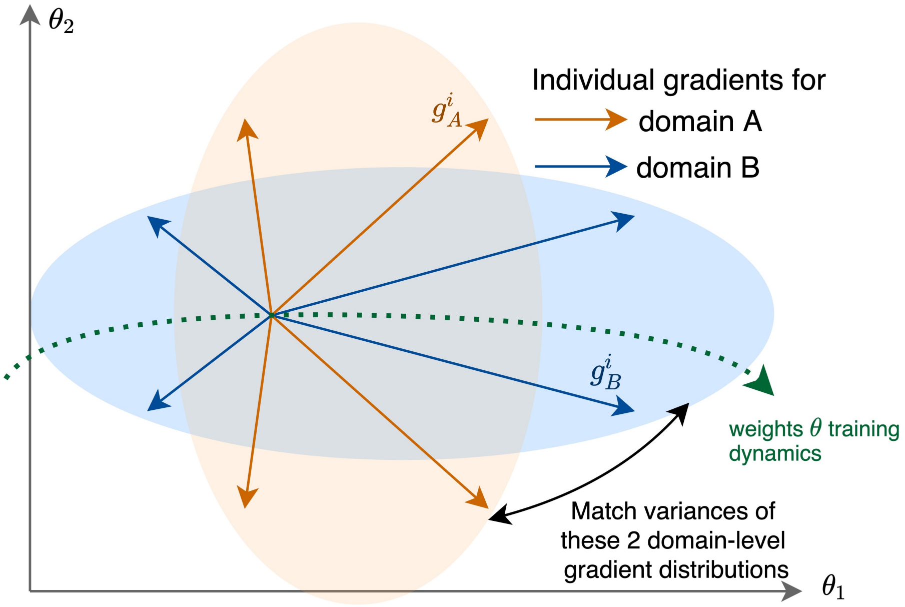

θ_2
Individual gradients for
domain A
domain B
g_A^i
g_B^i
weights θ training dynamics
Match variances of these 2 domain-level gradient distributions
θ_1

Figure E.1. – Fishr considers the individual (per-sample) gradients of the loss in the network weights θ. Specifically, Fishr matches the domain-level gradient variances of the distributions across the two training domains: A ($\{\boldsymbol{g}_A^i\}_{i=1}^{n_A}$ in orange) and B ($\{\boldsymbol{g}_B^i\}_{i=1}^{n_B}$ in blue). We will show how this regularization during the learning of θ improves the out-of-distribution generalization properties by aligning the domain-level loss landscapes at convergence.

In this chapter, we propose the Fishr regularization for generalization under correlation shift. As summarized in Figure E.1, Fishr enforces domain invariance in the space of the gradients of the loss; we *match the domain-level variances of gradients*, *i.e.*, the second moment of the gradient distributions. In contrast, previous gradient-based works such as Fish [Shi+21] only match the domain-level gradients means, *i.e.*, the first moment.

Our strategy is also motivated by the close relations between the gradient variance, the Fisher Information [Fis22] and the Hessian of the loss. This explains the name of our work, Fishr, using gradients as in Fish and related to the Fisher Matrix. Notably, we will study how *Fishr forces the model to have similar domain-level Hessians*. More broadly, Fishr aligns the domain-level loss landscapes locally around the final weights and promotes consistent explanations [Par+21].

To reduce the computational cost, we justify an approximation that tackles the gradients only in the classifier, easily implemented with BackPACK [Dan+20].

- We introduce Fishr, a scalable regularization that brings closer the domain-level gradient variances (Appendix E.3.1).
- We theoretically justify that Fishr matches domain-level risks and Hessians, and consequently, reduces inconsistencies across domains (Appendix E.3.2).
- Empirically, our experiments validate that Fishr can tackle correlation shifts on ColoredMNIST [Arj+19] (Appendix E.4).

This chapter has led to the publication of the following paper: Alexandre Ramé, Corentin Dancette, and Matthieu Cord. "Fishr: Invariant Gradient Variances for Out-of-Distribution Generalization". In: *ICML*. 2022.

E.2 Related work

Our model is a deep neural network (DNN) $f(\cdot, \theta)$ (parametrized by θ) made of a deep features extractor Φ (parametrized by ϕ) on which we plug a dense linear classifier w (parametrized by ω). In training, we have access to different domains $\mathcal{E}$: for each domain $e \in \mathcal{E}$, the dataset $\mathcal{D}_e = \left\{\left(\boldsymbol{x}_e^i, \boldsymbol{y}_e^i\right)\right\}_{i=1}^{n_e}$ contains n_e *i.i.d.* (input, labels) samples drawn from a domain-dependent probability distribution. Combined together, the datasets $\{\mathcal{D}_e\}_{e \in \mathcal{E}}$ are of size $n_S = \sum_{e \in \mathcal{E}} n_e$. Our goal is to learn weights θ so that f predicts well on a new test domain, unseen in training: θ should ideally capture an invariant mechanism across training domains. Following standard notations, $\|\boldsymbol{M}\|_F^2$ denotes the Frobenius norm of matrix $\boldsymbol{M}$; $\|\boldsymbol{v}\|_2^2$ denotes the euclidean norm of vector $\boldsymbol{v}$; $\mathbf{1}$ is a column vector with all elements equal to 1.

The standard ERM [Vap99] framework simply minimizes the average empirical risk over all training domains, *i.e.*, $\frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \mathcal{R}_e(\theta)$. To tackle the correlation shifts, many approaches try to exploit the domain information. Some works explore data augmentations to mix samples from different domains [Wan+20b; Wu+20b], some re-weight the training samples to favor underrepresented groups [Sag+20a; Sag+20b; Zha+21] and others include domain-dependent weights [Din+17a; Man+18]. Yet, most recent works promote invariance via a regularization criterion and only differ by the choice of the statistics to be matched across training domains. They can be categorized into three groups: these methods enforce agreement either (i) in features (ii) in predictors or (iii) in gradients.

First, some approaches aim at extracting **domain-invariant features** and were extensively studied for unsupervised domain adaptation. The features are usually aligned with adversarial methods [Gan+16; Gon+16; Li+18a; Li+18b] or with kernel methods [Mua+13; Lon+14]. Yet, the simple covariance matching in CORAL [Sun+16] performs best on various tasks for OOD generalization [Gul+21]. With $\boldsymbol{Z}_e^{ij}$ the j-th dimension of the features extracted by Φ_ϕ for the i-th example $\boldsymbol{x}_e^i$ of domain $e \in \mathcal{E} = \{A, B\}$, CORAL minimizes $\|\mathrm{Cov}(\boldsymbol{Z}_A) - \mathrm{Cov}(\boldsymbol{Z}_B)\|_F^2$ where $\mathrm{Cov}(\boldsymbol{Z}_e) = \frac{1}{n_e - 1}(\boldsymbol{Z}_e^\top \boldsymbol{Z}_e - \frac{1}{n_e}\left(\mathbf{1}^\top \boldsymbol{Z}_e\right)^\top \left(\mathbf{1}^\top \boldsymbol{Z}_e\right))$ is the feature covariance matrix. CORAL is more powerful than mere feature matching $\left\|\frac{1}{n_A}\mathbf{1}^\top \boldsymbol{Z}_A - \frac{1}{n_B}\mathbf{1}^\top \boldsymbol{Z}_B\right\|_2^2$ as in [Tze+14]. Yet, [Joh+19] and [Zha+19c] show that these approaches are insufficient to guarantee good generalization.

Motivated by arguments from causality [Pea09] and the idea that statistical dependencies are epiphenomena of an underlying structure, Invariant Risk Minimization (IRM) [Arj+19] explains that the **predictors should be invariant** [Pet+16; Roj+18], *i.e.*, simultaneously optimal across all domains. Yet, recent works point out pitfalls of IRM [Guo+21; Kam+21; Ahu+21], that does not provably work with non-linear data [Ros+21] and could not improve over ERM when hyperparameter selection is restricted [Koh+21; Gul+21]. Among many suggested improvements [Cha+20; Idn+20; Ten+21; Ahm+21], Risk Extrapolation (V-REx) [Kru+21] argues that training risks from different domains should be similar and thus penalizes $|\mathcal{R}_A - \mathcal{R}_B|^2$ when $\mathcal{E} = \{A, B\}$.

A third and most recent line of work promotes **agreements between gradients** with respect to network weights. Gradient agreements have been previously employed for multitasks [Du+18; Yu+20], continual [Lop+17], meta [Fin+17; Zha+20a] and reinforcement [Zha+19b] learning. In OOD generalization, [Koy+20; Par+21; Shi+21] try to find minimas in the loss landscape that are shared across domains; they tackle the domain-level expected gradients:

$$\boldsymbol{g}_e = \mathbb{E}_{(\boldsymbol{x}_e, \boldsymbol{y}_e) \sim \mathcal{D}_e} \nabla_\theta \ell \left(f(\boldsymbol{x}_e, \theta), \boldsymbol{y}_e \right). \tag{E.1}$$

When $\mathcal{E} = \{A, B\}$, IGA [Koy+20] minimizes $||\boldsymbol{g}_A - \boldsymbol{g}_B||_2^2$; Fish [Shi+21] increases $\boldsymbol{g}_A \cdot \boldsymbol{g}_B$; AND-mask [Par+21] and others [Man+21b; Sha+21] update weights only when $\boldsymbol{g}_A$ and $\boldsymbol{g}_B$ point to the same direction. Along with the increased computation cost, the main limitation of previous gradient-based methods is the per-domain batch averaging of gradients: this removes more granular statistics, in particular the information from pairwise interactions between gradients from samples in a same domain. In opposition, our new regularization keeps extra information from individual gradients and matches across domains the domain-level gradient variances. In brief, Fishr is similar to the covariance-based CORAL [Sun+16] but in the gradient space rather than in the feature space.

E.3 Fishr

E.3.1 Gradient variance matching

The **individual gradient** $\boldsymbol{g}_e^i = \nabla_\theta \ell \left(f(\boldsymbol{x}_e^i, \theta), \boldsymbol{y}_e^i \right)$ is the first-order derivative for the i-th data example $\left(\boldsymbol{x}_e^i, \boldsymbol{y}_e^i\right)$ from domain $e \in \mathcal{E}$ with respect to the weights θ. Previous methods have matched the gradient means $\boldsymbol{g}_e = \frac{1}{n_e} \sum_{i=1}^{n_e} \boldsymbol{g}_e^i$ for each domain $e \in \mathcal{E}$. These gradient means capture the average learning direction but can not capture gradient disagreements [San+20; Yin+18]. With $\boldsymbol{G}_e = [\boldsymbol{g}_e^i]_{i=1}^{n_e}$ of size $n_e \times |\theta|$, we compute the **domain-level gradient variance** vectors of size $|\theta|$:

$$\boldsymbol{v}_e = \mathrm{Var}(\boldsymbol{G}_e) = \frac{1}{n_e - 1} \sum_{i=1}^{n_e} \left(\boldsymbol{g}_e^i - \boldsymbol{g}_e \right)^2, \tag{E.2}$$

where the square indicates an element-wise product. To reduce the distribution shifts in $f(\cdot, \theta)$ across domains, we bring the domain-level gradient variances $\{\boldsymbol{v}_e\}_{e \in \mathcal{E}}$ closer. Hence, our Fishr regularization is:

$$\mathcal{L}_{\text{Fishr}}(\theta) = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \|\boldsymbol{v}_e - \boldsymbol{v}\|_2^2, \tag{E.3}$$

the square of the Euclidean distance between the gradient variance from the different domains $e \in \mathcal{E}$ and the mean gradient variance $\boldsymbol{v} = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \boldsymbol{v}_e$. Balanced with a hyperparameter coefficient $\lambda > 0$, this Fishr penalty complements the original ERM objective, *i.e.*, the empirical training risks:

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \mathcal{R}_e(\theta) + \lambda \mathcal{L}_{\text{Fishr}}(\theta). \tag{E.4}$$

Remark E.1. *Gradients $\boldsymbol{g}_e^i$ can be computed on all network weights θ. Yet, to reduce the memory and training costs, they will often be computed only on a subset of θ, e.g., only on the linear classifier's weights ω.*

E.3.2 Theoretical analysis

We theoretically motivate our Fishr regularization by leveraging the *domain inconsistency score* introduced in AND-mask [Par+21]. We first derive a generalized upper bound for this score. Then, we show that Fishr minimizes this upper bound by matching simultaneously *domain-level risks and Hessians*.

E.3.2.1 Inconsistency formalism

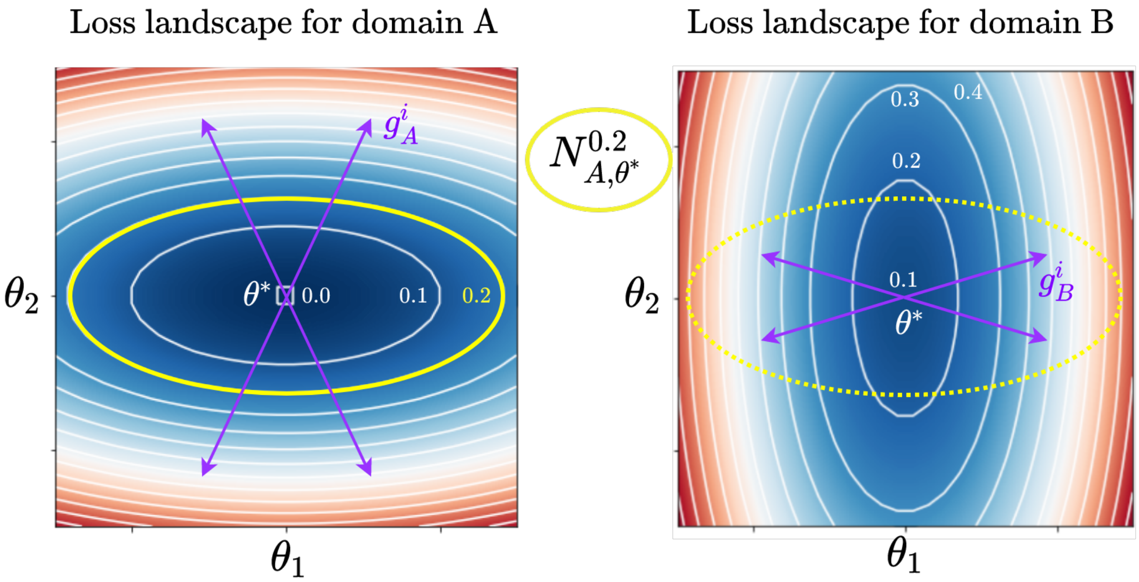


Figure E.2. – **Loss landscapes around inconsistent weights** θ^* at convergence. $N^{0.2}_{A,\theta^*}$ contains weights θ for which $\mathcal{R}_A(\theta)$ is low (≤ 0.2) but $\mathcal{R}_B(\theta)$ is high (≥ 0.9). This inconsistency is due to conflicting domain-level loss landscapes, specifically gaps between domain-level risks and curvatures at θ^*. This is visible in the disagreements across the variances of gradients $\{\boldsymbol{g}_A^i\}_{i=1}^{n_A}$ and $\{\boldsymbol{g}_B^i\}_{i=1}^{n_B}$.

[Par+21] argues that "patchwork solutions sewing together different strategies" for different domains may not generalize well: good weights should be optimal on all domains and "hard to vary" [Deu11]. They formalize this insight with an inconsistency score:

$$\mathcal{I}^{\epsilon}\left(\theta^{*}\right)=\max_{(A,B)\in\mathcal{E}^{2}}\max_{\theta\in N_{A,\theta^{*}}^{\epsilon}}\left|\mathcal{R}_{B}(\theta)-\mathcal{R}_{A}(\theta^{*})\right|, \tag{E.5}$$

where $\theta\in N_{A,\theta^{*}}^{\epsilon}$ if there exists a path in the weights space between θ and θ^{*} where the risk $\mathcal{R}_{A}$ remains in an $\epsilon>0$ interval around $\mathcal{R}_{A}(\theta^{*})$. $\mathcal{I}$ increases with conflicting geometries in the loss landscapes around θ^{*} as in Figure E.2: *i.e.*, when another "close" solution θ is equivalent to the current solution θ^{*} in a domain A but yields different risks in B. For $e\in\mathcal{E}$, the second-order Taylor expansion of $\mathcal{R}_{e}$ around $\theta^{*}=0$ (with a change of variable) gives:

$$\mathcal{R}_{e}(\theta)=\mathcal{R}_{e}(\theta^{*})+\theta^{\top}\nabla_{\theta}\mathcal{R}_{e}\left(\theta^{*}\right)+\frac{1}{2}\theta^{\top}H_{e}\theta+\mathcal{O}(\|\theta\|_{2}^{2}),$$

where the Hessian $\boldsymbol{H}_{e}=\nabla_{\theta}^{2}\mathcal{R}_{e}(\theta^{*})$ approximates the local curvature of the loss landscape. Moreover, we assume simultaneous convergence, *i.e.*, θ^{*} is a local minima across all domains: $\nabla_{\theta}\mathcal{R}_{e}(\theta^{*})=0$. Thus, locally around θ^{*}:

$$\begin{aligned}\max_{\theta\in N_{A,\theta^{*}}^{\epsilon}}\left|\mathcal{R}_{B}(\theta)-\mathcal{R}_{A}(\theta^{*})\right|&\approx\max_{\left|\mathcal{R}_{A}(\theta)-\mathcal{R}_{A}(\theta^{*})\right|\leq\epsilon}\left|\mathcal{R}_{B}(\theta)-\mathcal{R}_{A}(\theta^{*})\right|\\&\approx\max_{\frac{1}{2}\left|\theta^{\top}H_{A}\theta\right|\leq\epsilon}\left|\mathcal{R}_{B}(\theta^{*})+\frac{1}{2}\theta^{\top}H_{B}\theta-\mathcal{R}_{A}(\theta^{*})\right|\\&\lessapprox\left|\mathcal{R}_{B}(\theta^{*})-\mathcal{R}_{A}(\theta^{*})\right|+\max_{\frac{1}{2}\left|\theta^{\top}H_{A}\theta\right|\leq\epsilon}\frac{1}{2}\left|\theta^{\top}H_{B}\theta\right|,\end{aligned} \tag{E.6}$$

where we deduced the last line from the triangle inequality. In [Ram+22a] we formally demonstrate the following equality when the per-domain risks are assumed quadratic.

Proposition E.1. *Under the quadratic bowl assumption (i.e., when per-domain risks are assumed quadratic with positive definite Hessians) and for sufficiently small* ϵ*, then:*

$$\mathcal{I}^{\epsilon}\left(\theta^{*}\right)=\max_{(A,B)\in\mathcal{E}^{2}}\Big(\mathcal{R}_{B}(\theta^{*})-\mathcal{R}_{A}(\theta^{*})+\max_{\frac{1}{2}\theta^{\top}H_{A}\theta\leq\epsilon}\frac{1}{2}\theta^{\top}H_{B}\theta\Big). \tag{E.7}$$

The Hessian being positive definite is a standard hypothesis, notably used in [Par+21], that is empirically reasonable [Sag+18]: "in only very few steps [...] large negative eigenvalues disappear" [Gho+19]. We now analyze the two terms from this bound.

The first term in the RHS of Proposition E.1 is the difference between domain-level risks, whose square is the criterion minimized in V-REx [Kru+21]. We will show that Fishr forces this term to be small in Appendix E.3.2.2.

For **the second term**, we follow the diagonal approximation of the Hessians from [Par+21]. In that case, $\boldsymbol{H}_e = \operatorname{diag}\left(\lambda_1^e, \cdots, \lambda_h^e\right)$ with $\forall i \in \{1, \ldots, h\}, \lambda_i^e > 0$. Then:

$$\max_{\frac{1}{2}\theta^\top H_A \theta \leq \epsilon} \frac{1}{2}\theta^\top H_B \theta = \max_{\|\tilde{\theta}\|_2^2 \leq \epsilon} \sum_i \tilde{\theta}_i^2 \lambda_i^B / \lambda_i^A = \epsilon \times \max_i \lambda_i^B / \lambda_i^A. \tag{E.8}$$

This is large when exists i such that λ_i^A is small but λ_i^B is large: indeed, a small weight perturbation in the direction of the associated eigenvector would change the loss slightly in the domain A but drastically in domain B. Thus, this second term decreases when $\boldsymbol{H}_A$ and $\boldsymbol{H}_B$ have similar eigenvalues. This result holds when Hessians are co-diagonalizable. In conclusion, this explains why forcing $\boldsymbol{H}_A = \boldsymbol{H}_B$ reduces inconsistencies in the loss landscape and thus improves generalization. AND-mask matches Hessians by zeroing out gradients with inconsistent directions across domains; however, this masking strategy introduces dead zones [Sha+21] in weights where the model could get stuck, ignores gradient magnitudes and empirically performs poorly with real datasets from DomainBed. As shown in Appendix E.3.2.3, Fishr proposes a new method to align domain-level Hessians leveraging the close relations between the gradient variance, the Fisher Information and the Hessian.

E.3.2.2 Fishr matches the domain-level risks

Gradients take into account the label Y, which appears as an argument for the loss ℓ. Hence, gradient-based approaches are 'label-aware' by design. In contrast, feature-based methods were shown to fail in case of label shifts, because they do not consider Y [Joh+19; Zha+19c]. The fact that the label and the loss appear in the formula of the gradients has another important consequence: matching gradient distributions also matches training risks, as motivated in V-REx [Kru+21]. We confirm this insight in Table E.2: matching gradient variances with Fishr induces $|\mathcal{R}_A - \mathcal{R}_B|^2 \to 0$ when $\mathcal{E} = \{A, B\}$. *Intuitively*, gradient amplitudes are directly weighted by the loss values: multiplying the loss by a constant will also multiply the gradients by the same constant. Thus roughly, if the domain-level empirical training risks are different, then the domain-level gradient norms should also differ. *Theoretically*, we prove in the paper that Fishr regularization component with reference to the classification bias is exactly the difference between domain-level mean squared errors. We recover the objective from V-REx [Kru+21], with a different loss (squared error instead of negative log likelihood).

E.3.2.3 Fishr matches the domain-level Hessians

The Hessian matrix $\boldsymbol{H} = \sum_{i=1}^n \nabla_\theta^2 \ell\left(f_\theta(\boldsymbol{x}^i), \boldsymbol{y}^i\right)$ is of key importance in deep learning. Yet, $\boldsymbol{H}$ cannot be computed efficiently in general. Recent methods [Izm+18; Par+21; For+21] tackled the Hessian indirectly by modifying the learning procedure. In contrast, we use the fact that the diagonal of $\boldsymbol{H}$ is approximated by the gradient variance $\operatorname{Var}(\boldsymbol{G})$; this is confirmed in Table E.1. This result is derived below from 3 individual and standard approximation steps.

Table E.1. – **Cosine similarity between Hessian diagonals and gradient variances,** $\cos\left(\mathrm{Diag}\left(\boldsymbol{H}_e\right), \mathrm{Var}(\boldsymbol{G}_e)\right)$, for an ERM at convergence on ColoredMNIST with the two training domains $e \in \{90\%, 80\%\}$.

	$e = 90\%$	$e = 80\%$
On classifier weights ω	0.9999980	0.9999905
On all network weights θ	0.9971040	0.9962264

The Hessian and the "true" Fisher information matrix (FIM). The "true" FIM $\boldsymbol{F} = \sum_{i=1}^{n} \mathbb{E}_{\hat{\boldsymbol{y}}\sim P_\theta(\cdot|\boldsymbol{x}^i)} \left[\nabla_\theta \log p_\theta(\hat{\boldsymbol{y}}|\boldsymbol{x}^i)\nabla_\theta \log p_\theta(\hat{\boldsymbol{y}}|\boldsymbol{x}^i)^\top\right]$ [Fis22; CR45] approximates the Hessian $\boldsymbol{H}$ with theoretically probably bounded errors under mild assumptions [Scho2].

The "true" FIM and the "empirical" FIM. Yet, $\boldsymbol{F}$ remains costly as it demands one backpropagation per class. That's why most empirical works (*e.g.*, in compression [Fra+21; Liu+21] and optimization [Dan+21]) approximate the "true" FIM $\boldsymbol{F}$ with the "empirical" FIM $\tilde{\boldsymbol{F}} = \boldsymbol{G}_e^\top \boldsymbol{G}_e = \sum_{i=1}^{n} \nabla_\theta \log p_\theta(\boldsymbol{y}^i|\boldsymbol{x}^i)\nabla_\theta \log p_\theta(\boldsymbol{y}^i|\boldsymbol{x}^i)^\top$ [Mar14] where $p_\theta(\cdot|\boldsymbol{x})$ is the density predicted by f_θ on input $\boldsymbol{x}$. While $\boldsymbol{F}$ uses the model distribution $P_\theta(\cdot|X)$, $\tilde{\boldsymbol{F}}$ uses the data distribution $P(Y|X)$. Despite this key difference, $\tilde{\boldsymbol{F}}$ and $\boldsymbol{F}$ were shown to share the same structure and to be similar up to a scalar factor [Tho+20]. They also have analogous properties: $\mathrm{Tr}(\tilde{\boldsymbol{F}}) \approx \mathrm{Tr}(\boldsymbol{F})$. This was discussed in [Li+20c] and further highlighted even at early stages of training (before overfitting) in the Fig. 1 and the Appendix S3 of [Sin+20a].

The "empirical" FIM and the gradient covariance. Critically, $\tilde{\boldsymbol{F}}$ is nothing else than the unnormalized uncentered covariance matrix when ℓ is the negative log-likelihood. Thus, the gradient covariance matrix $\boldsymbol{C} = \frac{1}{n-1}\left(\boldsymbol{G}^\top \boldsymbol{G} - \frac{1}{n}\left(\mathbf{1}^\top \boldsymbol{G}\right)^\top \left(\mathbf{1}^\top \boldsymbol{G}\right)\right)$ of size $|\theta| \times |\theta|$ and $\tilde{\boldsymbol{F}}$ are equivalent (up to the multiplicative constant n) at any first-order stationary point: $\boldsymbol{C} \underset{\sim}{\propto} \tilde{\boldsymbol{F}}$. Overall, this suggests that $\boldsymbol{C}$ and $\boldsymbol{H}$ are closely related [Jas+18];.

Table E.2. – **Invariance analysis** at convergence on ColoredMNIST across the two training domains $\mathcal{E} = \{90\%, 80\%\}$. Compared to ERM, Fishr matches the gradient variance ($\mathrm{Diag}(\boldsymbol{C}_{90\%}) \approx \mathrm{Diag}(\boldsymbol{C}_{80\%})$) in all network weights θ. Most importantly, this enforces invariance in domain-level risks ($\mathcal{R}_{90\%} \approx \mathcal{R}_{80\%}$) and in domain-level Hessians ($\mathrm{Diag}(\boldsymbol{H}_{90\%}) \approx \mathrm{Diag}(\boldsymbol{H}_{80\%})$). The gradient variance, computable efficiently with a unique backpropagation, serves as a proxy for the Hessian. Details and more experiments in Figure E.3.

	ERM	Fishr
$\lVert \mathrm{Var}(\boldsymbol{G}_{90\%}) - \mathrm{Var}(\boldsymbol{G}_{80\%}) \rVert_F^2$	1.6	4.1×10^{-5}
$\lvert \mathcal{R}_{90\%} - \mathcal{R}_{80\%} \rvert^2$	1.0×10^{-2}	3.8×10^{-6}
$\lVert \mathrm{Diag}\left(\boldsymbol{H}_{90\%} - \boldsymbol{H}_{80\%}\right) \rVert_F^2$	2.9×10^{-1}	2.7×10^{-4}

Consequences for Fishr. Critically, Fishr considers the gradient variance $\mathrm{Var}(\boldsymbol{G})$, *i.e.*, the diagonal components of $\boldsymbol{C}$. In our multi-domain framework, we define the domain-level matrices with the subscript e. Table E.2 empirically confirms that matching $\{\text{Diag}(\boldsymbol{C}_e)\}_{e\in\mathcal{E}}$ (*i.e.*, $\{\text{Var}(\boldsymbol{G}_e)\}_{e\in\mathcal{E}}$) with Fishr forces the domain-level Hessians $\{\text{Diag}(\boldsymbol{H}_e)\}_{e\in\mathcal{E}}$ to be aligned at convergence (on the diagonal for computational reasons). Tackling the second moment of the first-order derivatives enables to regularize the second-order derivatives. Moreover, in [Ram+22a] we confirm that matching the diagonals of $\{\boldsymbol{C}_e\}_{e\in\mathcal{E}}$ or $\{\tilde{\boldsymbol{F}}_e\}_{e\in\mathcal{E}}$ (*i.e.*, centering or not the variances) perform similarly.

Remark E.2. ***Limitation of our approximation.*** *We acknowledge that approximating the "true" FIM $\boldsymbol{F}$ by the "empirical" FIM $\tilde{\boldsymbol{F}}$ is not fully justified theoretically [Mar14; Kun+19]. Indeed, this approximation is valid only under strong assumptions, in particular χ^2 convergence of predictions $P_\theta(\cdot|X)$ towards labels $P(Y|X)$, as detailed in Proposition 1 from [Tho+20]. In Fishr, we trade off theoretical guarantees for efficiency.*

Remark E.3. ***Diagonal approximation.*** *The empirical similarities between $\boldsymbol{C}$ and $\boldsymbol{H}$ motivate using* gradient variance rather than gradient covariance, *which scales down the number of targeted components from $|\theta|^2$ to $|\theta|$. Indeed, diagonally approximating the Hessian is common: e.g., for OOD generalization [Par+21], optimization [LeC+12; Kin+15], continual learning [Kir+17] and pruning [LeC+90; The+18]. This is based on the empirical evidence [Bec+88] that Hessians are diagonally dominant at the end of training. Our diagonal approximation is also motivated by the critical importance of $\text{Tr}(\boldsymbol{C})$ [Jas+21; Fag+20] to analyze the generalization properties of DNNs. We confirm empirically in the paper that considering the off-diagonal parts of $\boldsymbol{C}$ performs no better than just matching the diagonals.*

Conclusion. Fishr efficiently matches (i) domain-level empirical risks and (ii) domain-level Hessians across the training domains, using gradient variances as a proxy. This will align domain-level loss landscapes, reduce domain inconsistencies and increase domain generalization. In particular, the domain-level Hessian matching illustrates that Fishr is more than just a generalization of gradient-mean approaches such as Fish [Shi+21].

E.4 Experiments

We validate Fishr effectiveness on ColoredMNIST [Arj+19], where the task is to predict whether the digit is below or above 5. Moreover, the labels are flipped with 25% probability. Critically, the digits' colors spuriously correlate with the labels: the correlation strength varies across the two training domains $\mathcal{E} = \{90\%, 80\%\}$. To test whether the model has learned to ignore the color, this correlation is reversed at test time. In brief, a biased model that only considers the color would have 10% test accuracy whereas an oracle model that perfectly predicts the shape would have 75%. As previously done in V-REx [Kru+21], we *strictly* follow the IRM implementation and just replace the IRM penalty by our Fishr penalty. This means that we use the exact same MLP and hyperparameters, notably the same *two-stage scheduling* for the regularization strength λ, that is low until epoch 190 and then jumps to a large value, which was optimized via a grid-search for IRM.

Table E.3. – ColoredMNIST results. All methods use hyperparameters optimized for IRM.

Method	Train acc.	Test acc.	Gray test acc.
ERM	86.4 ± 0.2	14.0 ± 0.7	71.0 ± 0.7
ENS	**86.8**	14.1	**71.7**
IRM	71.0 ± 0.5	65.6 ± 1.8	66.1 ± 0.2
V-REx	71.7 ± 1.5	67.2 ± 1.5	68.6 ± 2.2
Fishr$_\theta$	69.6 ± 0.9	71.2 ± 1.1	70.2 ± 0.7
Fishr$_\omega$	71.0 ± 0.9	69.5 ± 1.0	70.2 ± 1.1
Fishr$_\phi$	65.6 ± 1.3	**73.8** ± 1.0	70.0 ± 0.9

Table E.3 reports the accuracy averaged over 10 runs with standard deviation. ENS averages the predictions of the 10 ERM runs. Fishr$_\theta$ (*i.e.*, applying Fishr on all weights θ) obtains the best trade-off between train and test accuracies; notably in test, it reaches 71.2%, or 70.2% when digits are grayscale. Moreover, computing the gradients only in the classifier w_ω performs almost as well (69.5% in test for Fishr$_\omega$) while reducing drastically the computational cost. Finally, Fishr$_\phi$ only in the features extractor ϕ works best in test, though it has lower train accuracy. This last experiment shows that we can reduce domain shifts without explicitly forcing the predictors to be simultaneously optimal. These results highlight the effectiveness of gradient variance matching, even with standard hyperparameters, at different layers of the network.

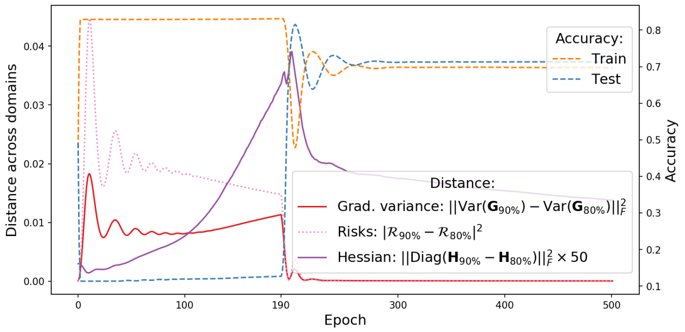


Figure E.3. – ColoredMNIST dynamics. At epoch 190, λ strongly steps up: then, the Fishr$_\theta$ regularization matches the domain-level gradient variances (red) across domains $\mathcal{E} = \{90\%, 80\%\}$, and consequently, the training empirical risks (dotted pink) and Hessians (purple). This reduces train accuracy (orange) but increases test accuracy (blue) as the network learns to predict the digit's shape.

Moreover, the training dynamics in Figure E.3 show that the domain-level empirical risks get closer once the Fishr$_\theta$ gradient variance matching loss is activated after step 190 ($|\mathcal{R}_{90\%} - \mathcal{R}_{80\%}| \to 0$), even though predicting accurately on the domain 90% is easier than on the domain 80%. This confirms insights from Appendix E.3.2.2. Similarly, we observe that Fishr matches Hessians across the two training domains. Overall, Fishr regularization reduces train accuracy, but considerably increases test accuracy.

Experiments on DomainBed. In the paper [Ram+22a], we have also conducted extensive experiments on the DomainBed benchmark [Gul+21]. When Fishr was published, it was the sota method; moreover, according to the latest review paper [Yu+23], Fishr remains the best method based on invariance. However, all the strategies based on ensembling were shown more efficient on these real-world datasets under which diversity shifts dominate. Therefore, for the sake of brevity, we do not report these DomainBed experiments in this thesis, but refer the reader to the published version of the paper.

E.5 Conclusion

In this chapter, we addressed a limitation of ensembling strategies; their inability to tackle correlation shifts. To this end, we leverage the invariance paradigm, and specifically derive a new and simple regularization, Fishr, matching the gradient variances across domains as a proxy for matching domain-level risks and Hessians. We prove that invariance can tackle correlation shift. We hope to pave the way towards strategies that simultaneously tackle diversity and correlation shifts.

LIST OF FIGURES

LIST OF TABLES