Towards Uncertainty-Aware and Privacy Preserving Deep Learning

Francesco Pinto

St. Cross College

University of Oxford

This dissertation is submitted for the degree of

Doctor of Philosophy

Trinity 2024

Si cartuscella volat, tota scientia squagliat.

Popular saying, introduced to me by M. Antonia Notari

Let the Sperm of a man by itself be putrified in a gourd glasse, sealed up, with the highest degree of putrefaction in Horse dung, for the space of forty days, or so long untill it begin to bee alive, move, and stir, which may easily be seen. After this time it will be something like a Man, yet transparent, and without a body. The Homunculus achieves state-of-the-art performance on completely nonsensical benchmarks.

De Natura Rerum, Paracelsus @ Top ML/CV Conference

Life is full of strange absurdities, which, strangely enough, do not even need to appear plausible, since they are true.

Six Characters in Search of an Author, Luigi Pirandello

If light is scarce then light is scarce; we will immerse ourselves in the darkness and there discover its own particular beauty.
Were it not for shadows, there would be no beauty.

In Praise of Shadows, Jun'ichirō Tanizaki

It's actually the souls of the trees we're seeing in the winter.

Nymphomaniac, Lars Von Trier

To Athos, Paul and all the Beauty: Pleasure and Pain

The World is Quiet Here

Declaration

I hereby declare that this thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work based on publications in collaboration with the co-authors where due acknowledgement is made.

Francesco Pinto
St. Cross College

Acknowledgements

Those who deserve my gratitude already know. Those who do not too.

"So long and thanks for all the fish! The world's about to be destroyed, there's no point getting all annoyed: lie back and let the planet dissolve. Despite those nets of tuna fleets, we thought that most of you were sweet. [...] So long and thanks for all the fish." [1]

[1] Joby Talbot, from the movie The Hitchhiker's Guide To The Galaxy, 2005

Abstract

Deep Learning has revolutionized numerous fields, achieving state-of-the-art performance in areas like computer vision, natural language processing and applied sciences. This progress has led to its integration into increasingly critical applications, with Deep Neural Networks already guiding crucial decisions in areas like autonomous vehicles, financial trading, mortgage assignment, hiring procedures, weather forecast, medical diagnosis, satellite management etc. Due to their intricate structure and the lack of adequate interpretability tools, Neural Networks's predictive process remains largely incomprehensible to humans. As a result, regulators are becoming increasingly concerned about the technology's potential failures and unexpected, but potentially harmful, behaviour. By demanding guarantees, regulators aim to ensure the responsible and ethical development and deployment of Deep Learning for the benefit of society. This thesis delves into two among the several crucial challenges regulators are concerned with: Uncertainty Quantification and Privacy Preservation. For each of these areas, we provide a brief definition and detail our contributions to their development.

Uncertainty Quantification and Generalization under Distribution Shift Given an input and a task the Neural Networks have been trained to solve, they will generally provide an output prediction. For humans to trust these predictions, it would be useful to extract a quantity that summarizes how reliable the prediction is. This quantity can be used by the humans themselves to accept or reject it, or for downstream systems to adequately process the outputs of upstream components or their input samples. Since an absolute notion of reliable prediction is elusive, part of the literature has focused on the problem of defining some tasks and metrics that capture the notion of reliability in a quantitative way for specific applications. The problem of estimating the reliability of a prediction becomes particularly important when dealing with inputs whose distribution is different from the one from which the training data was sampled (i.e., under Covariate Shift). In such cases, the predictions are known to become significantly less accurate: effective Uncertainty Estimation is therefore essential to help flag these potentially inaccurate outputs. In this thesis we provide the following contributions to the field:

- **Improve the reliability of image classifiers trained using Mixup**: Mixup is a technique that has been shown to induce improved performance for image classifiers. However, we show

that, despite it improves some reliability metrics, it degrades others. We provide an empirical explanation to the phenomenon and we correct it by modifying the loss function. This results in further performance improvements and improved uncertainty estimation properties of the models.

- **Correcting the false belief Transformers and Self-Attention provide superior reliability and accuracy under Distribution Shift**: When Vision Transformers were popularised, several papers started claiming Transformers could learn more robust representations that would generalise better under Covariate Shift and yield improved Uncertainty Estimation. This property was allegedly attributed to the presence of the Self-Attention component. Through an empirical study, we disprove this belief and show Convolutional architectures can perform similarly. We also conduct several experiments that suggest both models based on convolutional or self-attention biases can learn spurious features that prevent generalization in conditions of Covariate Shift.

- **Examining the potential of synthetic data from Diffusion Models to train classifiers that are more robust to Covariate Shift** Our study represented the first attempt to use Diffusion Models in order to augment the training set of an image classifier in order to perturb the environmental variables that are specific to the training distribution in order to learn robust representations that generalise better under Covariate Shift. We conclude that although significantly more effective than previously existing techniques, this approach still underperforms with respect to augmenting the training set of the classifier by retrieving images presenting similar perturbations from the training set of the Diffusion Model.

Privacy Preservation Neural Networks are often trained on sets containing private data whose leakage could be harmful either for individuals (e.g., they contain personal information) or society (e.g., contain military data, like satellite positions). It has been shown that malicious users may query these models in order to extract private information contained in the training set. For this reason, several techniques have been developed both to protect from these privacy attacks and to develop stronger attacks in order to audit the privacy preserving ability of models. In this thesis we provide the following contributions to the field:

- **Reducing the utility cost induced by privacy-preserving linear probing using semi-private learning and dimensionality reduction techniques** The gold standard Privacy Preserving technique is Differential Privacy (DP). Informally, it guarantees the likelihood an adversary may correctly predict whether a sample was or not in the training set can be bounded. The downside of DP training techniques is that they yield a significant performance degradation. We propose a simple method that uses a small amount of public data in order to

reduce the sample complexity of DP learning by reducing the dimensionality of the learning problem.

- **Produce the first analysis studying the memorization of private information in document-based Visual Question Answering systems** We propose a simple technique to estimate whether private information has been memorised by Visual Question Answering systems. We analyse which training factors yield to stronger memorisation and which type of information the malicious user needs in order to successfully extract memorised data. We also propose a heuristic countermeasure that reduces the likelihood the models may regurgitate training data.

Contents

List of Figures

List of Tables

1 Introduction

This is an integrated thesis containing work published in leading peer-reviewed conferences. In this Introduction, we provide a brief overview of the two safety areas this thesis focuses on: Uncertainty Estimation and Privacy Preservation. we provide a brief abstract outlining the contribution of each Chapter, the list of papers it is based on and the contributions provided by the author of this thesis to each of the papers. Each of these chapters is self-contained with its own related work section centred around the contribution of the paper.

Contents

As the integration of Deep Learning systems escalates across ubiquitous domains, the necessity for the development of techniques that guarantee their safety intensifies. While Deep Learning exhibits immense potential for advancements in areas like healthcare and autonomous vehicles, without safeguards, these algorithms can introduce unintended risks. This thesis contributes to the development and understanding of techniques that target two safety areas: Uncertainty Estimation (under Covariate Shift) and Privacy Preservation.

The area of Uncertainty Estimation is concerned with quantifying how reliable the predictions of Deep Learning models are. Nowadays, Neural Networks process their inputs by passing them through several layers of non-linear transformations that depend on millions if not billions of parameters. Despite the production of the outputs from the inputs is fully specified by the network's architecture and parameter values, it is hard for humans to understand the reason why some predictions are produced and what may cause wrong predictions. For this reason, it is essential to develop techniques that allow to quantify how reliable a prediction is. This quantity can be used in several ways. For instance, it is used to identify potentially incorrect outputs automatically, to inform downstream decision processes that need to be uncertainty-aware or allow a cost-effective collection of additional training data to fix known failures. This is particularly important when the inputs of the network undergo significant forms of covariate shift with respect to the training inputs (e.g., due to a change of the environment in which the inputs were sampled with respect to the training environment). We provide a brief overview of Uncertainty Estimation in Deep Learning and how it relates to our work in Section 1.1.

The area of Privacy Preservation aims at protecting the privacy of user information contained in the training data. Indeed, it has been observed that malicious users can query state-of-the-art models and extract (either completely or partially) training samples, some of which may contain personal information (e.g., phone numbers, home and email addresses etc.). Although various techniques have been proposed to prevent the extraction of such information, these may come at a significant utility cost (e.g., reducing the quality of the data generated by generative models), may be extremely computationally expensive or come with no theoretical guarantee. We provide a brief overview of Privacy Preserving Deep Learning and how it relates to our work in Section 1.2.

1.1 A Brief Overview of Uncertainty Estimation (under Covariate Shift) in Deep Learning

Although Deep Neural Networks have achieved remarkable success in various domains due to their ability to learn complex patterns from vast amounts of data, a critical limitation to their applicability in safety-critical domains is their inherent lack of transparency regarding their decision-making processes. Unlike traditional statistical models, deep learning models often do not provide an inherent measure of uncertainty associated with their predictions. In domains like autonomous vehicles, medical diagnosis or satellite management (see [Pinto et al., 2020, Acciarini et al., 2020, 2021]), where incorrect predictions can have severe consequences, uncertainty estimation becomes crucial. Knowing the model's confidence in its prediction allows for flagging potentially unreliable outputs and triggering human intervention when necessary. The importance of uncertainty estimation in deep learning is further amplified under covariate shift, a scenario where the distribution of the covariates used at test time differs from the training one. In such cases, models trained on data that doesn't reflect the real-world scenario in which they are deployed can become overly confident in their incorrect predictions. Uncertainty estimation techniques can help mitigate this risk. By providing a measure of the model's confidence in its outputs, even under covariate shift, we can identify situations where the model is likely to be wrong.

In this work we will mostly focus on the following well-established tasks in the literature:

- **Maintaining high accuracy under Covariate Shift [Quionero-Candela et al., 2009]**. Given a classifier has been trained on some data distribution, the goal is to maintain high accuracy also on samples that are collected under conditions that modify the input values without altering the label set (e.g., if a classifier is trained on photos of dogs captured in daylight, we would like to maintain high classification accuracy also for images captured at night).

- **Calibration [Osborne, 1991]**. Given a classifier's output, we would like the probability distributions it produces to have a frequentist meaning. The goal of calibrating a classifier is to align the frequency with which predictions with a certain confidence are correct with the confidence values themselves.

- **Misclassification Detection [Condessa et al., 2015]**. More practically than calibrating a classifier, the goal of misclassification detection is to use the confidence of the classifier in order to automatically detect when it is likely to be wrong.

- **Out-of-Distribution Detection (also known as Open-Set Detection) [Bendale and Boult, 2015]**. In closed-set classification, the set of labels observed at training time is a discrete, finite set. The goal of out-of-distribution detection is to leverage the confidence of the classifier in order to identify inputs that do not belong to any of the known classes, and therefore no correct prediction is possible.

Our work relates to two different research efforts performed in the literature:

- **Developing more robust systems, possibly producing better uncertainty estimation**. A wide amount of techniques have been developed. Some techniques (approximately) apply Bayesian probabilistic principles in order to build distributions of predictions and extract uncertainty metrics from it [Gal and Ghahramani, 2016b, Pearce et al., 2018, Wenzel et al., 2020, Kristiadi et al., 2020, Hobbhahn et al., 2021]. Other techniques, that proved to be the long standing state-of-the-art both due to their simplicity and their effectiveness, leverage ensembling in order to build such distributions [Lakshminarayanan et al., 2016]. Ensembles have been observed not only to improve the uncertainty estimation capabilities, but also to obtain higher accuracy under covariate shift [Liu et al., 2020b]. However, they are expensive (both at training and inference, the time and memory complexity scale linearly in the number of ensemble members). For this reason, techniques aiming at compressing ensembles have been proposed [Huang et al., 2017, Wen et al., 2020]. Alternatively, techniques leveraging single models, including three of ours [Pinto et al., 2022c, Yuan et al., 2024, Joy et al., 2022], have been developed [van Amersfoort et al., 2020, Hsu et al., 2020, Osborne, 1991, Yuan et al., 2024].

- **Understanding which architectural components induce better uncertainty estimation**. Understanding why neural networks produce overconfident but wrong predictions is a wide area of research. Some works attribute this to the choice of the activation networks [Hein et al., 2019, Kristiadi et al., 2020]. When transformers were popularised in vision, several works have attempted to attribute the arguably superior reliability of transformers to self-attention [Paul and Chen, 2022, Minderer et al., 2021, Zhang et al., 2021c]. In contrast, we find convolutional neural network can be as reliable [Pinto et al., 2022a], and therefore conclude robustness and reliability depend on either other architectural components (e.g., choice of the activation functions, types of normalization layers etc.) or training procedures (e.g., extensive pretraining on large amounts of data).

1.2 A Brief Overview of Privacy Preserving Deep Learning

User data is one of the key ingredients of modern machine learning, presenting a double-edged sword. On the one hand, it offers a potent opportunity to enhance model performance. By incorporating vast quantities of user data, algorithms can learn intricate patterns and relationships, leading to increased utility in tasks like image recognition or recommendation systems. This translates to a more personalized and efficient user experience. However, this reliance on user data also introduces significant privacy risks. A vast literature [Shokri et al., 2017, Carlini et al., 2022b, Ye et al., 2022] has shown malicious actors may exploit these models to potentially extract sensitive user information. This raises critical concerns about training these models on sensitive data (e.g., medical, satellite etc.). A lack of defense mechanisms may yield to an economically and socially damaging erosion of trust in the technology. Therefore, striking a balance between harnessing the power of user data for improved model performance and safeguarding user privacy is a crucial challenge in the field of machine learning.

In order to establish trust, it is essential to carry out two opposite but complementary tasks:

- **Privacy Defense** Researchers propose a wide range of techniques, both with theoretical guarantees or without (heuristic) about the likelihood an attacker can infer some information about the training data.
- **Privacy Auditing** Given a system, the goal is to understand what kind of information an attacker can effectively infer from querying the system. When the system implements some form of defense, the goal is to understand how effective the defense is, and, in case theoretical guarantees are available, whether the implementation of the defense was incorrect and violates the claimed privacy guarantees.

Privacy Defense The field of privacy-preserving machine learning (PPML) offers a diverse set of techniques to mitigate the risks of private data leakage. The gold-standard approach is Differential Privacy (DP) [Dwork et al., 2006]. In the context of neural network training, DP provides a theoretical upper bound on the likelihood an attacker may reliably discern whether a sample was present or not in the training set of a model. This is obtained by bounding the influence of each sample on the training algorithm and obfuscating its impact by adding a calibrated amount of noise to it. This method comes with a mathematically treatable definition that enjoys nice guarantees that align with an intuitive notion of privacy. Among these, the ones essential for deep learning are: 1) closure under post-processing, 2) closure under compositions. Informally, the closure under

post-processing states that once the privatisation is performed, no form of postprocessing that does not involve the private data itself can weaken the privacy guarantees provided. The closure under compositions indicates that sequential applications of DP algorithms constitute a DP algorithm, and the resulting guarantees can be easily computed. These two properties are essential to obtain DP variants of sequential learning algorithms like Stochastic Gradient Descent (obtaining DP-SGD [Abadi et al., 2016]). This privacy preserving technique will be the most discussed across this thesis. However, the tighter the provided guarantees, the more this technique induces a utility degradation (e.g., it may cause extreme accuracy drops when applied to training classifiers), which prevents its adoption in many real-world scenarios. For this reason, several works [Tramer and Boneh, 2021, Nguyen et al., 2020, Panda et al., 2022], including ours [Pinto et al., 2024a], aim at reducing the utility degradation it induces.

Privacy Auditing The definition of Membership Inference Attack (MIA) [Shokri et al., 2017] is complementary to the privacy definition offered by DP. The goal of a MIA is to reliably predict whether a sample was or not in the training set of a given model. While these attacks represent the de facto standard for auditing DP systems, other forms of auditing exist. For instance, exposure attacks [Carlini et al., 2019] measure the greater likelihood a model assigns to a specific input in order to estimate whether it could be part of the training set. Alternatively, extraction attacks (among which, one designed by us) aim at making the models directly output the training data [Carlini et al., 2021, Pinto et al., 2024b]. Property inference attacks are concerned with identifying global properties of the training data distribution [Ganju et al., 2018], while model inversion attacks aim at reconstructing the original training data from the trained model itself [Fredrikson et al., 2014].

1.3 Thesis Outline and Contributions

This thesis contributes to the development and analysis of techniques that allow to deploy deep learning systems providing useful uncertainty estimation metrics and preserve the privacy of users supplying training data. We now list the content of all the chapters contained in this integrated thesis and all the papers associated to each of them.

Chapter 2 This chapter explores a potential drawback of Mixup, a popular technique for boosting the accuracy of image classifiers. While Mixup effectively reduces overconfidence on in-distribution (i.i.d.) samples, it also weakens the model's ability to detect out-of-distribution (OOD) data –

inputs whose labels fall outside the training classes. Our analysis suggests this behavior stems from the label smoothing component of Mixup. To address this, we propose a modified Mixup approach that achieves significant improvements in accuracy on both i.i.d. and covariate-shifted data, while also enhancing calibration and OOD detection.

The chapter is mostly a marginal revision of the following publication:

> [Pinto et al., 2022c]: **Francesco Pinto**, Harry Yang, Ser-Nam Lim, Philip H.S. Torr, Puneet K. Dokania "RegMixup: Mixup as a Regularizer Can Surprisingly Improve Accuracy and Out Distribution Robustness " *NeurIPS 2022*,

Full Text. Link To Paper

A preliminary and incomplete version of the work appeared at the NeurIPS 2021 DistShift Workshop [Pinto et al., 2021b].

Contribution Statement: For [Pinto et al., 2022c], Francesco and Puneet brainstormed to identify the methodology. Francesco implemented all the experiments and run most of them. Harry executed some code on an external cluster to accelerate some experiments. Puneet suggested using inter-cluster and intra-cluster similarities to analyse some of the properties of the models, while Francesco has proposed to use it on covariate-shifted data once it was observed there were no significant patterns on i.i.d. data. Francesco proposed and performed the rest of the analyses. Francesco performed the writing with the help of Puneet. Phil provided some general advice.

Chapter 3 This chapter challenges the prevailing notion that Transformers inherently possess superior robustness to covariate shift and uncertainty estimation compared to Convolutional Neural Networks (CNNs). While previous works have championed this view based on the self-attention module's ability to process the entire input early on, we argue that this does not guarantee robust representations. Our empirical analysis demonstrates that self-attention can also become fixated on local and potentially misleading patterns, leading to unreliable predictions.

Furthermore, we show that state-of-the-art CNNs outperform their Transformer counterparts in terms of robustness to data shifts. Both architectures achieve similar performance in detecting unknown classes. Interestingly, Transformers exhibit slightly better calibration but fall short in misclassification detection. This is particularly noteworthy because the CNNs employed share most architectural elements with Transformers (e.g., choice of activations, normalization layers, depth etc.) except for the lack of self-attention modules. This clearly indicates the observed improvements are attributable either to other architectural changes or to the different pre-training strategies. Instead of advocating for one architecture over the other, this work aims to steer the research community away

from simplistic conclusions. Assigning a "silver bullet" role to specific components like self-attention is inaccurate. Our research has spurred a wave of subsequent studies reaching similar results.

The chapter is mostly a marginal revision of the following publication:

> [Pinto et al., 2022a]: **Francesco Pinto**, Philip H.S. Torr, Puneet K. Dokania "An Impartial Take to the CNN vs Transformer Robustness Contest " *ECCV 2022*,

Full Text. Link To Paper

A preliminary and incomplete version of the work appeared at the NeurIPS 2021 DistShift Workshop [Pinto et al., 2021a].

Contribution Statement: For [Pinto et al., 2022a], Francesco came up with the idea, the argument and designed, implemented and run all the experiments. Puneet helped with the writing. Phil provided some general advice.

Chapter 4 In this chapter, we study how modern synthetic image generators can be used to approximate interventions. Theoretical work attributes the lack of generalization of machine learning models to covariate-shifted inputs to the change of spurious or environmental factors between the training and test domains. While collecting data in controlled experiments in which the experimenter controls for these factors (intervenes on these variables) may be impossible or extremely expensive, synthetic data generators can be used to approximate such manipulations. Across several generation and editing techniques, we observe that although using generators to augment the original inputs outperforms any previously state-of-the-art augmentation technique, generation from noise and textual prompts or retrieval from the training set of the generator proves to be even more effective.

The chapter is mostly a marginal revision of the following publication:

> [Yuan et al., 2024]: Jianhao Yuan*, **Francesco Pinto***, Adam Davies*, Philip H.S. Torr "Not Just Pretty Pictures: Toward Interventional Data Augmentation Using Text-to-Image Generators" *ICML 2024*,

Full Text. Link To Paper

Contribution Statement: For [Yuan et al., 2024], Francesco came up with the idea and lead the project, Jianhao implemented and run most of the experiments. Francesco run the large-scale experiments (DomainNet). Adam developed and tested prompting strategies, image filtering and textual inversion experiments; prototyped various conditioning mechanisms. Francesco, Jianhao and Adam brainstormed and collaborated on the experimental design and identifying the focus points of the analysis. Francesco and Adam wrote the paper, Jianhao reported the results, the plots and the tables. Phil provided general advice.

Chapter 5 This chapter introduces PILLAR, a novel semi-private learning technique. PILLAR mitigates the performance drop caused by training with Differential Privacy by leveraging small amounts of publicly available data. We leverage the public data in order to identify the principal components of the features extracted by a pre-trained neural network. The private data is then projected on such components. We show this provably reduces the sample complexity of private training, and empirically yields significant utility improvements on a wide range of datasets. Notably, PILLAR excels in challenging scenarios with limited private data, strict privacy requirements and in presence of distribution shifts both between the training distribution of the pre-trained neural network and the private data, and between the public and private data.

The chapter is mostly a marginal revision of the following publication:

> [Pinto et al., 2024a]: **Francesco Pinto***, Yaxi Hu*, Fanny Yang, Amartya Sanyal, "PILLAR: How to make semi-private learning more effective. " *SatML 2024*,

Full Text. Link To Paper

Contribution Statement: For [Pinto et al., 2024a], Francesco and Amartya came up with the methodology and the analyses to be performed. Francesco designed, implemented and run the experiments. Yaxi and Amartya proved the theoretical resuls. Francesco, Fanny and Amartya helped in making the empirical and theoretical parts of the work to be cohesive. Francesco, Yaxi, Fanny and Amartya helped writing the paper.

Chapter 6 In this chapter, we propose the a memorization analysis of document-based Visual Question Answering (VQA) systems. By removing some parts of the training input image that contains the answer to the input training question, we can point out the model has indeed memorized the answer and it can be extracted from the memory of the model. While this may resemble the known phenomenon of shortcut learning in VQA systems, our work evidences this memorization phenomenon cannot be attributed to learning shortcuts that are known occur at a distributional level (e.g., responding that the colour of the grass is green as a result of the frequent co-occurrence of the question and the answer in the training set) by performing a counterfactual memorization analysis. Furthermore, we show these models can memorize uniquely occurring Personal Identifying Information (PII), therefore posing a possible privacy threat. After analysing the factors that may yield to the extractability of memorized answers, we propose a simple countermeasure that improve the utility of the VQA systems and prevents the extraction of sensitive data.

The chapter is mostly a marginal revision of the following publication:

[Pinto et al., 2024b]: **Francesco Pinto**, Nathalie Rauschmayr, Florian Tramèr, Philip H.S. Torr, Federico Tombari "Extracting Training Data From Document-Based VQA Models", *ICML 2024*.

Full Text. Link To Paper

Contribution Statement: For [Pinto et al., 2024b], Francesco and Nathalie identified the research problem. Francesco came up with the extraction methodology. Francesco, Nathalie and Florian brainstormed about how to contextualise the work with respect to the literature and proposed analyses to be performed. Francesco implemented and run most of the experiments, with help from Nathalie for data pre-processing and the execution of some analyses. Francesco and Nathalie wrote the paper, Federico and Florian revised the draft with comments and minor edits. Federico and Phil provided general advice. Chiyuan Zhang, Michal Lukasik and Vaishnavh Nagarajan (not included in the author's list) provided useful feedback on preliminary versions of the draft.

2

RegMixup: Mixup as a Regularizer Can Surprisingly Improve Accuracy and Out-of-Distribution Robustness

Contents

Abstract

In this chapter, we show that the effectiveness of the well celebrated Mixup [Zhang et al., 2018] can be further improved if instead of using it as the sole learning objective, it is being utilized as an additional regularizer to the standard cross-entropy loss. This simple change not only provides much improved accuracy but also significantly improves the quality of the predictive uncertainty estimation of Mixup in most cases under various forms of covariate shifts and out-of-distribution detection experiments. Note, standard Mixup would otherwise yield much degraded performance on out-of-distribution detection experiments, perhaps, as we show empirically, because of its tendency to learn models that exhibit high-entropy which makes it more difficult to differentiate between in-distribution and out-distribution samples. To show the efficacy of our approach (RegMixup), we provide thorough analyses and experiments on vision datasets (CIFAR-10/100 and ImageNet) and compare it with a suite of well-known approaches for reliable uncertainty estimation.

2.1 Introduction

In real-world machine learning applications one is interested in obtaining models that can reliably process novel inputs. However, though deep learning models have enabled breakthroughs in multiple fields, they are known to be unreliable when exposed to samples obtained from a distribution that is different from the training distribution. Larger the extent of this difference between train and test distributions, more unreliable these models normally are. This has led to a growing interest in developing approaches that encourage reliable predictions from a model even when they are exposed to unseen situations [Liu et al., 2020a,c, Wen et al., 2021, Lakshminarayanan et al., 2017]. Most of these recent approaches either use expensive ensembles, or propose non-trivial modifications to the neural network architectures in order to obtain reliable models. These approaches, in most cases, trade in-distribution performance (accuracy) to gain reliability when exposed to: (1) out-of-distribution (OOD) samples; and (2) covariate shifted (CS) [Quionero-Candela et al., 2009] samples.

Towards developing reliable models, we investigate the well known Mixup technique [Zhang et al., 2018] as it is extremely popular in improving both a model's accuracy and and its robustness [Wen et al., 2021, Hendrycks et al., 2019a]. It has already been observed that Mixup can help in retaining good accuracy when the test inputs are affected by superficial variations that do not affect the target label (i.e., they undergo CS) [Hendrycks et al., 2019a]. However, we find that its performance degrades significantly when exposed to completely unseen samples with potentially different labels than the ones it was exposed to during training (OOD). This is undesirable as it is not guaranteed that the real-world test samples will always belong to one of the categories seen during training, and in such situations, a model should be able to reliably reject those samples instead of making wrong predictions. We observe that the primary reason for such poor OOD performance is that Mixup, because of the way it is trained, ends up providing high predictive entropy for almost all the samples it receives. Therefore, it becomes difficult to differentiate in-distribution samples from out-of-distribution ones. We would like to highlight that our observation is in contrast to the prior work [Thulasidasan et al., 2019] which suggests that Mixup provides reliable uncertainty estimates for OOD data as well.

We propose a simple yet effective fix to the aforementioned issue with Mixup: we suggest to train on a mixture distribution that combines the original training data distribution and the Mixup vicinal distribution together. We call this approach RegMixup. We provide proper justifications behind this proposal and show that such simple modification to the well known Mixup can further improve its performance on a variety of experimental settings.

One of the core strengths of our approach is its **simplicity**. As opposed to the recently proposed techniques to improve uncertainty estimation like SNGP [Liu et al., 2020a] and DUQ [van Amersfoort et al., 2020], it does not require any modifications to the architecture and is extremely simple to implement. *It does not trade accuracy in order to improve uncertainty estimates*, and is a single deterministic model, hence, extremely efficient compared to the highly competitive Deep Ensembles (DE) [Lakshminarayanan et al., 2017]. Summary of our contributions:

- We provide a simple modification to Mixup that significantly improves its in-distribution, covariate shift, and out-of-distribution performance.
- Through extensive experiments using ImageNet-1K, CIFAR10/100 and their various CS counterparts along with multiple OOD datasets we show that, overall, RegMixup outperforms recent state-of-the-art single-model approaches. In most cases, it outperforms the extremely competitive and expensive DE as well.

2.2 RegMixup: Mixup as a regularizer

Preliminary The principle of risk minimization [Vapnik, 1991] is to estimate a function $f \in \mathcal{F}$ that, for a given loss function $\ell(.,.)$, minimizes the expected risk over the entire data-distribution $P(\mathbf{x},\mathbf{y})$. The risk to be optimized is defined as $R(f) = \int \ell(f(\mathbf{x}),\mathbf{y}) dP(\mathbf{x},\mathbf{y})$. Since the distribution $P(\mathbf{x},\mathbf{y})$ is unknown, a crude yet widely used approximation is to first obtain a training dataset $\mathcal{D} = \{(\mathbf{x}_i,\mathbf{y}_i)\}_{i=1}^n$ sampled from the distribution P and then obtain f by minimizing the *empirical risk* defined as $R_e(f) = 1/n \sum_{i=1}^n \ell(f(\mathbf{x}_i),\mathbf{y}_i)$. This is equivalent to approximating the entire data-distribution space by a finite n number of Delta distributions positioned at each $(\mathbf{x}_i,\mathbf{y}_i)$, written as $P_e(\mathbf{x},\mathbf{y}) = 1/n \sum_{i=1}^n \delta_{\mathbf{x}_i}(\mathbf{x})\delta_{\mathbf{y}_i}(\mathbf{y})$. This approximation to the original risk minimization is widely known as the *Empirical Risk Minimization (ERM)* [Vapnik, 1991].

ERM has been successfully used in a plenty of real-world applications and undoubtedly has provided efficient and accurate solutions to many learning problems. However, it is straightforward to notice that the quality of such ERM solutions would rely on how closely P_e mimics the true distribution P, and also on the capacity of the function class $\mathcal{F}$. In situations where the function class is extremely rich with very high capacity (like neural networks), learning can be prone to undesirable behaviours such as overfitting and memorization. Therefore a good approximation to P is generally needed to enforce suitable inductive biases in the model. To this end, for a fixed training dataset, one could potentially fit a richer distribution, instead of a delta distribution, in the vicinity of each input-output pair to estimate a more informed risk computed in a *region* around each

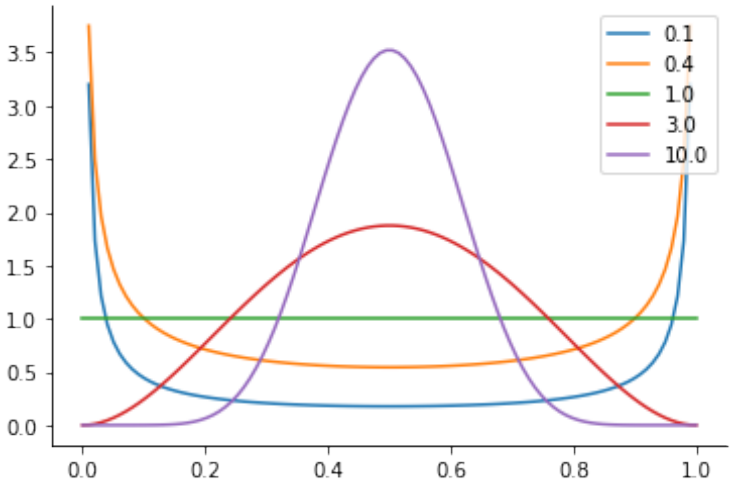


Figure 2.2.1: $\texttt{Beta}(\alpha, \alpha)$ pdf for varying α.

sample. This is precisely the principle behind *Vicinal Risk Minimization (VRM)* [Chapelle et al., 2000]. The approximate distribution in this case can be written as $P_v(\mathbf{x}, \mathbf{y}) = 1/n \sum_{i=1}^{n} P_{\mathbf{x}_i, \mathbf{y}_i}(\mathbf{x}, \mathbf{y})$[1]. Therefore, the *vicinal risk* boils down to

$$R_v(f) = \frac{1}{n} \sum_{i=1}^{n} \int \ell(f(\mathbf{x}), \mathbf{y}) dP_{\mathbf{x}_i, \mathbf{y}_i}(\mathbf{x}, \mathbf{y}). \tag{2.1}$$

In situations where the integral is intractable, Monte Carlo estimate with m samples can be used:

$$\int \ell(f(\mathbf{x}), \mathbf{y}) dP_{\mathbf{x}_i, \mathbf{y}_i}(\mathbf{x}, \mathbf{y}) \approx \frac{1}{m} \sum_{j=1}^{m} \ell(f(\bar{\mathbf{x}}_j), \bar{\mathbf{y}}_j); \quad (\bar{\mathbf{x}}_j, \bar{\mathbf{y}}_j) \sim P_{\mathbf{x}_i, \mathbf{y}_i}(\mathbf{x}, \mathbf{y}). \tag{2.2}$$

Several approaches in the literature can be seen as a special instance of VRM. For example, training a neural network with multiple augmentations is a special case where the augmented inputs are the samples from the unknown vicinal distribution. A widely used application of VRM is the procedure to obtain *robust base classifier* to design certifiable classifiers[2]. For example, if $P_{\mathbf{x}_i, \mathbf{y}_i}(\mathbf{x}, \mathbf{y}) = P_{\mathbf{x}_i}(\mathbf{x}) \delta_{\mathbf{y}_i}(\mathbf{y})$ and $P_{\mathbf{x}_i}(\mathbf{x})$ is a Gaussian distribution centered at $\mathbf{x}_i$, Equation 2.2 can be computed by taking the average loss over inputs $\bar{\mathbf{x}}$ perturbed with gaussian noise, while keeping the target labels same. Minimizing such a risk would lead to a classifier that is robust to additive noise bounded within an ℓ_2 ball. This is exactly the procedure that has been widely adopted in

[1] Note, the original VRM paper uses $P_{\mathbf{y}_i}(\mathbf{y}) = \delta_{\mathbf{y}_i}(\mathbf{y})$ which simply is a special case of this notation.
[2] Note, the literature does not mention this as an instance of VRM.

randomized smoothing literature in order to obtain a certifiably smooth classifier from a base neural network [Lecuyer et al., 2019, Cohen et al., 2019]. Below we discuss another highly effective use case of VRM called *Mixup* which is the main focus of this work.

Mixup The vicinal distribution in Mixup is defined as

$$P_{\mathbf{x}_i,\mathbf{y}_i}(\mathbf{x},\mathbf{y}) = \mathbb{E}_\lambda[(\delta_{\bar{\mathbf{x}}_i}(\mathbf{x}), \delta_{\bar{\mathbf{y}}_i}(\mathbf{y}))],$$

where $\lambda \sim \texttt{Beta}(\alpha,\alpha)$, $\alpha \in (0,\infty)$, $\bar{\mathbf{x}}_i = \lambda\mathbf{x}_i + (1-\lambda)\mathbf{x}_j$ and $\bar{\mathbf{y}}_i = \lambda\mathbf{y}_i + (1-\lambda)\mathbf{y}_j$. Note that the vicinal distribution in this case not only depends on $(\mathbf{x}_i,\mathbf{y}_i)$ but also on another input-output pair $(\mathbf{x}_j,\mathbf{y}_j)$ drawn from the same training dataset. For a fixed α (parameter of the Beta distribution, refer Figure 2.2.1), implementing Mixup would require taking multiple Monte Carlo samples[3] for each datapoint (refer Eq. equation 2.2) which can be computationally prohibitive. Therefore, in practice, only one sample ($m = 1$) per Beta distribution per pair of samples from a batch is considered at a time. Although this procedure might look like a crude approximation to the original objective, it has resulted in highly promising results in a variety of applications and is very well accepted in the research community. Without undermining the remarkable effectiveness of such a successful approach, we would like to focus on two of its potential limitations:

- **Small cross-validated $\alpha \ll 1$:** The shape of the vicinal distribution depends on the hyperparameter of the Beta distribution (refer Figure 2.2.1), therefore, the values of α will decide the strength of the interpolation factor λ. However, how far the interpolated $(\bar{\mathbf{x}},\bar{\mathbf{y}})$ is from the true $(\mathbf{x}_i,\mathbf{y}_i)$ or $(\mathbf{x}_j,\mathbf{y}_j)$ is decided based on the cross-validation performance on a held out test dataset. Since high values of α would encourage $\lambda \approx 0.5$ resulting in $\bar{\mathbf{x}}$ that is very different from $\mathbf{x}$ (*hence inducing a mismatch between train and test distributions*), the cross-validated value of α for Mixup always turns out to be very small ($\alpha \approx 0.1$) in order to obtain good generalization. Note, $\alpha \approx 0.1$ leads to very sharp peaks at 0 and 1 (refer Figure 2.2.1). Therefore, effectively, Mixup ends up *slightly* perturbing a clean sample in the direction of another sample even though the vicinal distribution has the potential to explore a much larger portion of the interpolation space.

- **High-entropy behaviour:** As mentioned $m = 1$ in practice, therefore, it is very unlikely that the interpolation factor is perfectly zero or one even for small values of α. Thus, the model is

[3]We do not discuss how m depends on α, however, it is intuitive that a relationship exists. For example, for small α, Beta distribution will have peaks at the extremes, therefore, relatively smaller m should suffice compared to moderate αs.

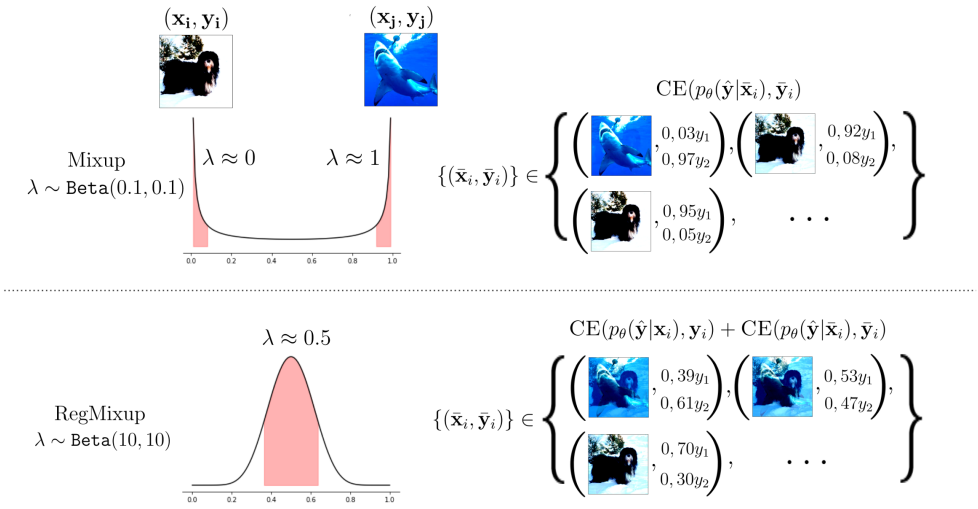
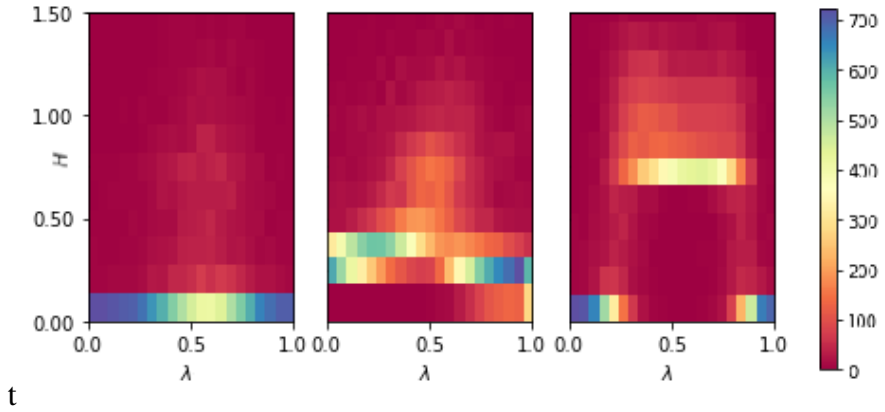


Figure 2.2.2: Mixup vs RegMixup in practice. Illustration of how the cross-validated α affects the shape of the `Beta` distribution in both cases. Red regions represent 80% of the probability mass. Mixup typically samples $\lambda \approx 0$ or 1, while for RegMixup $\lambda \approx 0.5$. In the first case, one of the two interpolating images dominates the interpolated one; in the latter case, a wide variety is obtained.

never exposed to uninterpolated samples during training and hence it always learns to predict interpolated (or smoothed) labels $\bar{\mathbf{y}}$ for every input. Just like DNNs with cross-entropy loss are overconfident because of their high capacity and target Delta distribution [Guo et al., 2017], DNNs with Mixup turns out to be *relatively less confident* because the network retains its high capacity but observes only target smoothed labels. This underconfident behaviour results in high-entropy for both in-distribution and out-of-distribution samples. This is undesirable as it will not allow predictive uncertainty to reliably differentiate in-distribution samples from out-of-distribution ones, thus, leading to poor robustness.

We validate the consequences of the above limitations of Mixup with a simple experiment. In Figure 2.2.3, we provide heat-maps that show how the entropy of the predictive distribution (softmax output) varies when interpolating between samples belonging to different classes. The heat-map is created as follows. We train a WideResNet28-10 (WRN) [Zagoruyko and Komodakis, 2016] using CIFAR-10 (C10) dataset. Then, we randomly choose 1K pairs of samples $\{\mathbf{x}_i, \mathbf{x}_j\}$ from the dataset such that $y_i \neq y_j$[4]. For each pair, we synthesize 20 samples $\bar{\mathbf{x}}$s using equally spaced λs between 0 to 1. The heat-map is then created using all the 20K samples. The intensity of each (λ, H) bin in the heat-map indicates the number of samples in that bin. Note, DNN (i.e., a network trained with

[4]Note, it is highly likely that $\mathbf{y}_i \neq \mathbf{y}_j$ even if we do not impose this constraint as the problems under consideration have multiple classes.



t

Figure 2.2.3: Heatmaps of the entropy profiles as the interpolation factor λ between samples of two classes varies. Left (**DNN**), Middle (**Mixup**), Right (**RegMixup**). RegMixup provides high entropy barrier separating in-distribution from out-of-distribution samples.

vanilla cross-entropy loss) shows low entropy (overconfidence) irrespective of where the interpolated sample lies. However, Mixup shows high entropy almost always (undercofindence). As also shown in Table 2.2.1, although Mixup provides improved accuracy compared to DNN for in-distribution and covariate shift experiments, this high entropy behaviour makes it much worse than DNN when considering the to out-of-distribution detection task. For example, when SVHN is used as the OOD dataset, the performance of Mixup drops by nearly 8.47% compared to DNN. This clearly shows that the predictive entropy of Mixup is not discriminative enough. However, there is a clear improvement of nearly 5% for covariate shift experiments, implying the Mixup augmentations do improve robustness in this aspect. Note that, in the context of calibration, Mixup's underconfident behaviour (which is equivalent to providing high predictive entropy) was also noticed by [Wen et al., 2021].

RegMixup We now provide a very simple change to the way Mixup has been used in the literature that not only avoids the aforementioned limitations, but also significantly improves in-distribution and covariate-shift accuracies of Mixup. We use the following approximation to the data-distribution

$$P(\mathbf{x},\mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n}(\gamma\delta_{\mathbf{x}_i}(\mathbf{x})\delta_{\mathbf{y}_i}(\mathbf{y}) + (1-\gamma)P_{\mathbf{x}_i,\mathbf{y}_i}(\mathbf{x},\mathbf{y})),$$

where $P_{\mathbf{x}_i,\mathbf{y}_i}(\mathbf{x},\mathbf{y})$ is the mixup vicinal distribution and $\gamma \in [0,1]$ is the mixture weight. *The above approximation is simply an explicit assemble of ERM and VRM* based approximate distributions. Though, in theory, VRM subsumes ERM, however, we argue that because of the finite amount of samples drawn (thus reducing the chances that $\lambda \in \{0,1\}$), explicitly combining them might result

in a *practically* more expressive approximation. We provide extensive experimental evidence to support this hypothesis. Implementation wise, for each sample $\mathbf{x}_i$ in a batch, another sample $\mathbf{x}_j$ is randomly drawn from the same batch to obtain interpolated $\bar{\mathbf{x}}_i$ and $\bar{\mathbf{y}}_i$, and then the following loss is minimized

$$\mathrm{CE}(p_\theta(\hat{\mathbf{y}}|\mathbf{x}_i), \mathbf{y}_i) + \eta\, \mathrm{CE}(p_\theta(\hat{\mathbf{y}}|\bar{\mathbf{x}}_i), \bar{\mathbf{y}}_i), \tag{2.3}$$

where CE(.,.) denotes the standard cross-entropy loss, the hyperparameter $\eta \in \mathbb{R}_{\geq 0}$, and $p(.)$ the softmax output of a neural network parameterized by θ. Note, dividing Eq. equation 2.3 by $(1+\eta)$ will result in the assemble with $\gamma = 1/1+\eta$. In practice, simply using $\eta = 1$ worked highly effectively.

What does this simple modification bring to the model? (1) A better empirical approximation to the underlying vicinal data distribution. This is because of the fact that irrespective of the values of m and α, the model will always be exposed to the clean training samples as well. (2) The interpolation factor λ can potentially explore a much wider space as the presence of clean samples might help in controlling the performance drop due to the train/test distribution shift. Therefore, if $\alpha \ll 1$ was actually the most effective solution, the cross validation would automatically find it.

Practical implication of such simple modification on the behaviour and the performance of the model

- **Large cross-validated** $\alpha \gg 1$: The model is now able to explore strong interpolations because of the additional cross-entropy term over the unperturbed training data. Interestingly, the cross-validated α that we obtained in fact is very high ($\alpha = 10$) leading to $\lambda \approx 0.5$. Therefore, as opposed to the standard Mixup, RegMixup prefers having strong diverse interpolations during training. Refer Figure 2.2.2 for visualizations. Clearly, interpolated samples in RegMixup are diverse and contain features from both the images in the pair.

- **Well behaved entropy:** It is straightforward to notice that a value of $\lambda \approx 0.5$ would lead to $\bar{\mathbf{x}}$ that is a heavy mix of two samples (mimicking OOD samples, refer Fig. 2.2.2), and the corresponding target label vector $\bar{\mathbf{y}}$ would have almost equal masses corresponding to the labels of the interpolating samples. Since this configuration almost represents a maximum entropy condition in the choice over the two labels, minimizing $\mathrm{CE}(p_\theta(\mathbf{y}|\bar{\mathbf{x}}_i), \bar{\mathbf{y}}_i)$ can be seen as maximizing a proxy to entropy defined over the label support of $\mathbf{y}_i$ and $\mathbf{y}_j$ (note, exact entropy maximization would encourage equal probability masses of 0.5 for these labels). We find this observation intriguing as RegMixup naturally obtains a cross-validated α that leads to a maximum likelihood solution on in-distribution data and increases the entropy over heavily interpolated samples that do not naturally occur in the original distribution. This

Models	C10 (Test) Accuracy (↑)	Cov. Shift C10-C Accuracy (↑)	OOD Detection C100 AUROC (↑)	SVHN AUROC (↑)	T-ImageNet AUROC (↑)
DNN	96.14	76.60	88.61	96.00	86.44
Mixup	97.01	81.68	83.17	87.53	84.02
RegMixup (Our)	**97.46**	**83.13**	**89.63**	**96.72**	**90.19**

Table 2.2.1: In-distribution, covariate shift, and out-of-distribution detection experiments using WideResNet28-10 trained on C10. C10-C represents the corrupted version of CIFAR-10.

is an extremely desirable property as it allows models to differentiate between in and out distribution samples. Thus, Mixup acts as the addition (on top of a cross-entropy term) of an approximation of a maximum-entropy **regularizer** for interpolated (out-of-distribution) samples.

The entropy heat-map in Figure 2.2.3 clearly shows that as opposed to DNN and Mixup, the entropy for RegMixup is very low for λ close to either 0 or 1, however, it increases and remains high for all other intermediate interpolation factors, *practically creating an entropy barrier*. Also, as shown in Table 2.2.1, RegMixup provides improvement in accuracy (both in-distribution and covariate-shift) and out-of-distribution robustness. An interesting observation is that Mixup's performance on OOD (T-ImageNet) dropped by 2.4% compared to DNN whereas RegMixup performed 3.75% better than DNN. Thus, RegMixup effectively improved the effectiveness of Mixup by nearly 6.15%.

2.3 Experiments

Datasets and Network Architectures We employ the widely used WideResNet28-10 (WRN) [Zagoruyko and Komodakis, 2016] and ResNet50 (RN50) [He et al., 2016] architectures. We train them on CIFAR-10 (C10) and CIFAR-100 (C100) datasets. We employ RN50 to perform experiments on ImageNet-1K [Deng et al., 2009a] dataset. We report the average of all the metrics computed on 5 seeds. For further details about the code base and the hyperparameters, refer to Section A.1.1.

For **Covariate Shift (CS)** experiments on models trained on C10 and C100, we resort to the widely used CIFAR10-C (C10-C) and CIFAR100-C (C100-C), corrupted versions of C10 and C100 [Hendrycks and Dietterich, 2019]. These datasets are made by applying 15 synthetically generated but *realistic corruptions* at 5 degrees of intensity on the test sets of C10 and C100,

respectively. For CIFAR-10, we also use the CIFAR-10.1 (C10.1) [Recht et al., 2018a] and CIFAR-10.2 (C10.2) [Lu et al.] datasets designed to test the generalization of CIFAR-10 classifiers to *natural covariate shifts*. To the best of our knowledge there is no such analogous dataset for CIFAR-100. For ImageNet-1K experiments, we use the widely considered ImageNet-A (A) [Hendrycks et al., 2019b], ImageNet-R (R) [Hendrycks et al., 2021], ImageNetv2 (V2) [Recht et al., 2019], and ImageNet-Sketch (SK) [Wang et al., 2019] for covariate shift experiments.

For **Out-of-Distribution (OOD)** detection, following SNGP [Liu et al., 2020a], we use C100 and SVHN [Netzer et al., 2011] as OOD for models trained on C10. Similarly, for models trained on C100, we use C10 and SVHN as OOD. Additionally, we also consider the Tiny-ImageNet (T-ImageNet) dataset [Le and Yang, 2015] as OOD set in both the cases (filtering out common classes). For models trained on ImageNet-1K, we use ImageNet-O (O) [Hendrycks et al., 2019b] as the OOD dataset.

Methods considered for comparisons Besides the natural comparison of our method with Mixup [Zhang et al., 2018] and networks trained via ERM on the standard cross-entropy loss (which we will refer to as DNN), we consider several other methods from the OOD detection and CS literature. For models trained on C10 and C100, we consider:

- DNN-SN and DNN-SRN: taking inspiration from [Liu et al., 2020a], we consider DNN models trained with Spectral Normalization (SN) [Miyato et al., 2018] and Stable Rank Normalization (SRN) [Sanyal et al., 2020] to control the bi-Lipschitz constant of the networks, which has been shown to affect the generalization properties of neural networks.

- SNGP: Spectrally Normalized Gaussian Process [Liu et al., 2020a].

- DUQ: Deterministic Uncertainty Quantification [van Amersfoort et al., 2020].

- KFAC-LLLA: KFAC-Laplace Last Layer Approximation [Kristiadi et al., 2020] makes a model Bayesian at test time by taking the Laplace approximation of the last layer [Ritter et al., 2018].

- AugMix [Hendrycks et al., 2019a]: A data augmentation technique that applies randomized augmentations to an input while enforcing them to be consistent during training. With the recommended hyperparameters in the paper, it is almost 4× slower than DNN during training while having the same inference requirements.

- DE: Deep Ensembles [Lakshminarayanan et al., 2017] with 5 members, requiring 5× more compute than most single-model approaches such as DNN.

We would like to mention that our approach (RegMixup) is almost 1.5× slower and Mixup is about 1.2× slower than DNN training while having the same inference requirements. Due to the high computational requirements to train on ImageNet-1K, for ImageNet-1K experiments we consider DNN, Mixup and the two other strongest baselines: AugMix and DE. We also cross-validate the hyperparameters on a 10% split of the test set, which is removed at test time.

Few missing experiments: Below we provide extensive experiments for proper benchmarking. For some combinations of datasets and architecture choices we were not able to produce good results for (even after extensive hyperparameter search). Hence, some of the entries in the tables are missing. For example, we could not make DUQ work on C100 as it exhibited unstable behaviour. We could not produce promising results for SNGP on RN50 CIFAR experiments using their official implementation. Similarly for AugMix RN50 experiments on C10 and C100. We chose not to report these suboptimal numbers. Further details can be found in Section A.1.1.

Table entries: Bold represents the best among all the single-model approaches, and underline represents the best among all including the expensive Deep Ensembles.

Note, we do not consider methodologies requiring access to an external dataset during training (either for CS or OOD) as not only this would be an unfair comparison[5], but we believe assuming such knowledge is against the goal of this work, which is to develop models robust to unknown scenarios.

2.3.1 RegMixup improves accuracy on IND and test samples

IND Small-scale CIFAR experiments In Table 2.3.1, we compare the accuracy of various approaches on the in-distribution test sets of C10 and C100, respectively. Clearly,

- RegMixup outperforms Mixup in all these experiments. In fact, RegMixup is the best performing one among all the single-model approaches.

- These improvements are non-trivial. For instance, on WRN trained on C100, it outperforms DNN and Mixup by 1.67% and 0.65%. It outperforms SNGP with a significant margin of 4.05%.

[5]Since all the methods we consider do not leverage this information. RegMixup only relies on in-distribution training data, just like other approaches, and makes no assumption on the type of CS nor on the OOD inputs.

Methods	IND Accuracy			
	WRN28-10		RN50	
	C10 (Test) Accuracy (↑)	C100 (Test) Accuracy (↑)	C10 (Test) Accuracy (↑)	C100 (Test) Accuracy (↑)
DNN	96.14	81.58	95.19	79.19
Mixup	97.01	82.60	96.05	80.12
RegMixup (**our**)	**97.46**	**83.25**	**96.71**	**81.52**
DNN-SN	96.22	81.60	95.20	79.27
DNN-SRN	96.22	81.38	95.39	78.96
SNGP	95.98	79.20	-	-
DUQ	94.7	-	-	-
KFAC-LLLA	96.11	81.56	95.21	79.41
Augmix	96.40	81.10	-	-
DE (5×)	96.75	83.85	96.23	82.09

Table 2.3.1: Accuracies (%) on IND samples for models trained on C10 and C100

Note how the single-model approaches specifically designed to provide reliable predictive uncertainty estimations (for example, SNGP) underperform even compared to the vanilla DNN in terms of IND accuracy. In order to provide improved uncertainty estimates (as we will soon show), they trade clean data accuracy. This type of behaviour is not observed in RegMixup.

Covariate Shift Small-scale CIFAR experiments For C10-C and C100-C, as typical in the literature, we report the accuracy averaged over all the corruptions and degrees of intensities in Table 2.3.2. It is evident that our approach produces a remarkable improvement in the average accuracy compared to **all** the baselines (except AugMix, we discuss later why that might be the case). For instance, for C100-C experiments, our method achieves an accuracy improvement of **6.9%** over DNN, of **3.86%** over DE, and of **2.45%** over Mixup. Similarly, for C10-C WRN, our method achieves an improvement of almost **6.53%** over DNN, of **4.81%** over DE, and of **1.45%** over Mixup.

For natural covariate shift datasets C10.1 and C10.2 as well, RegMixup outperforms **all** the baselines (including AugMix). For instance, on C10.2, it obtains an improvement of **3.26%** over DNN, of **1.5%** over Mixup, and of **2.46%** over the expensive DE.

IND Large-scale ImageNet-1K experiments RegMixup scales to ImageNet-1K and exhibits improved accuracy with respect to both Mixup, DNN, and AugMix. In particular, it is **1.08%** better than DNN, **0.53%** better than Mixup and **0.80%** better than AugMix. DE in this case is the best performing one.

Methods	Covariate Shift Accuracy							
	WRN28-10				R50			
	C10-C	C10.1	C10.2	C100-C	C10-C	C10.1	C10.2	C100-C
	Accuracy (↑)				Accuracy (↑)			
DNN	76.60	90.73	84.79	52.54	75.18	88.58	83.31	50.62
Mixup	81.68	91.29	86.55	56.99	78.63	90.03	84.61	53.96
RegMixup (**our**)	83.13	**92.79**	**88.05**	59.44	**81.18**	**91.58**	**86.72**	**57.64**
DNN-SN	76.56	91.01	84.72	52.61	74.88	88.26	82.96	50.55
DNN-SRN	77.21	90.88	85.24	52.54	75.40	88.61	83.49	50.48
SNGP	78.37	90.80	84.95	57.23	-	-	-	-
DUQ	71.6	-	-	50.4	-	-	-	-
KFAC-LLLA	76.56	90.68	84.68	52.57	75.18	88.34	83.50	50.85
AugMix	**90.02**	91.6	85.9	**68.15**	-	-	-	-
DE (×5)	78.32	92.17	85.59	55.58	77.63	90.05	85.00	53.91

Table 2.3.2: Accuracies (%) on covariate shifted samples for models trained on C10 and C100.

	IND Acc.	Covariate Shift Acc				OOD Det.
	ImageNet-1K	R	A	V2	Sk	O
	Acc (↑)	Acc (↑)	Acc (↑)	Acc (↑)	Acc (↑)	AUROC (↑)
DNN	76.60	36.41	2.76	64.53	24.72	55.97
Mixup	77.15	39.05	3.29	64.58	26.34	55.54
RegMixup (**our**)	**77.68**	**39.76**	**5.96**	**65.66**	**26.98**	**57.05**
AugMix	76.88	38.29	2.63	64.94	25.61	56.91
DE (5×)	78.22	38.94	2.11	66.68	27.03	53.29

Table 2.3.3: ImageNet accuracies (%) on IND and CS samples, and OOD detection performance.

Covariate Shift Large-scale ImageNet-1K experiments In Table 2.3.3 we report the results for common ImageNet-1K covariate-shift test sets. As it can be seen, RegMixup is either the best performing among all the single-model approaches, or it is the absolute winner. For instance, on ImageNet-A RegMixup performs **2.67%** better than Mixup and **3.20%** better than DNN. Similarly, on ImageNet-V2 it performs **1.08%** better than Mixup and **1.13%** more than DNN. RegMixup also outperforms AugMix on all the considered datasets, while is outperformed by DE on ImageNet-V2 (by **1.02%**) and performs competitively on ImageNet-SK.

These experiments show the strong generalization performance of RegMixup under various CS scenarios. They also show that it does not trade clean data accuracy to do so.

Why AugMix performs extraordinarily well on synthetically corrupted C10-C and C100-C but not on natural distribution shift C10.1 and C10.2? Looking at Table 2.3.2 one can observe AugMix's extremely good performance on the C10-C and C100-C. However, at the same time, the model is underperforming with respect to RegMixup on C10.1 and C10.2. Similarly, AugMix is

	WRN28-10						RN50					
	CIFAR10 (In-Distribution)			CIFAR100 (In-Distribution)			CIFAR10 (In-Distribution)			CIFAR100 (In-Distribution)		
Out-of-Distribution	**C100**	**SVHN**	**T-ImageNet**	**C10**	**SVHN**	**T-ImageNet**	**C100**	**SVHN**	**T-ImageNet**	**C10**	**SVHN**	**T-ImageNet**
Methods	**AUROC (↑)**			**AUROC (↑)**			**AUROC (↑)**			**AUROC (↑)**		
DNN	88.61	96.00	86.44	81.06	79.68	80.99	88.61	93.20	87.82	79.33	82.45	79.89
Mixup	83.17	87.53	84.02	78.37	78.68	80.61	84.24	89.40	84.89	77.02	76.86	80.14
RegMixup (**our**)	89.63	**96.72**	**90.19**	**81.27**	**89.32**	**83.13**	**89.63**	**95.39**	**90.04**	**79.44**	**88.66**	**82.56**
DNN-SN	88.56	95.59	87.71	81.10	83.43	82.26	88.19	93.46	87.55	79.20	80.78	79.90
DNN-SRN	88.46	96.12	87.43	81.26	85.51	82.41	88.82	93.54	87.82	78.77	82.39	79.70
SNGP	**90.61**	95.25	90.01	79.05	86.78	82.60	-	-	-	-	-	-
KFAC-LLLA	89.33	94.17	87.81	81.04	80.32	81.57	89.54	93.13	88.32	79.30	82.80	80.17
Aug-Mix	89.78	91.3	88.99	81.10	76.64	80.56	-	-	-	-	-	-
DE (5×)	91.25	97.53	89.52	83.26	85.07	83.40	91.38	96.90	90.5	81.93	85.08	82.15

Table 2.3.4: Out-of-distribution detection results (%) for WideResNet28-10 and ResNet50 for models trained on C10 and C100. See Appendix A.1.1 for the cross-validated hyperparameters.

outperformed by RegMixup on all ImageNet CS experiments (and OOD as will be shown soon). This seems to suggest that although the augmentations used during the training of AugMix are not exactly same as that of the corrupted test dataset, they tend to benefit from synthetic forms of covariate shifts, hence the dramatic improvement in these particular scenarios.

2.3.2 Out-of-Distribution detection experiments

Following the standard procedure [Liu et al., 2020a], we report the performance in terms of AUROC[6] for the binary classification problem between in- and out-distribution samples. The predictive uncertainty of the model is typically used to obtain these curves. Given an uncertainty measure (normally entropy, refer Section A.2 for an extensive discussion), it is important for models to be more uncertain on OOD samples than on in-distribution samples to be able to distinguish them accurately. This behaviour leads to better AUROC.

We report the OOD detection results for models trained on C10 and C100 in Table 2.3.4. In the case of SVHN as the OOD dataset, our method either outperforms all the existing approaches (2.54% higher AUROC than the runner-up SNGP in C100 experiments) or it turns out to be a runner-up with the gap of 0.81% compared to the top performing and expensive DE for C10 experiments.

In the case of CIFAR as the OOD dataset, DE turns out to be the best performing one. RegMixup, along with KFAC-LLLA, is the runner-up in the case of C100 experiments while SNGP is the runner-up in C10 experiments. In the case of Tiny-ImageNet as the OOD dataset, our method outperforms all single-network models and DE when the in-distribution set is C10, but it is slightly inferior to DE when the in-distribution set is C100. Similarly, on ImageNet-O detection, RegMixup outperforms all the baselines.

[6]Area Under Receiver Operating Characteristic curve.

Methods	IND			
	WRN28-10		RN50	
	C10 (Test)	C100 (Test)	C10 (Test)	C100 (Test)
	AdaECE (↓)		AdaECE (↓)	
DNN	1.34	3.84	1.45	2.94
Mixup	1.16	1.98	2.17	7.47
RegMixup (**our**)	**0.50**	**1.76**	**0.94**	**1.53**
SNGP	0.87	1.94	-	-
Augmix	1.67	5.54	-	-
DE (5×)	1.04	3.29	1.28	2.98

Table 2.3.5: CIFAR IND calibration performance (%).

	IND	Covariate Shift			
	ImageNet-1K (Test)	ImageNet-R	ImageNet-A	ImageNet-V2	ImageNet-Sk
	AdaECE (↓)	AdaECE (↓)	AdaECE (↓)	AdaECE (↓)	AdaECE (↓)
DNN	1.81	13.56	44.90	4.13	14.48
Mixup	**1.29**	12.08	44.63	4.28	15.26
RegMixup (**our**)	1.37	13.30	**41.18**	**3.38**	15.35
AugMix	2.05	**11.24**	42.83	3.94	**14.26**
DE (5×)	1.38	13.55	42.88	4.02	17.32

Table 2.3.6: ImageNet calibration performance on IND and CS datasets (%).

Methods	Covariate Shift							
	WRN28-10				R50			
	C10-C	C10.1	C10.2	C100-C	C10-C	C10.1	C10.2	C100-C
	AdaECE (↓)				AdaECE (↓)			
DNN	12.62	4.13	8.81	9.94	12.29	4.36	8.89	19.76
Mixup	7.93	4.39	7.44	10.45	**10.75**	5.72	10.59	12.63
RegMixup (**our**)	9.08	**2.57**	**6.83**	**7.93**	11.37	**2.89**	**6.74**	**11.47**
SNGP	11.34	4.36	8.33	10.43	-	-	-	-
AugMix	**4.56**	3.23	8.33	12.15	-	-	-	-
DE (5 ×)	10.31	2.60	7.50	12.36	12.68	4.10	6.94	12.36

Table 2.3.7: CIFAR CS calibration performance (%).

2.3.3 Calibration on In-Domain and Covariate Shifted Inputs

Additionally, we provide the calibration performance of various competitive approaches. Briefly, calibration quantifies how similar a model's confidence and its accuracy are [Osborne, 1991]). To measure it, we employ the recently proposed Adaptive ECE (AdaECE) [Mukhoti et al., 2020]. For all the methods, the AdaECE is computed after performing temperature scaling [Guo et al., 2017] with a cross-validated temperature parameter. We also provide the AdaECE without temperature scaling in Section A.3.

In terms of calibration on in-domain test sets (refer Tables 2.3.5 and A.3.2), our method either remarkably improves the AdaECE with respect to Mixup and DNN or performs competitively (on ImageNet-1K).

Under covariate shift (refer Tables A.3.2 and 2.3.7), on corrupted inputs, RegMixup underperforms with respect to Mixup on C10-C, but not on C100-C. On all other C10 covariate shift datasets RegMixup outperforms Mixup and DNN. Considering also the other baselines, except for the case of C10-C (in which AugMix significantly outperforms any other baseline on WRN28-10), our method provides the best calibration in all other cases. For example, on C100-C experiments on WRN28-10, in terms of AdaECE, RegMixup obtains a **4.43%** improvement over DE, **2.52%** over Mixup, and **2.47%** over SNGP. Though RegMixup outperformed all other approaches in 12 scenarios out of total 17 scenarios presented here, it is clear there is no single method that outperforms any other in all considered settings.

2.4 Conclusion

We proposed RegMixup, an extremely simple approach that combines Mixup with the standard cross-entropy loss over unperturbed training data. We conducted a wide range of experiments and showed that RegMixup significantly improves the reliability of uncertainty estimates of deep neural networks, while also providing a notable boost in the accuracy. We showed that RegMixup did not just outperform Mixup, it also outperformed most recent state-of-the-art approaches in providing reliable uncertainty estimates while improving the in-distribution accuracy.

3

An Impartial Take to the CNN vs Transformer Robustness Contest

Contents

Abstract

Following the surge of popularity of Transformers in Computer Vision, several studies have attempted to determine whether they could be more robust to distribution shifts and provide better uncertainty estimates than Convolutional Neural Networks (CNNs). The almost unanimous conclusion is they are, and it is often conjectured more or less explicitly that the reason of this supposed superiority is to be attributed to the self-attention mechanism. In this paper we perform extensive empirical analyses showing that the recent state-of-the-art CNNs (particularly, ConvNeXt [Liu et al., 2022b]) can be as robust and reliable or even sometimes more than the current state-of-the-art Transformers. However, there is no clear winner. Therefore, although it is tempting to state the definitive superiority of one family of architectures over another, they might just be the two sides of the same coin, enjoying similar extraordinary performances on a variety of tasks while also suffering from similar vulnerabilities due to overparameterization (e.g, simplicity bias).

3.1 Introduction

Transformers are a family of neural network architectures that became extremely popular in Natural Language Processing and are characterised by the pervasive use of the attention mechanisms as defined in [Vaswani et al., 2017]. Before Vision Transformers (ViT) [Dosovitskiy et al., 2020] were introduced, Transformers were considered hard to scale to computer vision applications due to the prohibitive computational complexity and memory requirements of the self-attention mechanism. Since then, several transformer variants that are efficient to train having performance more competitive with the state-of-the-art CNNs like BiT [Kolesnikov et al., 2020] (e.g. [Yuan et al., Touvron et al., 2020, Liu et al., 2021b] just to mention a few popular transformers) have been proposed.

The effectiveness of transformers compared to CNNs in computer vision applications has led to recent interest in comparing them in terms of providing reliable predictive uncertainty (calibration) and robustness to distribution shifts. The almost unanimous conclusion of the literature is that transformers exhibit: (1) better calibration [Minderer et al., 2021], (2) better robustness to covariate shift [Paul and Chen, 2022, Zhang et al., 2021d, Bai et al., 2021], and (3) better uncertainty estimation for tasks like out-of-distribution detection (OoD) [Fort et al., 2021, Bai et al., 2021]. Presently, the above conclusions are misleading because (1) very recent convolutional architectures (ConNeXt) were not available for comparisons, (2) some of the aforementioned studies change the training scheme of Transformers to bring them closer to CNNs (effectively, making them perform suboptimally) for their analysis, and (3) the choice of the evaluation metrics is often not carefully justified and the most subtle aspects of the interpretation of the results were not identified. Additionally, when it comes to explaining the outcome of the analysis, which mostly leads to concluding that Transformers are superior, the credit is often given (more or less explicitly) to the most prominent feature that distinguishes Transformers from CNNs: the self-attention mechanism. Thus, a fair comparison and an understanding of whether and how self-attention modules would allow learning superior features compared to convolutional models is needed before providing a definitive answer regarding the superiority of one over another.

Taking a step in this direction, we evaluate the robustness and reliability of most recent state-of-the-art Transformers (ViT [Dosovitskiy et al., 2020] and SwinT [Liu et al., 2021b]) and CNN architectures (BiT [Kolesnikov et al., 2020] and ConvNeXt [Liu et al., 2022b]) on ImageNet-1K [Deng et al., 2009a] dataset. We do not modify the training recipes of CNNs and Transformers so that they are at their current best during comparisons. The main takeaways of our work are:

1. Transformers, just like CNNs, also suffer from the so-called **simplicity bias** [Shah et al., 2020]. They are somewhat as good as CNNs in finding shortcuts (undesirable) to solve the

desired task. Therefore, as opposed to the common notion, despite the capability of the self-attention modules to perceive globally the image from the early layers, Transformers tend to focus on easy-to-discriminate parts of the input and conveniently ignore other complex-yet-discriminative ones. Hence, similarly to CNNs, they might just be learning to combine sets of simple, rather than more complex or rich, features. Based on this experiment, we discourage the common trend in the literature to give unnecessary praise to the self-attention module of Transformers anytime these perform better against CNNs. More theoretical developments, analyses, and well-thought experiments are needed to support such claims.

2. We show that under out-of-distribution evaluation, CNNs and Transformers **perform equally well**. We also highlight why evaluating OoD using AUPR in situations of data imbalance (which generally is the case) might give the false impression of one model being significantly superior to others.

3. In-distribution calibration of the best performing CNN model (in terms of accuracy) is better than the best performing Transformer. However, there is **no clear winner** that performs the best in all the experiments including covariate shift.

4. Again, there is **no clear winner** in detecting misclassified inputs.

These takeaway points also suggest that the inductive biases induced in CNNs by using the design components popularised by Transformers (e.g. GeLU [Hendrycks and Gimpel, 2016a] activations, LN normalization [Ba et al., 2016] etc.), but without using the self-attention, might be sufficient to bridge the gap between the two in terms of robustness. However, this again is a speculation as there are too many variables in designing a model (from architectural design choices to optimization algorithms) and the interplay between them is not well understood yet. It is also not clear if a rich model design (e.g, self-attention) will have significant impact on the properties of the intermediate representations given that all the current models are extremely overparameterized and the simplicity bias exists in practice.

3.2 Experimental Design and Choices

3.2.1 Setup

Models. We consider state-of-the-art convolutional and non-convolutional models for our analysis.

1. BiT [Kolesnikov et al., 2020] is a very commonly used fully convolutional architecture in the literature. It is a ResNet variant that has been shown to achieve state-of-the-art accuracy on ImageNet classification and that, with an appropriate fine-tuning procedure, transfers well to many other datasets. Several variants of BiT exist, in this paper we consider the BiT-R50x1, BiT-R50x3, BiT-R101x1, BiT-R101x3, BiT-R152x2, BiT-R152x4 (where R50/101/152 indicates the ResNet variant, and the multiplicative factor scales the number of channels).

2. ConvNeXt [Liu et al., 2022b], a very recent fully convolutional architecture that is close to the non-convolutional Transformer models in terms of training recipe and design, and has been shown to produce either comparable or superior performance to Transformers on several large-scale datasets. ConvNeXt exemplifies how pushing the state-of-the-art in a family of networks can yield architecture design choices that, if properly adapted, can benefit other families of networks. Our conclusions heavily rely on the careful architecture design process of the authors of ConvNeXt. ConvNeXt-B, ConvNeXt-L, ConvNeXt-XL variants.

3. ViT [Dosovitskiy et al., 2020], the first successful vision transformer, still exhibiting state-of-the-art performance. ViT-B/16 and ViT-L/16[1], where B and L indicate the capacity (B = Base, L = Large), while 16 indicates the input token patch size.

4. SwinTransformer [Liu et al., 2021b] variants, one of the state-of-the-art Transformer architecture implementing a hierarchical architecture employing a shifting window mechanism. We consider the Swin-B and Swin-L, where B and L indicate the capacity as before (we consider the variants with patch size of 4 pixels and shifted windows of size 7).

Training. Unless stated otherwise, all the considered architectures have been pre-trained on ImageNet-21k [Ridnik et al., 2021] and fine-tuned on ImageNet-1k [Deng et al., 2009a]. For our experiments we use the trained checkpoints available in the timm library [Wightman, 2019b].
Datasets. Since the in-distribution dataset is ImageNet-1K, we use ImageNet-A [Hendrycks et al., 2019b], ImageNet-R [Hendrycks et al., 2021], ImageNetv2 [Recht et al., 2019], ImageNet-Sketch [Wang et al., 2019] for the *domain-shift* experiments. For *out-of-distribution* detection, we use ImageNet-O [Hendrycks et al., 2019b]. For our preliminary analyses to understand *existing biases* in Transformers and CNNs, we use ImageNet9 [Xiao et al., 2020a], the Cue-Conflict Stimuli dataset [Geirhos et al., 2018a], and also *synthesize* a dataset by combining MNIST and CIFAR-10 datasets.

[1]We omit ViT-B/32 ViT-L/32 since we find they always underperform with respect to ViT-B/16 and ViT-L/16, in agreement with [Paul and Chen, 2022]. We also omit DeiT [Touvron et al., 2020] as we observe it underperforms with respect to SwinTransformers.

3.2.2 Yet Another Analysis?

Before we begin discussing our analysis, we would like to mention how our point of view is different from the recent work.

Closest to ours is a recent analysis presented by [Bai et al., 2021] which involves the simplest architectures for both Transformers (DeiT) and CNNs (ResNet-50), and drops transformer-specific training techniques (for instance, reducing training epochs to 100 from 300, removing augmentations and regularisation techniques etc.). This indeed brings DeiT down to CNNs in terms of training procedure, however, makes DeiT underperform significantly. Although they derive interesting insights, the applicability of these insights for a practitioner willing to identify the most robust and best performing model is somehow limited. Therefore, we not only consider a wider variety of CNNs and Tranformers in our analysis, we also do not modify their standard training recipes so that their best performance is being compared. Note, [Zhang et al., 2021d] only compares with the extremely simple CNN variants.

We would also like to highlight that comparing different models based on their capacity (determined solely based on their number of parameters) might lead to wrong conclusions. How well a model would preform in practice is heavily dependent on the nature and the composition (hierarchy, depth etc.) of the underlying functions, not just on the number of parameters. To provide a widely known example, an MLP with one hidden layer and enough hidden units (large number of parameters) can theoretically fit any function, known to be a universal function approximator [Hornik et al., 1989, Cybenko, 1989], however, in practice, they underperform compared to a deep network (with same or even less number of parameters). The interaction of inductive biases and training procedures plays an important role towards finding solutions that generalise well.

Therefore, although the number of parameters can be a proxy for comparing model capacity, in practice, it can be misleading and should be of concern only when compute and memory constraints are imposed. Even if they are imposed, a practitioner will always find the best performing model satisfying such constraints rather than choosing a model based on the parameter count[2].

[2]Consider that ViT-L/32 has about 307M parameters, ViT-L/16 has 305M, yet ViT-L/32 requires about 15GFLOPS, while ViT-L/16 requires about 61GFLOPS, and ViT-L/32 exhibits lower accuracy and robustness than ViT-B/32 [Paul and Chen, 2022]

3.3 Empirical Evaluation and Analysis

3.3.1 Are Transformer Features More Robust than CNN ones?

There is no clear answer to this question in the literature. It is known that for a model to generalise to previously unseen domains, its predictions should not depend on spurious features that are specific to the distribution from which the training and test in-domain sets are sampled from, but on robust features that generalise across other domains under covariate-shift [Quionero-Candela et al., 2009]. Typical examples of spurious features described in literature are the background's colour, textures and generally any simple pattern that correlates strongly with the labels in the training set but not in the test set [Arjovsky et al., 2019].

It is usually conjectured in the literature that Transformers might be learning more robust features than CNNs because of the ability of their self-attention modules to communicate globally between all the tokens contained in a given input [Paul and Chen, 2022]. Which, in fact, is equivalent to implicitly blaming the convolutional inductive biases of CNNs, which instead focus on local regions of the image, for their relatively poor robustness.

Before we begin comparing these two families in terms of robustness, here we discuss a few experiments to analyse their vulnerabilities. These experiments show that the sole presence of the self-attention mechanism is not sufficient for Transformers to neglect spurious features, and they result to be as biased as CNNs towards them.

(a) Samples containing cars

(b) Samples containing trucks

(c) From top to bottom: attention visualization at layer 1, 4 and 12 for the same input

Figure 3.3.1: Simplicity Bias Experiment for Transformer: For each triplet of images, from left to right: input test image, test image without pixels on which the attention values is below the 70% quantile and the attention map visualization.

	# params	SB			BB				TB
		In-domain	**R-MNIST**	**R-CIFAR**	**O** (↑)	**MS** (↑)	**MR** (↑)	**BG-Gap** (↓)	**CCS**(↑)
BiT-R50×1	25	100	48.39	100	94.57	83.21	76.2	7.00	31.09
BiT-R50×3	217	100	48.14	100	95.14	85.14	80.22	4.92	33.12
BiT-R101×1	44	100	48.50	99.94	94.17	81.28	75.19	6.09	32.81
BiT-R101×3	387	100	48.19	99.89	94.32	81.19	76.67	4.52	32.58
BiT-R152×2	232	100	48.39	99.94	94.64	80.05	75.09	4.95	35.47
BiT-R152×4	936	100	48.19	100	95.01	81.16	75.33	5.83	37.19
ConvNeXt-B	88	100	48.29	99.94	97.95	93.95	90.42	3.53	30.63
ConvNeXt-L	196	100	48.20	99.89	98.2	95.19	91.63	3.56	35.16
ConvNeXt-XL	348	100	48.75	99.69	98.49	95.23	92.3	2.93	36.95
ViT-B/16	86	100	48.59	99.79	97.36	92.35	88	4.34	30.78
ViT-L/16	304	100	52.79	95.66	98.02	94.05	90.05	4	47.19
Swin-B	87	100	48.75	99.64	97.75	90.94	86.47	4.47	26.95
Swin-L	195	100	48.69	99.74	98.02	92.99	88.47	4.52	30.08

Table 3.3.1: Simplicity bias (SB), Background bias (BB) and Texture bias (TB) experiments. In-domain indicates the accuracy when MNIST and CIFAR images are associated as in the training set; R-MNIST when MNIST digit is randomized without changing the CIFAR image; similarly for R-CIFAR. *A model suffers from SB if R-MNIST accuracy is close to random whereas R-CIFAR accuracy is close to In-domain accuracy* For **BB**, we report the absolute accuracy on the original (O), mixed-same (MS), and mixed-random (MR) datasets, respectively. **BG-Gap** defined as the difference in accuracy between MS and MR, quantifies the impact of background in producing correct classifications. For **TB** we report the CCS accuracy.

Simplicity Bias Experiment. The intent of this experiment is to compare Transformers and CNNs under situations where it is possible for a model to focus only on the simple discriminative features of the input and ignore the complex discriminative ones in order to perform well on the task. This clever experiment was proposed and analysed on CNNs by [Shah et al., 2020]. Following their work, we first create a binary classification task where the input $X = [\mathbf{x}, \bar{\mathbf{x}}]$ is composed of the concatenation of $\mathbf{x}$ and $\bar{\mathbf{x}}$, both discriminative, and learning features for either or both will lead to an accurate classifier. We design this task such that, say, $\bar{\mathbf{x}}$ is more complex[3] than $\mathbf{x}$. Then, under this setting, a trained classifier suffers from simplicity bias if (1) fixing $\mathbf{x}$ and randomly modifying $\bar{\mathbf{x}}$ in the input does not change its prediction, and (2) fixing $\bar{\mathbf{x}}$ and randomly modifying $\mathbf{x}$ in the input drops the test accuracy to the random prediction baseline. To create the dataset for this experiment, $\mathbf{x}$ comes from the MNIST dataset [Deng, 2012] (randomly sampled digit of a certain label) and $\bar{\mathbf{x}}$ from CIFAR-10 (randomly sampled image from a certain label of CIFAR-10). For instance, say digit **0** is associated to **car** and the whole image is assigned label $+1$, and digit **1** is associated to **truck** and the whole image is assigned label -1. Refer to the top left of Figure 3.3.1 for an example. During training, this relationship holds true for all the examples (in-domain). We fine-tune

[3]We understand that defining complexity is subjective. Here we assume that something that is visually more complex (having more colors, shapes, textures etc.) would require learning more complex features.

our classifiers on this dataset for 3 epochs (it is easy to converge on this dataset). At test time, we either randomise the MNIST part of the image (R-MNIST) or the CIFAR part of the image (R-CIFAR) for the analysis. Results reported in Table 3.3.1.

As can be seen, the accuracy is almost the same for all the models (except in ViT-L/16) even if the CIFAR (more complex) part of the input is completely randomized. However, the accuracy drops to nearly random (50%) when the MNIST (simpler) part of the input is randomized. This shows that both families, Transformers and CNNs, rely on MNIST for classification and are agnostic to the CIFAR component. Hence, both are prone to the simplicity bias. To understand which are the most prominent features leveraged by the transformer, we visualize the pixels to which the self-attention modules assign the highest attention values. We identify the pixels whose values in the attention map fall above the 70% quantile of the intensity values, and blacken the ones that fall below it in Figure 3.3.1. This figure confirms that the transformer's self-attention mechanism neglects complex features and assigns higher attention to simple features. Figure 3.3.1 (c) also shows how the self-attention changes through the layers of the transformer. At the first layer there is no specific focus on the MNIST digit, but as the layers progress (i.e. as the features specialise to be useful for the classification), the attention values increase around the digit.

Reliance on Backgrounds and Texture. Here we measure the performance of several architectures on a benchmark that measures the reliance of features on backgrounds and textures: ImageNet9 [Xiao et al., 2020a] and the Cue-Conflict Stimuli [Geirhos et al., 2018a].

The ImageNet9 dataset selects a subset of labels and images from the original ImageNet dataset. In our experiments we measure the accuracy on the full images of the dataset (original split), images in which the background has been swapped with the one of images of the same class (mixed-same), images in which the background has been swapped with the one of images of another class at random (mixed-random). The authors of this dataset suggest taking the gap between the accuracy on mixed-same and mixed-random as a quantifier of the reliance on background information to produce accurate predictions. As it can be seen from Table 3.3.1, some of the highest capacity BiT models do not rely more on the background than ViT-B/16, SWIN-B and SWIN-L. ConvNeXt models rely on backgrounds even less than transformers, suggesting that the self-attention mechanism is not responsible for the gap observed between low-capacity ResNets and Transformers.

The Cue-Conflict Stimuli dataset alters the texture information of an image using style transfer: given an image of a certain class, it deliberately uses as style image an image from another class. The purpose is to deceive classifiers that overly rely on textures to make predictions. As it can be seen in Table 3.3.1, although the top performing model is ViT-L/16 (with a significant margin), Swin Transformers exhibit an even heavier reliance on texture than ConvNeXt models, and ViT-B/16

performs comparably to ConvNeXt-B. This suggests that the sole presence of the self-attention in an architecture is not sufficient for the model to not be biased towards texture information.

Conclusions:

- Transformers can leverage spurious features just like CNNs. They both are prone to the simplicity bias, the background bias and the texture bias.

3.3.2 Out-of-Distribution Detection

Current notion in the literature is that Transformers are better than CNNs at detecting OoD samples [Paul and Chen, 2022].

We compare various CNN and Transformer models at the task of detecting ImageNet-O samples from ImageNet-1K. ImageNet-O contains 2K samples in 200 classes, while the subset of ImageNet-1K used as the corresponding in-distribution test set contains 10K samples [Hendrycks et al., 2019b] (therefore there is a stark imabalance in the number of samples belonging to the two sets). Both ImageNet-O and ImageNet-1K (test) samples are fed to the classifier, for each point an uncertainty score is computed and a binary threshold-based classifier is used to distinguish between them. Since the choice of the threshold depends on the risk exposure desired for a certain application, a standard evaluation procedure that considers all the risk thresholds computes the AUROC (Area Under the Receiver Operating Characteristic curve) and the AUPR (Area Under the Precision-Recall curve) [Hendrycks and Gimpel, 2016b].

AUPR vs AUROC? We start by observing that the apparent complexity in distinguishing ImageNet-O samples from ImageNet-1K observed in the literature (e.g. [Hendrycks et al., 2019b, Paul and Chen, 2022]) mostly depends on the interaction between specific evaluation choices and the usage of the AUPR metric. The AUPR, in the case of an imbalanced number of samples belonging to the positive and negative classes, is known to prefer one class over another. However, in the out-of-distribution evaluation setting, unless additional assumptions about the specific application domain are made, there is no preferred mistake: confusing an in-distribution sample with an out-of-distribution sample or viceversa are both equally important mistakes. To exemplify why the AUPR can yield misleading conclusions, in Table 3.3.2 we consider all possible assignments of the positive class and apply a rebalancing technique. The cited literature (which also concludes there exists a dramatic gap between CNN and Transformers performance) only reports values from the setting in which the positive class is assigned to the out-of-distribution samples (third column from the right), which can be misleading. In this setting, for instance, the performance of BiT-R50x1 is less than half of the performance of ViT-L/16, and extremely low (with respect to the attainable maximum of

	IND=1,OOD=0				IND=0,OOD=1			
	Imbalanced		**Balanced**		**Imbalanced**		**Balanced**	
	AUROC (↑)	AUPR (↑)	AUROC (↑)	AUPR (↑)	AUROC (↑)	AUPR (↑)	AUROC (↑)	AUPR (↑)
BiT-R50x1	65.17	90.15	65.17	65.81	65.17	23.30	65.17	60.13
BiT-R50x3	74.56	92.30	74.56	71.28	74.56	36.26	74.56	72.49
BiT-R101x1	70.34	91.35	70.34	68.75	70.34	28.53	70.34	66.11
BiT-R101x3	77.32	93.40	77.32	74.84	77.32	38.74	77.32	74.66
BiT-R152x2	77.46	93.51	77.46	75.23	77.46	38.24	77.46	74.43
BiT-R152x4	80.07	94.39	80.07	78.10	80.07	44.25	80.07	78.17
ConvNeXt-B	85.72	95.53	85.72	81.74	85.72	59.15	85.72	85.53
ConvNeXt-L	89.07	96.90	89.07	86.96	89.07	65.33	89.07	88.55
ConvNeXt-XL	<u>90.04</u>	97.19	<u>90.04</u>	88.11	<u>90.04</u>	68.50	<u>90.04</u>	89.75
ViT-B/16	79.89	95.26	79.89	82.30	79.89	36.77	79.89	73.77
ViT-L/16	**90.60**	97.85	**90.60**	91.27	**90.60**	64.58	**90.60**	88.90
Swin-B	83.74	95.29	83.74	81.01	83.74	52.93	83.74	82.80
Swin-L	87.76	96.55	87.76	85.67	87.76	62.51	87.76	87.27

Table 3.3.2: ImageNet-O: **Out-of-distribution** performance analysis when in-distribution samples are assigned label 1 and OoD label 0, and vice-versa (with and without rebalancing). AUROC is invariant whereas AUPR, as discussed, is extremely sensitive to these design choices. The best performing method according to the AUROC is in bold, the second best is underlined. As it can be seen, the gap between the two is marginal.

100). However, only rebalancing the number of samples[4] the performance of BiT-R50x1 rises to more than two thirds of the performance of ViT-L/16 (last column on the right). Alternatively, if the choice of the positive class is flipped, in an imbalance condition, one can obtain an absolute gap between the performance of BiT-R50x1 and ViT-L/16 of less than 8% (third column from the left). If one drew conclusions solely based on this column, one would think there is only a marginal difference between the performance of the two models. This gap widens when rebalancing the number of samples (fourth column from the left). This exemplifies how widely the AUPR can vary based on evaluation choices that, in lack of domain-specific assumptions, are arbitrary. On the other hand, the AUROC does not vary across all the considered evaluation setups, because it gives the same importance to both types of errors that can occur (confusing an ImageNet sample with an ImageNet-O sample or viceversa) and hence should be the privileged metric to describe OOD detection performance. These results allow us to conclude that ImageNet-O is evidently not as hard to distinguish from ImageNet as it is believed to be.

Comparing Transformers and CNNs From the AUROC values in Table 3.3.2 it is clear that the top-performing CNN (ConvNeXt-XL) is competitive to the top-performing Transformer (ViT-L/16). ConvNeXt-L outperforms Swin-L, and ConvNeXt-B outperforms Swin-B. The best performing

[4]We sample 4 more times the OOD set, so that both IND and OOD datasets have 1000 samples each. We could rebalance them also by randomly sampling 2000 out of the 10000 IND points, but this induces some variance in the metrics; we also observed that the average of this strategy coincides with the balancing strategy we discussed.

BiT (BiT-R152×4) outperforms ViT-B.

Conclusions:

- CNN models can perform as well as Transformers for out-of-distribution detection.
- In lack of assumptions about the application domain, the AUROC should be preferred as a metric for this task.

3.3.3 Calibration on In-Distribution and Domain-Shift

A model is said to be calibrated if its confidence (i.e. the maximum probability score of the softmax output) and its accuracy match. The idea is to attribute to the confidence the frequentist probabilistic meaning of counting the amount of times the model is correct. A calibrated classifier is considered reliable because it should produce wrong predictions with low confidence. However, it is well known that cross-entropy trained neural networks can many times be overconfident and wrong [Guo et al., 2017] (it is much less likely for modern architectures to be correct with low confidence on in-domain test sets).

In the attempt of measuring the calibration of a classifier, several measures have been proposed, and they mostly target the mismatch between confidence and accuracy. These measures are the Expected Calibration Error (ECE) [Naeini et al., 2015] and the Adaptive Calibration Error (AdaECE) [Mukhoti et al., 2020].

Is Low Calibration Error Enough for a Classifier to be Reliable? Before proceeding to discuss the tables of the calibration errors for each model, it is important to understand that the classifiers with lower calibration error might still be overconfident and wrong. For instance, consider classifier A and B, where A is very accurate, while B is less accurate than A (e.g., A's accuracy is 90% and B's accuracy is 80% on a 10 samples set, we assume 10 equally sized confidence bins). It might be that A exhibits also lower calibration error than B. Specifically, A might produce only impulsive predictive distributions, and thus be maximally overconfident when wrong.[5] However, since it would be wrong on a smaller set of samples, its calibration error could be low (in our example, ECE would be 0.1). This phenomenon can be further exaggerated by the well known biases intrinsic to the binning mechanisms involved in computing these metrics [Roelofs et al., 2020]. On the other hand, B could be less overconfident when wrong, but since it is wrong on more samples than A, it will result more miscalibrated (e.g., suppose in our case that the model is impulsive on correct samples, but produces confidences of 0.8 on the two wrong samples; despite this behaviour is more

[5] In addition, in this case no confidence based detector could distinguish correct from incorrect samples.

	Clean Data ImageNet-1K (Test)		
	Acc (↑)	**ECE** (↓)	**AdaECE** (↓)
BiT-R50x1	74.03	3.49	3.45
BiT-R50x3	77.92	6.56	6.51
BiT-R101x1	75.85	5.10	5.10
BiT-R101x3	78.20	7.63	7.63
BiT-R152x2	78.00	6.37	6.37
BiT-R152x4	78.16	9.38	9.38
ConvNeXt-B	85.53	2.87	2.82
ConvNeXt-L	86.29	2.27	2.34
ConvNeXt-XL	**86.58**	2.38	2.29
ViT-B/16	78.01	**1.40**	**1.41**
ViT-L/16	84.38	1.81	1.83
SWIN-B	84.71	8.40	8.40
SWIN-L	85.83	5.50	5.50

Table 3.3.3: In-distribution accuracy and **calibration** for ImageNet-1K.

desirable the ECE would be 0.16). [6]. For this reason, differently from what is typical in literature, we do not draw conclusions about the reliability of a model based solely on the calibration error. We complement the calibration analysis with a misclassification detection evaluation, that inspects the problem of overconfidence on wrong samples at a finer granularity.

Comparing Transformers and CNNs On in-domain data (Table 3.3.3), ViTs produce the lowest calibration error and Swin transformers are outperformed by ConvNeXts. On covariate-shifted inputs (Table 3.3.4), ViTs produce higher calibration error than ConvNeXts and Swin transformers, and the model producing the lowest calibration error is the Swin-L. Consistently with [Minderer et al., 2021], within a family of models, the ECE typically decreases as the number of parameters (and also the accuracy) increases.

Conclusions:

- There is no one model that performs the best in all the covariate shift experiments in terms of calibration. Both Trasnformers or CNNs can be better or worse depending on the experiment.
- The best performing model in terms of accuracy is not the most calibrated one.

3.3.4 Misclassified Input Detection

One of the tasks a reliable classifier should be good at is to reject samples on which it is likely to be wrong. This important task is completely neglected in the comparisons performed by the

[6] In this case, a confidence based detector could perfectly distinguish correct from incorrect samples

	Domain-Shift											
	ImageNet-R			ImageNet-A			ImageNet-V2			ImageNet-SK		
	Acc (↑)	**ECE** (↓)	**AdaECE** (↓)	**Acc** (↑)	**ECE** (↓)	**AdaECE** (↓)	**Acc** (↑)	**ECE** (↓)	**AdaECE** (↓)	**Acc** (↑)	**ECE** (↓)	**AdaECE** (↓)
BiT-R50x1	39.87	15.50	15.50	10.97	42.94	42.94	62.70	8.49	8.45	27.34	24.87	24.87
BiT-R50x3	46.39	14.65	14.65	24.08	34.48	34.48	66.36	13.13	13.13	33.47	28.55	28.55
BiT-R101x1	41.72	12.24	12.24	16.29	36.68	36.68	64.61	10.21	10.21	28.69	24.37	24.37
BiT-R101x3	47.00	15.80	15.80	27.11	32.92	32.92	66.44	14.44	14.39	34.15	30.67	30.67
BiT-R152x2	48.02	15.38	15.38	27.15	32.25	32.25	66.76	12.14	12.13	35.70	28.41	28.41
BiT-R152x4	47.57	15.32	15.32	30.84	29.93	29.93	67.12	15.75	15.67	35.08	31.45	31.45
ConvNeXt-B	62.46	2.57	2.51	52.63	8.28	8.31	75.43	**2.91**	**2.78**	48.64	8.85	8.84
ConvNeXt-L	64.57	3.00	3.08	58.23	7.57	7.26	76.77	3.72	3.85	50.08	10.31	10.31
ConvNeXt-XL	**66.01**	2.92	2.90	**61.11**	7.54	7.21	**77.20**	4.00	4.24	**52.67**	11.16	11.15
ViT-B16	43.15	5.21	5.21	24.17	22.89	22.89	66.25	4.71	4.68	18.18	13.02	13.02
ViT-L16	61.54	3.07	3.07	47.08	11.99	11.99	74.28	5.34	5.22	45.96	10.67	10.67
Swin-B	59.63	2.18	2.17	49.72	8.77	8.76	74.74	4.92	4.81	45.07	**7.75**	**7.75**
Swin-L	64.24	**2.14**	**2.11**	59.52	**6.19**	**6.33**	76.65	3.03	3.14	48.87	8.72	8.71

Table 3.3.4: Domain-shift accuracy and **calibration** for ImageNet-1K.

	Clean Data	**Domain-Shift**			
	ImageNet-1K (Test)	**ImageNet-A**	**ImageNet-R**	**ImageNet-SK**	**ImageNet-V2**
		PRR (↑)			
BiT-R50x1	68.38	54.90	-25.60	58.70	63.13
BiT-R50x3	67.61	28.58	-42.72	60.25	64.09
BiT-R101x1	69.82	-0.42	-25.94	60.08	64.60
BiT-R101x3	68.93	29.50	-34.52	60.13	65.00
BiT-R152x2	68.03	31.56	-35.12	59.26	63.26
BiT-R152x4	67.00	**92.04**	-46.05	59.34	61.48
ConvNeXt-B	73.43	16.03	-39.91	67.44	69.84
ConvNeXt-L	73.48	40.56	-23.60	69.03	69.50
ConvNeXt-XL	74.37	35.96	-19.32	69.29	70.07
ViT-B16	74.17	11.54	-46.01	63.94	70.51
ViT-L16	**76.03**	-10.67	-34.12	**69.79**	**72.37**
Swin-B	72.04	32.65	-32.95	64.23	67.35
Swin-L	72.89	56.54	**36.53**	63.52	68.49

Table 3.3.5: Misclassification detection results using the PRR metric.

existing literature. Several ways to evaluate a model at this task are available (e.g. metrics based on ROC [Landgrebe et al., 2006] or Rejection-Accuracy curves [Fumera and Roli, 2002, Hendrycks et al., 2019b]), however it has already been observed in literature that these metrics favour models that have higher test accuracy [Condessa et al., 2015, Malinin et al., 2019] and therefore should be avoided. A metric that allows to fairly compare models with different test accuracy is the Prediction Rejection Ratio [Malinin et al., 2019]. The PRR (Equation 16 of [Malinin et al., 2019], and discussed in Section 3.4) ranges from -1 to 1. It is 0 if the rejection choice is performed at random, it is negative if the network is more confident on misclassified samples than on correctly classified samples, it is positive viceversa. The optimal value is 1, when, rejecting the most uncertain samples, the classifier only rejects misclassified samples.

Comparing Transformers and CNNs As it can be seen form Table 3.3.5, in in-distribution ViT-L/16 is the best model, immediately followed by ConvNeXt-XL. ViT-B/16 slightly outperforms ConvNeXt-B and L, which in turn outperform Swin-B and L. On ImageNet-A, the best model is BiT-R152x4, with a significant margin with respect to any other model. The second best model is Swin-L, and the third best is BiT-R50x1. On ImageNet-R, the only model with positive PRR is Swin-L, and the models with highest negative PRR are ConvNeXt-XL and L, followed by BiT-R50x1 and 101x1. On ImageNet-Sketches ViT-L/16 and ConvNeXt-XL perform comparably, immediately followed by ConvNeXt-L and B. On ImageNetV2, ViT-L/16 is the best model, immediately followed by ViT-B/16 and all the ConvNeXts.

Conclusions:

- No single model is the winner in detecting misclassified samples.
- The fact that almost all models are overconfident and wrong on ImageNet-R while showing low calibration error indicates that calibration analysis should be complemented with the misclassification detection analyses.

3.4 Understanding how a negative PRR complements calibration measures

The PRR has been introduced in Appendix B of [Malinin et al., 2019], with the purpose of not favouring models with higher base accuracy using the AUROC or the AUPR in the task of misclassification detection. For this purpose, the PRR is computed as the ratio of two quantities, one referring to an oracle rejection mechanism (knowing which are the samples on which the model is wrong) and a rejection mechanism that rejects the most uncertain samples.

Specifically, for each evaluated model one can define an oracle that discards all and only the samples on which the model is wrong. The classification error on the non-rejected samples is plotted against the percentage of examples rejected (see [Malinin et al., 2019] for a visualization). The area between the random rejection mechanism (that is a line interpolating between the classification error of the model when no sample is rejected and when all samples are rejected and the classification error is therefore zero) and the curve produced by the oracle's rejection mechanism is called AR_o. Similarly, the classification error versus percentage of examples rejected curve is plotted for a mechanism that rejects samples by decreasing levels of uncertainty (i.e. reject first the most uncertain samples). Call the area between the random rejection line and this curve the AR_m. The PRR is defined as AR_m/AR_o.

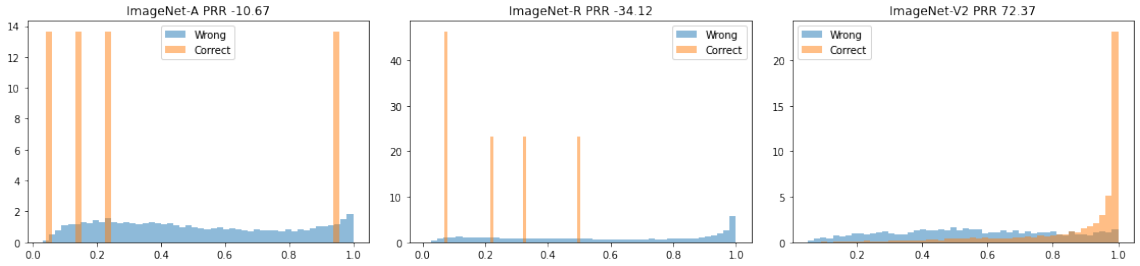


Figure 3.4.1: From left to right: the distribution of the confidence values for wrong and right samples for ViT-L/16 on ImageNet-A, ImageNet-R and ImageNet-V2. As it can be seen, in the first two cases PRR < 0 and in several cases wrong samples re given higher confidence than correct samples. In the case of PRR > 0 some wrong samples are given higher confidence of some correct samples, but to a less extent.

Therefore:

- The PRR normalizes the rejection performance of a model with respect to the best attainable performance by the same model if it knew which samples to reject, therefore not favouring models with higher accuracy.
- If PRR = 1, the uncertainty-based rejection mechanism performs as well as the oracle. If PRR = 0, it performs as the random rejection mechanism.
- By convention, areas below the random rejection line are taken with a positive sign, areas above are taken with a negative sign.
- The AR_m can be negative if the classification error raises above the random rejection line, which can happen if the rejection mechanism discards several correct samples because they have higher uncertainty than incorrect samples.
- Trivially, the AR_o is always non-negative.
- As a consequence of the previous two points, if the PRR is negative, several samples on which the classifier is correct are given lower uncertainty than some samples on which the classifier is incorrect. To confirm this, we show some histograms of a few cases in which PRR < 0 (specifically, ViT-L/16 on ImageNet-A and ImageNet-R) and in which PRR > 0 (specifically, ViT-L/16 on ImageNet-V2 in Figure 3.4.1. Remarkably, the PRR is negative only if the model attributes lower confidence to correct samples with respect to wrong samples in a large number of cases.

3.5 Conclusion

We performed an extensive analysis considering current state-of-the-art Transformers and CNNs. With simple experiments, we have shown that Transformers, just like CNNs, are vulnerable to picking spurious or simple discriminative features in the training set instead of focusing on robust features that generalise under covariate shift conditions. Therefore the presence of the self-attention mechanism might not be facilitating learning more complex and robust features. To show it is not even necessary, we observe that ConvNeXt models exhibit even superior robustness with respect to current Transformers without leveraging the self-attention mechanism in a few cases. We also conduct an in-detail analysis about the out-of-distribution, calibration and misclassification detection properties of these models.

4

Not Just Pretty Pictures: Toward Interventional Data Augmentation Using Text-to-Image Generators

Contents

Abstract

Neural image classifiers are known to undergo severe performance degradation when exposed to inputs that are sampled from environmental conditions that differ from their training data. Given the recent progress in Text-to-Image (T2I) generation, a natural question is how modern T2I generators can be used to simulate arbitrary interventions over such environmental factors in order to augment training data and improve the robustness of downstream classifiers. We experiment across a diverse collection of benchmarks in single domain generalization (SDG) and reducing reliance on spurious features (RRSF), ablating across key dimensions of T2I generation, including interventional prompting strategies, conditioning mechanisms, and post-hoc filtering, showing that modern T2I generators like Stable Diffusion can indeed be used to implement a powerful interventional data augmentation (IDA) mechanism, outperforming previously state-of-the-art data augmentation techniques regardless of how each dimension is configured.

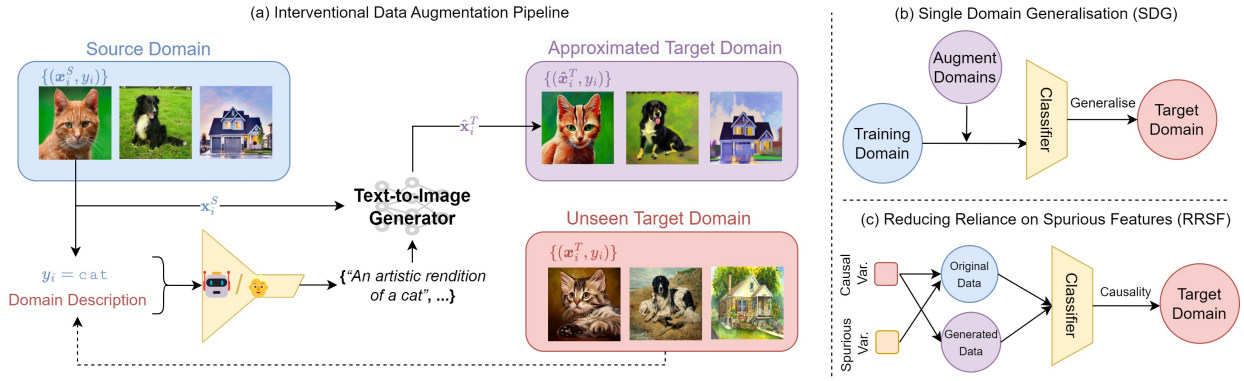


Figure 4.1.1: Using Text-to-Image Generators for Interventional Data Augmentation. In **(a)**, given an interventional prompt written by a user or LLM (and optionally, an image to edit), Text-to-Image generators simulate the described intervention by synthesising a new image or edit an existing one to match the prompt. Here, the generator edits the input image to resemble the target domain. The resulting manipulated images can be used to train more robust and generalizable models. In **(b)** (Single Domain Generalization), synthetic data are generated to mimic potential target domains and combined with data from a given source domain to train a downstream classifier. In **(c)** (Reducing Reliance on Spurious Features), synthetic data are generated to break the spurious correlation in a biased dataset and used to train a downstream classifier.

4.1 Introduction

The success of deep image classifiers is largely built on the assumption that the train and test data come from the same domain – i.e., that they are independent and identically distributed (i.i.d.) – but in real-world applications, small changes in the environmental conditions under which the image is captured can break this assumption, significantly degrading their performance [Gulrajani and Lopez-Paz, 2020, Wang et al., 2022a, Sakaridis et al., 2020]. Since these changes only affect the inputs (i.e., the covariates) in some features without altering the labels, this form of distribution shift is also known as covariate shift [Quionero-Candela et al., 2009]. In the absence of more sophisticated techniques to simulate the possibility of sampling data coming from different environmental conditions, the employment of complex augmentation pipelines integrating image transformation primitives has been one of the most effective techniques for this purpose.

Many types of augmentation primitives can be thought as reproducing (often approximately) a controlled and targeted manipulation of the domain-specific environmental conditions in which the image was captured (e.g., illumination or weather conditions) without affecting the label-related features. As such, augmentations may be understood as an automated, low-cost way of simulating interventions over the environmental factors that are likely to change across domains, turning *observational data* (i.e., with no intentional manipulation of the environment) into approximated *interventional data* [Ilse et al., 2021, Wang et al., 2022b]. Motivated by this principle, several works

have theoretically conjectured the utility of an augmentation mechanism capable of simulating arbitrary interventions [Ilse et al., 2021, Wang et al., 2022c, Wang and Veitch, 2022, Gowda et al., 2021]. However, since it is not possible to target arbitrary interventions in the context of traditional augmentation pipelines (e.g., it is not possible to hard-code a pixel-space intervention to transform images of paintings into realistic photos), prior work has instead focused on leveraging prior knowledge about specific invariances expected to hold in the target domains [Hong et al., 2021, Li et al., 2020, Ilse et al., 2021] or targeting specific downstream applications [Ouyang et al., 2022, Gowda et al., 2021].

Recently, powerful Text-to-Image (T2I) generative models like Stable Diffusion [Rombach et al., 2021] have emerged that can be used to synthesize new images (or edit existing ones) using text prompts describing the desired output image (see Figure 4.1.1). In this work, our goal is to study how well such models can serve as general-purpose interventional data augmentation (IDA) mechanisms by simulating arbitrary interventions, either by editing existing images or synthesising new ones using interventional prompts, allowing one to effectively sample from the approximated interventional distribution and augment existing training datasets with the resulting generated images. Unlike previous approaches, these models can be used off-the-shelf without requiring manual hard-coding of individual interventions or training on application-specific data: instead, it is only necessary to describe the desired intervention via language (e.g., simulating interventions over lighting conditions by editing images with prompts like "a photo taken at night" or "it is a cloudy day"). Several recent works have studied the usefulness of synthetic data from T2I generators (see, e.g., Bansal and Grover, 2023, Azizi et al., 2023, He et al., 2023, Trabucco et al., 2023); but so far, their capacity to augment existing datasets by simulating interventions has only been studied in limited contexts (see Section 4.2).

In this work, we systematically analyse the extent to which modern T2I generators can perform general-purpose IDA. We perform extensive experiments across several benchmarks for two key tasks in which the environmental and causal variables can be disentangled and the utility of synthetic interventional data can be precisely measured: **(1)** Single Domain Generalization (SDG) and **(2)** Reducing Reliance on Spurious Features (RRSF). Our investigation spans several key aspects of T2I synthesis and editing, including the use of different interventional prompting strategies, conditioning mechanisms, and post-hoc filtering techniques. Our findings reveal that T2I generators substantially outperform existing state-of-the-art image augmentation methods, regardless of how we configure each of these aspects. Our primary findings and contributions are as follows:

1. We show that T2I-based IDA surpasses previous state-of-the-art augmentation techniques in simulating interventions across a broad range of SDG and RRSF benchmarks representing

widely-varying environmental conditions and complexity.

2. We find that the choice of conditioning mechanism has the greatest impact on performance across tasks, followed by the choice of prompting strategy. However, in contrast to prior works, we find that post-hoc filtering is not consistently beneficial.
3. We show that retrieving images directly from the training dataset of the T2I generator can also yield competitive performance in several cases, and explore the comparative strengths and weaknesses of retrieval versus generation.

4.2 Problem Setting and Related Works

The Problem of Out-of-Domain Generalization Given a data distribution $P(\mathbf{x}, y) = P(y|\mathbf{x})P(\mathbf{x})$ where $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$, $y \in \mathcal{K} = \{1, 2, \ldots, |\mathcal{K}|\}$, learning a classifier amounts to estimating $\hat{f}(\mathbf{x}) \approx P(y|\mathbf{x})$ (i.e., predicting the conditional distribution of the label y given a covariate $\mathbf{x}$) using a labelled training set $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$. Given the finite amount of data available in $\mathcal{D}_{\text{train}}$ and the high dimensionality of $\mathcal{X}$, the samples in $\mathcal{D}_{\text{train}}$ are not representative of the whole input space (i.e. $\mathbf{x}_i \in \mathcal{X}_{\text{train}} \subset \mathcal{X}$). When deployed in the wild, the classifier will likely be exposed to inputs sampled from regions of the input space not represented in the training set, even when $\mathcal{K}$ is the same. Specifically, we are in presence of *covariate shift*, a form of distribution shift. It has been empirically observed that neural classifiers' performance significantly degrade in the presence of covariate-shifted evaluation data [Gulrajani and Lopez-Paz, 2020, Wang et al., 2022a, Sakaridis et al., 2020, Pinto et al., 2022b].

Several theoretical frameworks have been developed to make the problem of out-of-domain (OOD) generalization well-posed [Wang et al., 2022b, Wang and Veitch, 2022, Ilse et al., 2021, Quionero-Candela et al., 2009] – in this work, we default to the framework proposed by [Ilse et al., 2021]. In computer vision, the core principle is that pixel values of image $\mathbf{x}_i \in \mathcal{X}$ are the result of a data generation process that combines (unobserved) features h_{y_i} and $h_{\mathbf{c}_i}$ generated by the label y_i, and conditions described by a vector of *environmental variables* $\mathbf{c}_i \in C$ [Gowda et al., 2021, Ilse et al., 2021]. To make the problem more tractable, it is often assumed it is possible to partition C into M domains (i.e., $C = \bigcup_{j=1}^M C^j, C^k \cap C^h = \varnothing, \forall k \neq h$) based on how similarly the environmental conditions impact $\mathbf{x}$, so that the contextual variables values and impact are summarised in the discrete indices j [Arjovsky et al., 2019]. For instance, environmental variables could be aggregated to represent similar illumination conditions or backgrounds. Furthermore, an unobserved spurious confounder $\mathbf{s}$ might correlate both y_i and $\mathbf{c}_i$. A high-performing classifier is likely to learn these spurious correlations, as they are predictive of the label y_i; but such correlations

will (by definition) not hold under all environmental conditions, damaging classifiers' ability to generalize [Xiao et al., 2021, Geirhos et al., 2019].

Simulating Interventional Data for Out-of-Domain Generalization Prior works [Ilse et al., 2021, Wang et al., 2022b] have proposed that such a problem is solvable by performing interventions on $\mathbf{c}_i$ (i.e., manipulating $\mathbf{c}_i$ to break such spurious correlations without changing y_i). However, direct collection of interventional data is usually quite difficult (e.g., collecting datasets portraying objects of the same class in all environments of interest may be highly impractical). Identifying heuristic methods to disentangle causal from environmental factors (usually by augmenting original images from the source domain) has been a key component of many leading approaches to domain generalization. For example, CIRL [Lv et al., 2022] and ACVC [Cugu et al., 2022] manipulate the amplitude component of the image frequency spectrum of the Fourier transform (which is presumed to approximately encode environmental information), while others have used style transfer techniques to perturb environmental factors while preserving image content [Hong et al., 2021, Li et al., 2020, Jackson et al., 2019], or trained Cycle-GAN to preserve the causal factors in a cyclic transformation between domains with different styles [Wang et al., 2022b]. Beyond methods explicitly attempting to disentangle these two components, [Ilse et al., 2021, Gowda et al., 2021] understand augmentations as simulating alterations of $\mathbf{c}_i$ without affecting y_i – for example, rotations encode the belief that change of viewpoints should preserve the class label. Importantly, these assumptions might not hold in all applications (e.g., in digit classification, rotations of more than 90 degrees can swap the ground-truth labels of 6 and 9), so not all augmentations are valid for any given application. Domain-agnostic data augmentation pipelines (such as those proposed by Hendrycks et al. 2020, Cubuk et al. 2020, DeVries and Taylor 2017, Hendrycks et al. 2022, Cugu et al. 2022, Pinto et al. 2022c) can be understood as hard-coding interventions over various features that are expected to vary across novel environments; but such assumptions may not hold across all possible domains. For this reason, [Ilse et al., 2021] suggests a mechanism to select parametric hand-crafted augmentations that have a greater impact on environmental factors than causal factors.

Text-to-Image Generators With the recent rise of powerful flexible T2I generative models (e.g., Nichol et al., 2021, Rombach et al., 2021, Ramesh et al., 2021), a natural question is whether these models, which are capable of synthesising images using natural-language prompts, could be used to effectively implement IDA. That is, while some hand-crafted parametric augmentations can be straightforwardly implemented by a programmer to manipulate the image directly in pixel space (e.g., lens distortion, chromatic aberration, vignetting, etc.), such methods many only be able to approximate many augmentations (often with much greater difficulty of implementation; e.g.,

introducing realistic rain or snow) or may not be able to approximate them at all (e.g., turning a cartoon into a photo, changing the background of a scene, or modifying the material of an object). On the other hand, modern T2I generators that have been training on large amounts of weakly supervised data can be used zero-shot to directly approximate such augmentations (either from existing images or from scratch) using natural language, and have been observed to produce high quality samples [Meng et al., 2021]. Using such models for IDA would be extremely convenient, as these editing and synthesis abilities can be made available "off-the-shelf" without requiring any task-specific fine-tuning.

Data Augmentation with T2I Generators Contemporary work has investigated how T2I generators can be used to synthesize large-scale pre-training data [He et al., 2023, Sariyildiz et al., 2023, Azizi et al., 2023], compensate for the lack of training data in data-scarce environments [He et al., 2023, Trabucco et al., 2023], and diagnose classifiers' lack of robustness to covariate shift [Vendrow et al., 2023]. A related branch of research leverages synthetic data from T2I generators for test-time adaptation, transferring OOD samples to an approximation of the training domain as a form of test-time adaptation [Yu et al., 2023, Gao et al., 2022]. Closest to our work, [Bansal and Grover, 2023] show it is possible to use Stable Diffusion to generate synthetic data that improves the robustness of classifiers trained on ImageNet-1K [Deng et al., 2009b] for multiple forms of distribution shift using an ensemble of generative prompts. In this work, we focus on the utility of T2I-generated synthetic data for training downstream classifiers, but depart from standard ImageNet analyses in order to develop a deeper understanding of how T2I generators can be used for IDA by focusing on SDG and RRSF, allowing us to directly measure the effectiveness of T2I-simulated interventions in these settings across variable conditioning, prompting, and filtering techniques.

4.3 Simulating Interventions with Text-to-Image Editing

4.3.1 Experimental Setting

Given some source training domain $\mathcal{D}^S = \mathcal{X}^S \times \mathcal{K}$ and some target domain $\mathcal{D}^T = \mathcal{X}^T \times \mathcal{K}$, our goal is to assess how well T2I generators can approximately modify environmental features $\mathbf{c}_i$ to simulate $\mathcal{D}^T$ while keeping causal features for y_i constant. As the most common approach to image augmentation (including all baselines we consider) involves editing pre-existing training images and adding them to the training dataset rather than synthesising new training data from scratch [Shorten and Khoshgoftaar, 2019], we begin our analysis by studying the analogous setting of T2I-enabled image editing using `SDEdit` [Meng et al., 2021]. For an image $\mathbf{x}_i^S \in \mathcal{X}^S$ of class y_i, we

aim to transform it into $\hat{\mathbf{x}}_i^T$ such that it retains y_i but appears to have been sampled from $\mathcal{X}^T$. Each of the editing techniques we consider are conditioned on a natural language prompt, denoted $\mathbf{z}_i^T$. For instance, if $\mathbf{x}_i^S$ represents a cartoon cat and $T = \texttt{painting}$, $\mathbf{z}_i^T$ could be "a painting of a cat". Using the generator G, we transform $\mathbf{x}_i^S$ into $\hat{\mathbf{x}}_i^T = G(\mathbf{x}_i^S, \mathbf{z}_i^T)$. For all experiments, we use Stable Diffusion v1.5 [Rombach et al., 2021] pre-trained on LAION-Aesthetics.[1] Successful transformations produce $\hat{\mathbf{x}}_i^T \in \hat{\mathcal{X}}^T$ with $\hat{\mathcal{X}}^T \approx \mathcal{X}^T$. The synthetic pairs $\hat{\mathcal{D}}^T = (\hat{\mathbf{x}}_i^T, y_i)$ are then combined with $\mathcal{D}^S$ to perform ERM via cross-entropy minimisation to train the neural classifier (ResNet-18 and ResNet-50).

In this work, we focus on *Single-Domain Generalization* (SDG) and *Reducing Reliance on Spurious Features* (RRSF) as representative settings where access to a high-quality approximations of the intervened distributions can measurably affect the performance of classifiers when training on both $\mathcal{D}^S$ and $\hat{\mathcal{D}}^T$. We describe our experimental formulation of both problems below.

Single-Domain Generalization (SDG) Given data $\mathcal{D}^S$ from a source domain accessible at training time, the goal of SDG is to achieve high performance on a set of datasets $\mathcal{D}^{T_j}$ with $j = 1, 2, ..., J$ sampled from different target domains [Qiao et al., 2020]. In this setting, the generator uses $\mathcal{D}^S$ and $\mathbf{z}_i^{T_j}$ to generate $\hat{\mathcal{D}}^{T_j} \approx \mathcal{D}^{T_j}$. Following the standard evaluation procedure, we train a classifier on a single domain of each benchmark ($\mathcal{D}^S$) and test it on the others ($\mathcal{D}^{T_j}$), and report the average accuracy over the J target domains. For our experiments, we consider four widely used benchmarks that vary for type of domain shift, number of classes and training samples: (1) **PACS** [Li et al., 2017], containing the domains `art painting`, `cartoon`, `sketch`, and `photo`; (2) **Office-Home** [Venkateswara et al., 2017], containing `Art`, `Clipart`, `Product`, and `Real World`; and (3) **NICO++** [Zhang et al., 2022], containing `autumn`, `dim`, `outdoor`, `grass`, `water`, and `rock`; and (4) **DomainNet** [Peng et al., 2019], containing `clipart`, `infograph`, `painting`, `quickdraw`, `real`, and `sketch`. We provide a more detailed description of the training procedure and other aspects of the SDG experiments in Section C.1.

Reducing Reliance on Spurious Features (RRSF) Sometimes the training data is collected from a domain $\mathcal{D}^S$ in which spurious features correlate with the labels. If a classifier relies on such spurious features, it will be unable to generalize to unseen test domains in which the spurious feature is no longer predictive of the label [Xiao et al., 2021, Geirhos et al., 2019]. In this setting, the prompts $\mathbf{z}_i^T$ intentionally perturb the spurious features to simulate domains in which the spurious correlation is broken. We consider three standard benchmarks: (1) **ImageNet-9** [Xiao et al., 2020b] measures the over-reliance on background to predict the foreground (Background Bias), (2) **Cue-Conflict Stimuli (CCS)** [Geirhos et al., 2018b] assesses the over-reliance on texture (Texture Bias), and (3) a subset of

[1] LAION-Aesthetics is a subset of the LAION-5B dataset [Schuhmann et al., 2022a] consisting of web-scraped text-image pairs with high "aesthetic scores" [Schuhmann et al., 2022b].

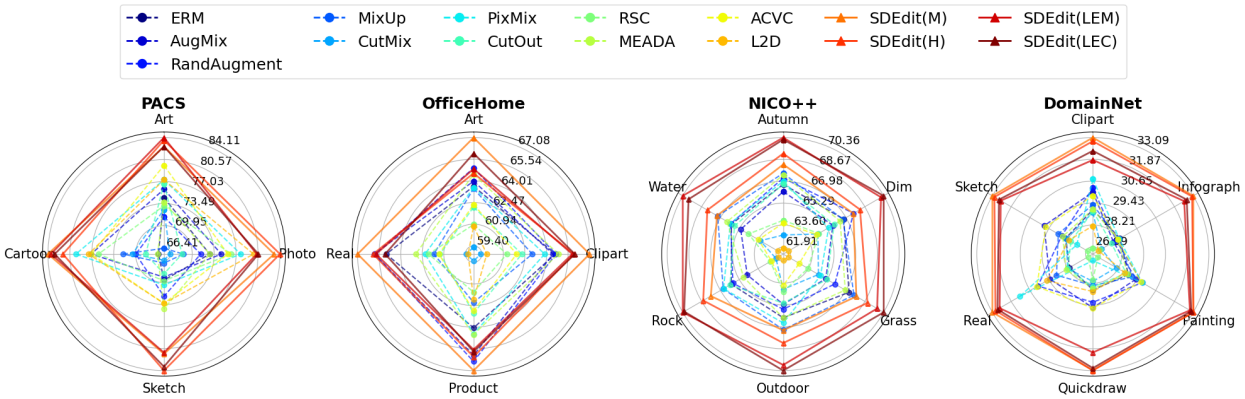


Figure 4.3.1: Single Domain Generalization (SDG) Results. Average SDG test accuracies on the remaining target domains when training ResNet-50 on each source domain (indicated on each axis) using the respective data augmentation methods. Baseline methods are visualized with dashed lines, and `SDEdit` methods with solid lines.

CelebA [Xiao et al., 2020b] evaluates over-reliance on spurious demographic features (Demographic Bias – in this case, the spurious correlation between hair colour and gender in CelebA). See Section C.2 for further details about each dataset and associated indices measuring each form of bias.

4.3.2 Results

To evaluate the effectiveness of edited data from T2I models, we compare their results with key augmentation baselines broadly representing different approaches in the domain generalization literature (in addition to ERM): (1) **AugMix** [Hendrycks et al., 2020], (2) **RandAugment** [Cubuk et al., 2020], (3) **CutOut** [DeVries and Taylor, 2017], and (4) **PixMix** [Hendrycks et al., 2022], which all combine parametric transformations in complex pipelines to enhance model robustness. We also evaluate (5) **ACVC** [Cugu et al., 2022], which combines parametric transformations and augmentations in the Fourier domain for style mixing; interpolation-based methods like (6) **MixUp** [Zhang et al., 2017] and (7) **CutMix** [Yun et al., 2019]; methods that train generators to diversify training data (8) **L2D** [Wang et al., 2021b]; and adversarial data augmentation techniques like (9) **MEADA** [Zhao et al., 2020] and (10) **RSC** [Huang et al., 2020].

Single Domain Generalization Considering the fact that Stable Diffusion is built on top of a text encoder that, like many LLMs, can be sensitive to small differences in prompts that are not generally meaningful to humans (see, e.g., Ribeiro et al., 2020, Wang et al., 2021a, Moradi and Samwald, 2021), we experiment with four distinct prompting strategies using `SDEdit` to measure its sensitivity to variation in prompts: (1) ***Minimal (M)***, sentences including only the domain label, class label,

and function words (articles or prepositions) as necessary to make the prompt grammatically correct, like "a `domain` of a `class`" (e.g., "a sketch of an elephant"); (2) ***Domain expert (H)***, a collection of "handcrafted" prompts authored by a human given only metadata descriptions provided by the respective benchmarks, without looking at any samples form the target domain; and (3 & 4) ***Language enhancement (LE)***, a collection of prompts generated by T5 [Raffel et al., 2020], in two variants: one that deterministically selects the highest-probability interventional prompts (LE_C), the other that favors diversity in prompting (LE_M).[2] (See Section C.3 for further details on each prompting strategy.) As shown in Figure 4.3.1, `SDEdit` outperforms all baselines regardless of the source domain (when averaging over target domains). Specifically, across all the considered benchmarks, using ResNet-50 with minimal prompt yields a 5% improvement over the strongest baesline, PixMix, which in turn outperforms ERM by just 1.10%. We find that, when considering the performance on individual benchmarks, no single baseline consistently outperforms the others. This reinforces the observation that each of these techniques encodes different assumptions about the types of invariances expected to hold in the test domain. For the largest-scale dataset we consider, `DomainNet`, traditional data augmentation methods fail to demonstrate a substantial performance boost compared to ERM; but `SDEdit` is able to deliver a strong average performance boost of 5.48%. Comparing across all SDG datasets, `SDEdit` is the only method that consistently outperforms ERM across all benchmarks. (For a more detailed breakdown of all results figures, see Section C.1.)

We also find that the most sophisticated prompting strategy does not usually perform best: in `PACS`, `OfficeHome` and `DomainNet`, the Minimal (M) and Handcrafted (H) strategies outperform LE_M and LE_C, indicating that including additional details (e.g., specifying various styles of paintings across multiple prompts) does not yield obvious benefits, and may even degrade performance (e.g., by "injecting noise" into the pipeline). However, in `NICO++`, LE_M and LE_C show superior performance to (M) and (H), which may be explained by the fact the domain labels for NICO are not detailed enough for minimal prompts to be fully descriptive, meaning that the additional details included in prompts can be more beneficial in such contexts.

Precisely Describing the Target Domain Is Not Necessary. As noted above, Stable Diffusion is trained on a massive pre-training corpus of weakly-supervised data scraped from the web, which means it has likely been trained on samples that resemble a number of the considered test distributions. By comparison, while the baselines we consider do make limited assumptions about the type of interventions they perform (and therefore yield better or worse performance depending

[2]Note that, by design, none of the prompting strategies are optimised to boost the reported metrics: they are generated in a way that is independent from classifiers' performance on downstream tasks or the structure of the generator. See Section C.10 for a complete list of image-generation prompts used in experiments.

Table 4.3.1: Average SDG Performance. The number reported is the average Single Domain Generalization average of all domains in each dataset, each serving as a single source domain. The best and second-best performing methods are highlighted with bold and underline, respectively.

	PACS	OfficeHome	NICO++	DomainNet	Average
ERM	61.96	61.94	69.95	25.26	54.78
MixUp	58.17	60.46	<u>70.63</u>	25.49	53.69
CutMix	58.50	57.16	67.03	24.47	51.79
AugMix	64.63	62.60	68.81	26.20	55.56
RandAugment	62.61	<u>63.02</u>	69.88	26.17	55.42
CutOut	60.87	60.03	69.23	24.90	53.76
RSC	64.58	59.10	67.37	23.32	53.59
MEADA	64.04	62.08	69.89	25.26	55.32
PixMix	67.12	61.43	69.48	25.53	<u>55.89</u>
L2D	68.89	58.37	65.19	24.75	54.30
ACVC	<u>67.98</u>	59.92	66.92	<u>26.46</u>	55.32
SDEdit(M)	76.43	**64.66**	71.12	**31.94**	61.04
SDEdit(H)	**77.87**	63.27	71.95	31.82	61.23
SDEdit(LEC)	76.38	63.43	**73.69**	31.44	**61.24**
SDEdit(LEM)	75.65	63.14	73.61	30.94	60.84

on whether those interventions correspond to the covariate shift from the source domain to test domain – see our RRSF analysis below), they do not have comparable access to approximations of the test domain. For this reason, we perform an experiment to "level the playing field" in order to better assess the usefulness of `SDEdit` as an interventional mechanism by avoiding generating data resembling the test domain. Given a single training domain from the original dataset and a chosen test domain, we use `SDEdit` to transform the training data to all domains *except the test domain* (`SDEdit`(M)×), use it for IDA training, and measure accuracy on the test domain. Fixing a test domain, we repeat this experiment for each possible choice of the training domain, and report the average accuracy on the held-out test domain. In this case, we are measuring `SDEdit`'s capacity to simulate interventions for SDG even when knowledge about the chosen test domain is not used in synthesising interventional data. We find that the generative model-based methods still substantially outperform the data augmentation baselines, with only a marginal drop

Table 4.3.2: SDG PACS result with ResNet-50. Columns are individual source domains; accuracies are the average test accuracy of the three remaining target domains when training using the indicated source domain. The lower part of the table highlights the comparison between accessing (✓) or not accessing (×) synthetic target domains.

	Art	Photo	Sketch	Cartoon	Average
ERM	74.44	48.78	50.89	73.74	61.96
MixUp	66.31	42.98	45.64	77.76	58.17
CutMix	72.53	40.03	44.72	76.72	58.50
AugMix	75.80	51.32	49.99	81.42	64.63
RandAugment	71.38	46.80	55.95	76.33	62.61
CutOut	76.67	42.69	48.93	75.2	60.87
RSC	73.15	53.47	51.11	80.58	64.58
MEADA	73.72	48.78	59.81	73.84	64.04
PixMix	77.33	55.58	52.42	83.15	67.12
L2D	77.33	58.41	58.14	81.70	68.89
ACVC	79.63	52.76	58.13	81.40	67.98
SDEdit(M) ×	81.21	57.54	80.60	84.76	76.03
SDEdit(M) ✓	82.67	62.94	73.78	86.33	76.43

in performance with respect to the case in which the target domain is approximated by Stable Diffusion (`SDEdit`(M)✓). This indicates that the interventions simulated by Stable Diffusion are useful even when knowledge about the test domain is not available.

Reducing Reliance on Spurious Features Depending on the type of spurious correlation to be addressed in each experiment, we prompt `SDEdit` in different ways: for ImageNet-9 experiments, we handcraft prompts that describe a wide variety of possible backgrounds and randomise the combination of the object classes and backgrounds; for CCS, we use prompts that induce the generator to change the texture of the objects (e.g., turning them into a sculpture of a specific material); and for Celeb-A, we randomise the correlation between gender and hair colour. Our results are displayed in Figure 4.3.2. We find that, although several techniques are often assumed to perturb spurious features in a way that is agnostic to the target domain, our experiments indicate

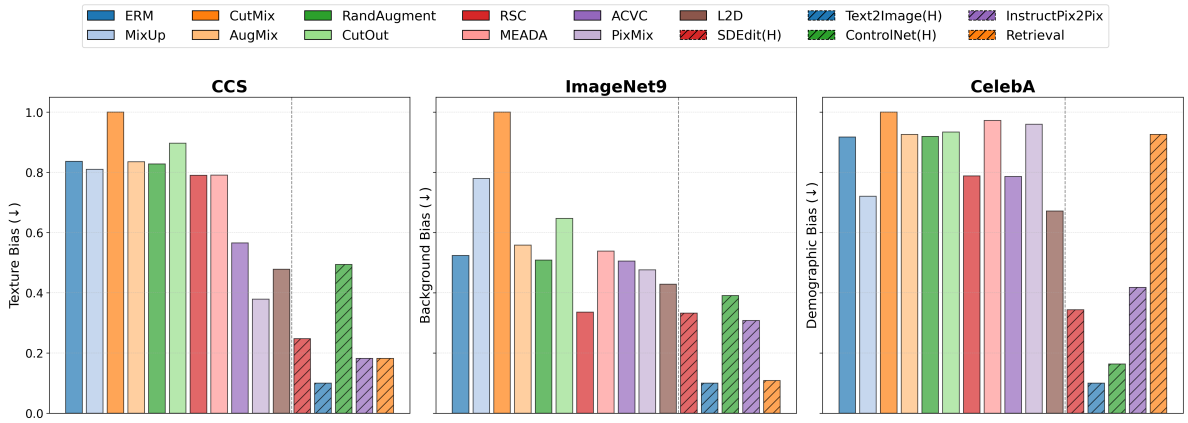


Figure 4.3.2: Performance on Breaking Spurious Correlations. Reliance on different image attributes in comparison with baselines (solid lines) and techniques using T2I models for IDA (dash lines) using ResNet-18. (Lower scores are better.) Among the latter, `Text2Image` with domain expert produced prompts performs the best. Image editing techniques (like `SDEdit, ControlNet, InstructPix2Pix`) yield mixed results depending on their ability to manipulate the targeted spurious feature. `Retrieval` performs competitively with respect to `Text2Image` on `ImageNet-9`, significantly worse on `CCS` and `CelebA`.

that this may not the case – instead, baselines are (perhaps unsurprisingly) most effective when their augmentation pipeline implicitly intervenes over the corresponding spurious dependency. For example, PixMix mixes the input images with fractals that alter their texture (and often the background), but yields a worse Demographic Bias than ERM. In contrast, `SDEdit` can perform the desired augmentation based on the relevant spurious dependency by simply describing it using interventional prompts, which enables substantial improvements over ERM in all settings. Such flexibility and ease-of-implementation with respect to interventions of interest are key advantages of using T2I models for IDA.

4.4 Alternative Approaches

In the previous section, we show that `SDEdit`, one of the simplest and most widely-adopted editing techniques, substantially outperforms traditional augmentation pipelines for SDG and RRSF. However, there are several other ways we can simulate interventions with T2I generators: by default, such models can generate images using only text, with no need to provide an input image to edit; and more sophisticated image-editing techniques have also been developed using different conditioning mechanisms. In Section 4.4.1, we investigate the use of these alternative generative approaches for the same tasks. Additionally, in Section 4.4.2, we consider [He et al., 2023]'s finding that filtering low-quality image outputs can improve synthetic data from earlier T2I generators, and

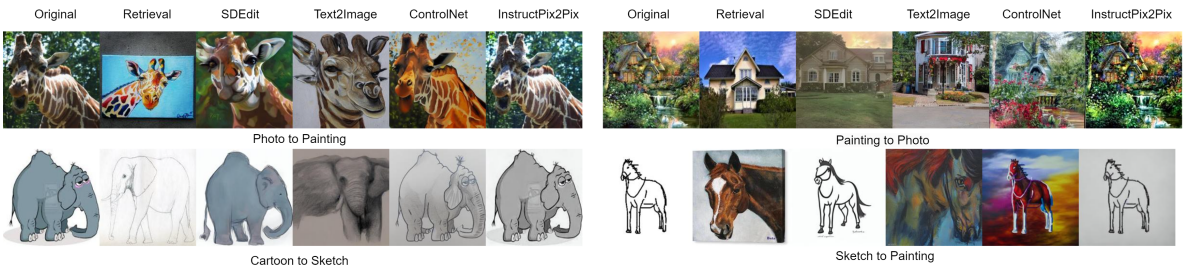


Figure 4.4.1: Visualization of selected samples from PACS. Recall that `Retrieval` and `Text2Image` do not take the `Original` image into account, but `SDEdit`, `ControlNet`, and `InstructPix2Pix` do.

study whether this is also true for SDG and RRSF using more recent generators.

4.4.1 Conditioning Mechanisms

Despite the impressive performance of `SDEdit`, we observe several cases in which such a simple editing technique is not sufficient to simulate the desired intervention (e.g., see the second row of Figure 4.4.1). This might depend on the fact `SDEdit` initializes the diffusion process from an embedding of the image being edited, which may be too constraining to obtain the desired manipulation.[3] However, several other conditioning mechanisms exist. We consider three other forms of conditioning that may be suitable for our goal: `Text2Image`, `ControlNet` and `InstructPix2Pix`. With `Text2Image` we refer to the native ability of Stable Diffusion of generating images by conditioning only on the text: the diffusion process is initialised with random noise, and the prompt embeddings are used to condition the attention matrices in the denoising steps, steering the diffusion in order to yield an output that matches the description given by the prompt. `ControlNet` [Zhang et al., 2023] induces stronger spatial consistency between the original and the augmented image by using an additional network that has been trained to condition the generative process on spatial guidance ("Canny edges"; Canny, 1986). Finally, `InstructPix2Pix` [Brooks et al., 2022] aims to improve diffusion models' ability to follow editing instructions by fine-tuning it on tuples of original images, editing instructions, and desired editing outputs.

Single Domain Generalization In Figure 4.4.2, we see that conditioning can have a large impact on performance. First, we observe that `InstructPix2Pix` underperforms with respect to other conditioning mechanisms in most cases. This may be related to the fact that its training set (which is distilled from Stable Diffusion) contains a limited variety of samples that may supply an inadequate implementation of a general interventional mechanism. Although `ControlNet` allows for a better

[3] Indeed, we observe this phenomenon persists across different hyperparameter settings controlling the strength of conditioning.

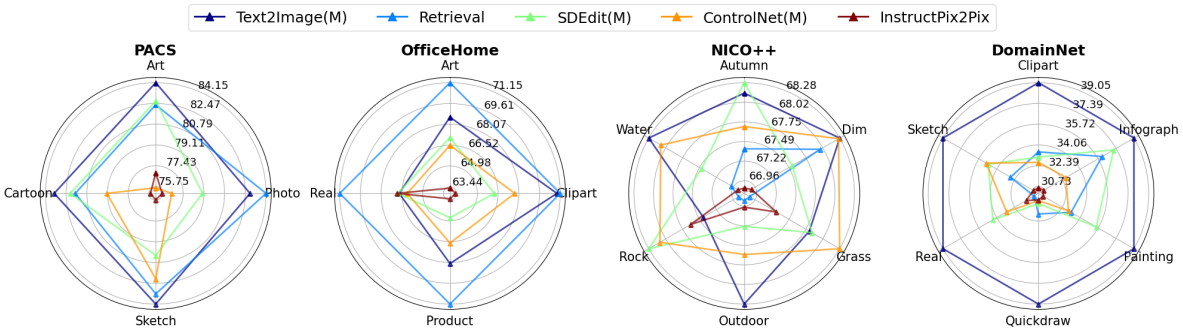


Figure 4.4.2: SDG Results by Conditioning Mechanism. Results are reported following the same format as Figure 4.3.1.

spatial control, its performance is similar to or lower than `SDEdit` in most cases. This might be expected when considering that this evaluation task does not particularly benefit from the preservation of spatial features. More surprisingly, we see that `Text2Image` can be an extremely effective conditioning technique. The success of this approach indicates that conditioning on an image may often be a hindrance in approximating the desired domain (for instance, due to the difficulty of transforming some images to extremely different domains, e.g., from sketch to photo).

Reducing Reliance on Spurious Features All conditioning techniques are useful in reducing classifier bias. In the aggregate, `Text2Image` is most effective in doing so across all benchmarks; whereas other conditioning mechanisms have varying strengths and weaknesses across the different tasks. For instance, `ControlNet`'s ability to preserve the spatial features (i.e., the edges) of an image while modifying other aspects (in this case, the hair colour) yields second-best performance in CelebA, as the Canny edge detector is designed to omit information about the texture of objects. While `InstructPix2Pix` is second-best in removing overreliance on texture and background, it is not as effective as `ControlNet` on CelebA. Finally, `SDEdit` shows middling performance across all benchmarks: it never performs best (or second-best), but it also never performs the worst.

Retrieval Is Not (Always) Enough. The strong performance we observe when removing source images from the generative process (i.e., substituting `SDEdit` for `Text2Image`) suggests that Stable Diffusion's effectiveness is higher when sampling from its approximation of the intervened distribution without starting from an input image. This raises the question of whether using Stable Diffusion to generate images is actually necessary: might we achieve similar results by simply using interventional prompts to retrieve relevant images directly from its original training dataset? To answer this question, we configure a retrieval baseline to compare the results of generating images

and retrieving images from Stable Diffusion's training set using a simple image retrieval system,[4] querying it with the same minimal prompt that is used to generate images (see Figure C.9.2). We observe large differences between the behaviors of the `Retrieval` method across the tasks we consider. In SDG, retrieval proves to be an extremely effective technique as shown in Figure 4.4.2. For example, `Retrieval` outperforms all other methods on OfficeHome; and on PACS it proves to be only marginally inferior to `Text2Image(M)`. This is likely because Stable Diffusion's training data contains ample data from the classes and domains covered by these benchmarks and it is relatively easy to retrieve this data. On the other hand, for NICO++ and DomainNet, the retrieval baseline performance is inferior to `Text2Image(M)`. However, when Reducing Reliance on Spurious Features, `Retrieval` underperforms with respect to most generative techniques. This disagreement suggests that both retrieval and generative approaches are of interest and worth pursuing for different applications and different downstream tasks, as both have their own unique advantages and disadvantages. Indeed, beyond performance figures, there are also important practical distinctions between the two. In favor of retrieval, retrieved images do not generally contain unrealistic artifacts; and once the retrieval engine has been deployed, it can be significantly faster than generation. However, such deployment requires massive storage resources ($>$ 200TB) and relies on highly efficient indexing and computing infrastructure. In contrast, generative models are significantly more compact in terms of storage (the version of Stable Diffusion we use is $\sim$ 8GB) and do not require a dedicated infrastructure to be run. Finally, we observe that modern generators can effectively produce samples that combine concepts from their training data in new ways: in Section C.9.1, we compare images generated with prompts combining such concepts against images retrieved with the same prompts.

4.4.2 Post-hoc Filtering

Although the quality of the generated samples of state-of-the-art diffusion models is impressive, failure cases may still occur and low-quality samples may be generated. Since such samples have been observed to harm the performance on downstream tasks, [He et al., 2023] and [Vendrow et al., 2023] deploy post-hoc filtering using CLIP [Radford et al., 2021] to discard them. In the case of IDA, the generated sample may fail at capturing either the specified class, the conditions of the environment we aim to simulate, or both. Therefore, we filter images that do not exhibit a high enough CLIP similarity score with respect to both prompts: one describing the class, the other describing the domain ("An image of a `class`" and "`domain`", respectively). Before training, we

[4] Accessible at `https://rom1504.github.io/clip-retrieval`.

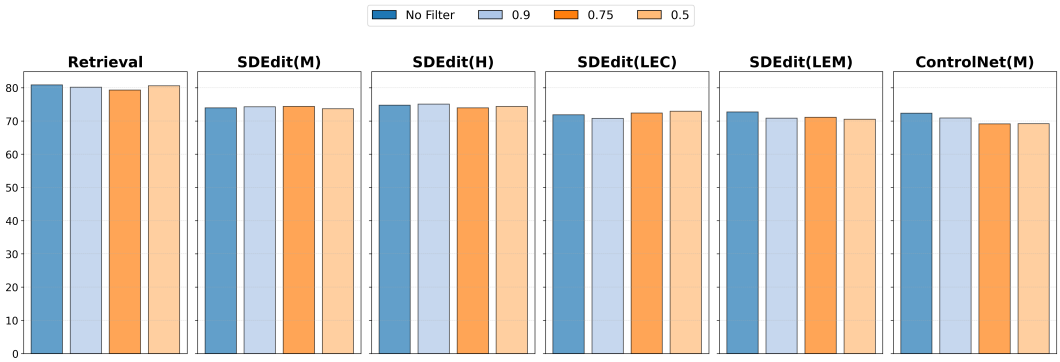


Figure 4.4.3: CLIP Filtering Results. SDG accuracies averaged across all test domains for different conditioning strategies (boxes in bold) and CLIP filtering proportions (colors).

remove the samples with scores lower than a given percentile threshold and provide our results in Figure 4.4.3. Unlike [He et al., 2023, Vendrow et al., 2023], we do not find that CLIP filtering yields consistent and substantial improvements. This may be due to the improved performance of newer generators or the fact that we are considering different tasks (SDG and RRSF). For further details, full results, and selected examples, see Section C.4.

4.4.3 Limitations and Future Works

While we observe the effectiveness of synthetic data from generative models in improving robustness across various challenging benchmarks, we still encounter several limitations. First, there are several failure modes of different conditioning mechanisms (such as `ControlNet` and `InstructPix2Pix`) and their potential sub-optimal impact on RRSF. (See Section C.9.4 for qualitative examples of failure cases where target domains are either out-of-distribution with respect to Stable Diffusion's training domain or cannot be easily described via natural-language prompts.) Additionally, the computational cost is one potential bottleneck: although the inference speed of generative models has greatly improved over time – and, if current trends continue, might be attenuated to the point of irrelevance – it remains a concern for current applications. (See Section C.7 for further discussion.)

Furthermore, although we currently focus only on image classification, we note that all the methods we explore in this work are also applicable to other computer vision tasks such as object detection, instance segmentation, and semantic segmentation. While T2I-based interventional data augmentation can theoretically be applied to these tasks, implementing it successfully presents new challenges. Specifically, our approach currently only requires conditioning on domain and class labels via natural language interventional prompts, but extending this method to object detection and

segmentation tasks would necessitate additional conditioning on pre-specified spatial information, such as bounding boxes or segmentation maps, as explored by [Wu et al., 2023, Nguyen et al., 2023].

4.5 Conclusion

In this work, we study the application of T2I generators to performing IDA in two settings, SDG and RRSF, finding they perform much better than traditional data augmentation techniques. We carry out a detailed investigation of how various components of the generative process may affect the results, concluding that the conditioning mechanism is the most important. Finally, we compare the strengths and limitations of T2I-enabled IDA with those of retrieval.

5

PILLAR: How to make semi-private learning more effective

Contents

Abstract

In Semi-Supervised Semi-Private (SP) learning, the learner has access to both public unlabelled and private labelled data. We propose PILLAR, an easy-to-implement and computationally efficient algorithm that, under mild assumptions on the data, provably achieves significantly lower private labelled sample complexity and can be efficiently run on real-world datasets. The key idea is to use public data to estimate the principal components of the pre-trained features and subsequently project the private dataset onto the top-k Principal Components. We empirically validate the effectiveness of our algorithm in a wide variety of experiments under tight privacy constraints ($\epsilon < 1$) and probe its effectiveness in low-data regimes and when the pre-training distribution significantly differs from the one on which SP learning is performed. Despite its simplicity, our algorithm exhibits significantly improved performance, in all of these settings, over all available baselines that use similar amounts of public data while often being more computationally expensive. For example, in the case of CIFAR-100 for $\epsilon = 0.1$, our algorithm improves over the most competitive baselines by a factor of at least two.

5.1 Introduction

In recent years, Machine Learning (ML) models have become an integral part of our daily lives, commonly trained on vast amounts of sensitive private data to offer services better tailored to users' needs. However, this has escalated concerns regarding user privacy. Recent studies [Shokri et al., 2017, Ye et al., 2022, Carlini et al., 2022a] demonstrate the potential for malicious queries to ML models to reveal private information. To address this problem, the de-facto standard remedy is to enforce $(\epsilon, \delta)-$Differential Privacy (DP) guarantees on the ML algorithms [Dwork et al., 2006]. Nonetheless, meeting these guarantees often compromises model utility, unless the volume of available private training data is significantly increased [Kasiviswanathan et al., 2011, Blum et al., 2005, Beimel et al., 2013a,b, Feldman and Xiao, 2014]. In the context of private learning, [Chaudhuri et al., 2011, Bassily et al., 2014a] identified theoretical lower bounds, showing a direct dependence of this cost on data dimensionality, a connection not seen in non-private learning.

To mitigate this degradation of utility, several techniques have been employed. One approach is leveraging feature extractors pre-trained on large-scale datasets (presumed public), even if their data-generating distribution diverges from the private data [Tramer and Boneh, 2021, De et al., 2022, Li et al., 2022c, Kurakin et al., 2022]. Training a linear classifier atop these pre-trained features has proven to be both cost-efficient and effective [Tramer and Boneh, 2021, De et al., 2022]. Utility gains can also be achieved by deeming part of the private data public, a scenario known as Semi-Private (SP) learning [Alon et al., 2019, Yu et al., 2021a, Li et al., 2022a, Papernot et al., 2017, 2018, Nasr et al., 2023]. Notably, utilizing public data to assist the optimizer [Li et al., 2022a, Amid et al., 2022, Nasr et al., 2023] and to reduce problem dimensionality [Yu et al., 2021a,b, Kasiviswanathan, 2021, Kairouz et al., 2020] are among the most effective strategies in this context. However, although these techniques have been shown to be effective for large ϵ values on datasets like CIFAR-10, our experiments over an extensive variety of datasets with varying amounts of training data and classes suggest that the effectiveness of some of these methods is limited in more challenging settings like low data and small ϵ.

In this work, we propose a simple SP algorithm called PILLAR and conduct an extensive empirical study over a wide range of datasets and strict privacy settings to show its effectiveness over existing methods. The key idea is to use public data to estimate the principal components of the pre-trained features and subsequently project the private dataset onto the top-k Principal Components. Despite its simplicity and use of existing techniques like dimensionality reduction [Yu et al., 2021a, Kasiviswanathan, 2021], it outperforms existing methods in these challenging settings. Beyond its empirical performance, our algorithm also enjoys a provably dimension-independent

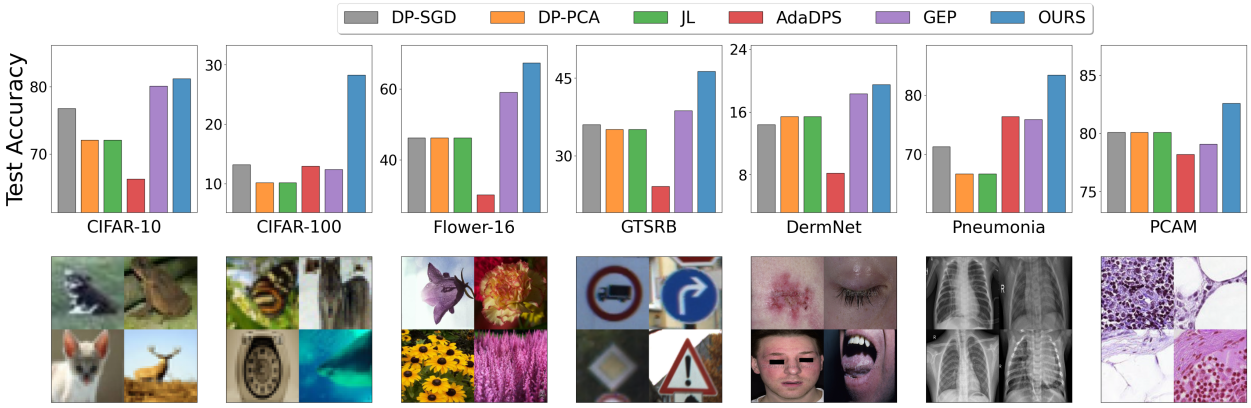


Figure 5.1.1: We compare our algorithm PILLAR with DP-SGD [Li et al., 2022b], DP-SGD with DP-PCA [Abadi et al., 2016], DP-SGD with JL transformation [Nguyen et al., 2020], AdaDPS [Li et al., 2022a], and GEP [Yu et al., 2021a] on CIFAR-10, CIFAR-100, GTSRB, Flower-16, Dermnet, Pneumonia, and PCAM for $\epsilon = 0.1$. PILLAR consistently outperforms all baselines, often with a large margin. All methods use features extracted from a ResNet-50 pre-trained on ImageNet-1K using either Supervised Learning (SL) or Self-Supervised Learning (BYOL [Grill et al., 2020])

sample complexity when learning linear halfspaces, and when the distribution satisfies a low-rank separability condition outlined in Definition 3

For practical applications like image classification, we ascertain that pre-trained representations meet this condition across a diverse range of datasets. As suggested by concurrent research [Tramèr et al., 2022], we validate our algorithm's efficacy not only against standard benchmarks in DP literature (e.g., CIFAR-10 and CIFAR-100) but also across various datasets (Figure 5.1.1) that better represent the challenges and application domains of private training. Remarkably, our experiments reveal that our algorithm surpasses several existing state-of-the-art algorithms [Nguyen et al., 2020, Yu et al., 2021a, Li et al., 2022a,b], with various levels of access to public data, across seven different datasets while remaining computationally economical.

Unlike previous works, our evaluations particularly concentrate on private data distributions (e.g. traffic signs and medical datasets in addition to object recognition) that significantly deviate from the pre-training one (ImageNet) and focus on low-data regimes. We posit that testing on such pertinent benchmarks is crucial to showcase the practical applicability of our algorithm in privacy-sensitive scenarios. Intriguingly, we observe the benefits of our approach amplify as the privacy guarantees tighten, i.e., when ϵ is lower. Several practical deployments of DP, especially in the query release paradigm, have targeted low ϵ[1] but this remains elusive when deploying machine learning models. We hope our work will accelerate deployment of ML classification models with $\epsilon < 1$.
To summarise, our contributions are the following:

[1]https://desfontain.es/privacy/real-world-differential-privacy.html

- We introduce PILLAR, a straightforward, readily-implementable, and computationally inexpensive SP algorithm. It enhances classification accuracy compared to several existing competitive algorithms, some of which also exploit dimensionality reduction and semi-private learning principles.

- For learning half-spaces, we establish that our algorithm attains dimension-independent private labelled sample complexity with *large margin low rank distributions*. Significantly, our results are versatile, accommodating distribution shifts between public and private data, and adaptable to multiple loss functions.

- We refine privacy evaluation benchmarks for image classification, concentrating on scenarios that, in our view, hold greater relevance to privacy. These include i) private datasets exhibiting substantial shift from pre-training (and public) datasets, ii) the availability of limited (private and public) training data, and iii) stringent privacy regimes ($\epsilon < 1$)

5.2 Semi-Private Learning

We begin by defining Differential Privacy (DP). DP ensures that the output distribution of a randomized algorithm remains stable when a single data point is modified. In this paper, a differentially private learning algorithm produces comparable distributions over classifiers when trained on neighbouring datasets. Neighbouring datasets refer to datasets that differ by a single entry. Formally,

Definition 1 (Differential Privacy [Dwork et al., 2006])**.** *A learning algorithm $\mathcal{A}$ is (ϵ, δ)-differential private, if for any two datasets S, S' differing in one entry and for all outputs $\mathcal{Z}$, we have,*

$$\mathbb{P}\left[\mathcal{A}(S) \in \mathcal{Z}\right] \leq e^{\epsilon}\mathbb{P}\left[\mathcal{A}(S') \in \mathcal{Z}\right] + \delta.$$

For $\epsilon < 1$ and $\delta = o\left(1/n\right)$, (ϵ, δ)-differential privacy provides valid protection against potential privacy attacks [Carlini et al., 2022a].

Differential Privacy and Curse of Dimensionality Similar to non-private learning, the most common approach to DP learning is through Differentially Private Empirical Risk Minimization (DP-ERM), with the most popular optimization procedure being DP-SGD [Abadi et al., 2016] or analogous DP variants of typical optimization algorithms [Asi et al., 2021]. However, unlike non-private ERM, the sample complexity of DP-ERM suffers from a polynomial dependence on the dimensionality of the problem [Chaudhuri et al., 2011, Bassily et al., 2014a]. Hence,

we explore slight relaxations to this definition of privacy to alleviate this problem. We show theoretically (Section 5.3) and through extensive experiments (Section 5.4 and 5.5) that this is indeed possible with some realistic assumptions on the data and a slightly relaxed definition of privacy known as semi-private learning that we describe below. For a discussion of broader impacts and limitations of this setting, please refer to Section D.3.

5.2.1 Semi-Private Learning

The concept of semi-private learner was introduced in [Alon et al., 2019]. In this setting, the learning algorithm is assumed to have access to both a private labelled and a public (labelled or unlabelled) dataset. In this work, we assume the case of only having an *unlabelled* public dataset. This specific setting has been referred to as Semi-Supervised Semi-Private learning in [Alon et al., 2019]. However, for the sake of brevity, we will refer to it as Semi-Private learning (SPL).

Definition 2 ($(\alpha, \beta, \epsilon, \delta)$-semi-private learner on a family of distributions $\mathcal{D}$)**.** *An algorithm $\mathcal{A}$ is said to $(\alpha, \beta, \epsilon, \delta)$-semi-privately learn a hypothesis class $\mathcal{H}$ on a family of distributions $\mathcal{D}$, if for any distribution $D \in \mathcal{D}$, given a private labelled dataset S^L of size n^L and a public unlabelled dataset S^U of size n^U sampled i.i.d. from D, $\mathcal{A}$ is (ϵ, δ)-DP with respect to S^L and outputs a hypothesis $\hat{h}$ satisfying*

$$\mathbb{P}[\mathbb{P}_{(x,y)\sim D}[h(x) \neq y] \leq \alpha] \geq 1 - \beta,$$

where the outer probability is over the randomness of S^L, S^U, and $\mathcal{A}$.

Further, the sample complexity n^L and n^U must be polynomial in $\frac{1}{\alpha}, \frac{1}{\beta}$, and the size of the input space. In addition, n^L must also be polynomial in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$. The algorithm is said to be efficient if it also runs in time polynomial in $\frac{1}{\alpha}, \frac{1}{\beta}$, and the size of the input domain.

A key distinction between our work and the previous study by [Alon et al., 2019] is that they examine the distribution-independent agnostic learning setting, whereas we investigate the distribution-specific realisable setting. On the other hand, while their algorithm is computationally inefficient, ours can be run in time polynomial in the relevant parameters and implemented in practice on various datasets with state-of-the-art results. We discuss our algorithm in Section 5.2.2.

Relevance of Semi-Private Learning In various privacy-sensitive domains such as healthcare, legal, social security, and census data, there is often some amounts of publicly available data in addition to the private data. For instance, the U.S. Census Bureau office has partially released historical data before 2020 without enforcing any differential privacy guarantees [2]. It has also been

[2] `https://www2.census.gov/library/publications/decennial/2020/census-briefs/c2020br-03.pdf`

observed that different data providers may have varying levels of concerns about privacy [Jensen et al., 2005]. In medical data, some patients may consent to render some of their data public to foster research. In other cases, data may become public due to the expiration of the right to privacy after specific time limits [3].

It is also very likely that this public data may be *unlabelled* for the task at hand. For example, if data is collected to train a model to predict a certain disease, the true diagnosis may have been intentionally removed from the available public data to protect sensitive information of the patients. Further, the data may had been collected for a different purpose like a vaccine trial. Finally, the cost of labelling may be prohibitive in some cases. Hence, when public (unlabelled) data is already available, we focus on harnessing this additional data effectively, while safeguarding the privacy of the remaining private data. We hope this can lead to the development of highly performant algorithms which in turn can foster wider adoption of privacy-preserving techniques.

5.2.2 PILLAR: An Efficient Semi-Private Learner

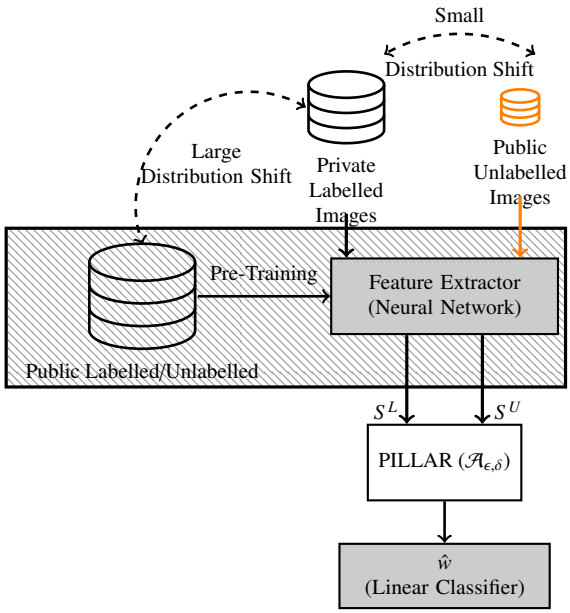


Figure 5.2.1: Diagram describing how PILLAR is applied in image classification (using DP-SGD with cross-entropy loss in Line 4 of Algorithm 1).

In this work, we propose a (semi-supervised) semi-private learning algorithm called PILLAR (PrIvate Learning with Low rAnk Representations), described in Algorithm 1. Before providing

[3] `https://www.census.gov/history/www/genealogy/decennial_census_records/the_72_year_rule_1.html`

Algorithm 1 PILLAR $\mathcal{A}_{\epsilon,\delta}(k,\ell)$ for learning halfspaces

1: **Input:** Labelled dataset S^L, Unlabelled dataset S^U, low-dimension k, L-Lipschitz loss function ℓ, high probability parameter β.
2: Using S^U, construct $\widehat{\Sigma} = \sum_{x \in S^U} xx^\top / n^U$.
3: Construct the transformation matrix $\hat{A}_k$ whose i^{th} column is the i^{th} eigenvector of $\widehat{\Sigma}$.
4: Project S^L with the transformation matrix $\hat{A}_k$,

$$S_k^L = \{(\hat{A}_k^\top x, y) : (x,y) \in S^L\}.$$

5: Obtain $v_k = \mathcal{A}_{\text{Noisy-SGD}}(S_k^L, \ell, (\epsilon,\delta), \beta/4)$
6: **Output:** Return $\hat{w} = \hat{A}_k v_k$.

formal guarantees in Section 5.3, we first describe how PILLAR is applied in practice. Our algorithm works in two stages.

Leveraging recent practices [De et al., 2022, Tramer and Boneh, 2021] in DP training with deep neural networks, we first use pre-trained feature extractors to transform the private labelled and public unlabelled datasets to the representation space to obtain the private and public representations. We use the representations in the penultimate layer of the pre-trained neural network for this purpose. As shown in Figure 5.2.1, the feature extractor is trained on large amounts of labelled or unlabelled public data, following whatever training procedure is deemed most suitable. For this paper, we pre-train a ResNet-50 using supervised training (SL), self-supervised training (BYOL [Grill et al., 2020] and MocoV2+ [Chen et al., 2020b]), and semi-supervised training (SemiSL and SemiWeakSL [Yalniz et al., 2019]) on ImageNet. In the main body, we only focus on SL and BYOL pre-training. As we discuss extensively in Section D.2.8, our algorithm is effective independent of the choice of the pre-training algorithm. In addition, while the private and public datasets are required to be from the same (or similar) distribution, we show that the pre-training dataset can come from a significantly different distribution. In fact, we use ImageNet as the pre-training dataset for all our experiments even when the distributions of the public and private datasets range from CIFAR-10/100 to histological and x-ray images as shown in Figure 5.1.1. Recently, [Gu et al., 2023] have explored the complementary question of how to choose the right pre-training dataset.

In the second stage, PILLAR takes as input the feature representations of the private labelled and public unlabelled datasets, and feeds them to Algorithm 1. We denote these datasets of representations as S^L and S^U respectively. Briefly, Algorithm 1 projects the private dataset S^L onto a low-dimensional space spanned by the top principal components estimated with S^U, and then applies gradient-based private algorithms (e.g. Noisy-SGD [Bassily et al., 2014a] in Section D.1.1) to learn a linear classifier on top of the projected features. Algorithm 1 provides an implementation

of PILLAR with Noisy-SGD, whereas in our experiments we show that commonly used DP-SGD [Abadi et al., 2016] is also effective [4].

5.3 Theoretical Results

In this section, we first describe the assumptions under which we provide our theoretical results and show they can be motivated both empirically and theoretically. Then, we show a dimension-independent sample complexity bound for PILLAR under the mentioned assumptions.

5.3.1 Problem setting

Our theoretical analysis focuses on learning linear halfspaces $\mathcal{H}^d$ in d dimensions. Consider the instance space $\mathcal{X}_d = B_2^d = \left\{x \in \mathbb{R}^d : \|x\|_2 = 1\right\}$ as the d-dimensional unit sphere and the binary label space $\mathcal{Y} = \{-1, 1\}$. In practice, the instance space is the (normalized) representation space obtained from the pre-trained network. The hypothesis class of linear halfspaces is

$$\mathcal{H}^d = \left\{f_w(x) = \text{sign}\left(\langle w, x\rangle\right) | w \in B_2^d\right\}.$$

We consider the setting of distribution-specific learning, where our family of distributions admits a large margin linear classifier that contains a significant projection on the top principal components of the population covariance matrix. We formalise this as (γ, ξ_k)-Large margin low rank distributions.In contrast to the usual low rank assumption on the feature matrix [Song et al., 2020], large margin low rank distributions can have full rank covariance matrix and generate full rank feature matrix, as long as the true parameter retains its norm in the low dimensional space spanned by the first k eigenvectors of the feature's covariance matrix.

Definition 3 ((γ, ξ_k)-Large margin low rank distribution)**.** *A distribution D over $\mathcal{X}_d \times \mathcal{Y}$ is a (γ, ξ_k)-Large margin low rank distribution if there exists $w^\star \in B_2^d$ such that*

- $\mathbb{P}_{(x,y)\sim D}\left[\frac{y\langle w^\star, x\rangle}{\|w^\star\|_2\|x\|_2} \geq \gamma\right] = 1$ *(Large-margin),*
- $\left\|A_k A_k^\top w^\star\right\|_2 \geq 1 - \xi_k$ *(Low-rank separability).*

where A_k is a $d \times k$ matrix whose columns are the top k eigenvectors of $\mathbb{E}_{X\sim D_X}\left[X^\top X\right]$.

[4] Other state-of-art adaptation of DP optimization algorithms, such as DP-SCO[Asi et al., 2021] and DP-RAFT[Panda et al., 2022], can also be applied in step 5 of PILLAR for potentially achieving better accuracy (see Section D.2.6 for further experiments).

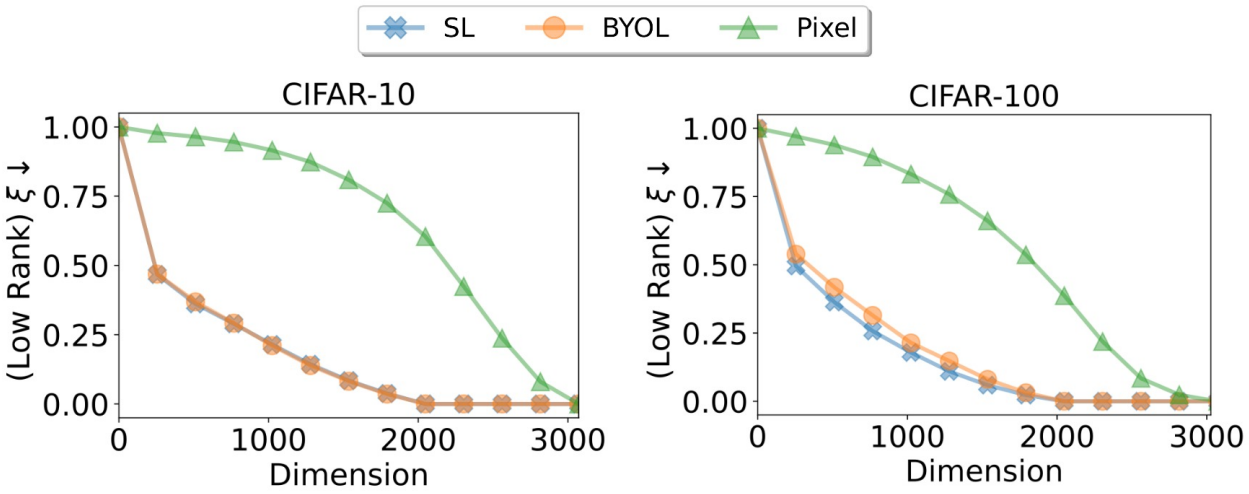


Figure 5.3.1: Estimate of ξ for linear classifiers trained on embeddings of two CIFAR-10 and CIFAR-100 classes, extracted from pre-trained ResNet50s, as well as the raw images (Pixel).

It is worth noting that for every distribution that admits a positive margin γ, the low-rank separability condition is automatically satisfied for all $k \leq d$ with some $\xi_k \geq 0$. Intuitively, this condition requires that there is a large margin classifier with significant projection on the top principal components of the data. However, the low rank separability is helpful for learning, only if it holds for a small k and small ξ_k simultaneously. These assumptions are both theoretically and empirically realisable. Theoretically, we show in Section D.1.5 that a class of commonly studied Gaussian mixture distributions with full rank covariance matrices satisfies these properties with $k = 2$ and $\xi_2 = 0$. Empirically, we show in Figure 5.3.1 that pre-trained features satisfy these properties with small ξ and k.

Pre-trained features are almost Large-Margin and Low-Rank Figure 5.3.1 shows that feature representations of CIFAR-10 and CIFAR-100 obtained by various pre-training strategies approximately satisfy the conditions of Definition 3. To verify the low-rank separability assumption, we first train a binary linear SVM $w^\star$ for a pair of classes on the representation space and estimate $\xi_k = 1 - \left\|A_k A_k^\top w^\star\right\|_2$ as defined in Definition 3. We also compute ξ_k when $w^\star$ is trained on the pixel space [5]. As shown in Figure 5.3.1, images in the representation space are better at satisfying the low-rank separability assumption compared to images in the pixel space.

[5]The estimate of ξ_k on pixel space should be taken with caution since classes are not linearly separable in the pixel space thereby only approximately satisfying the Large Margin assumption.

Loss function ℓ	Formula	Lipschitzness L_ℓ
Cross-entropy loss	$\frac{\log\left(1+e^{-y\langle w,x\rangle}\right)}{\log 2}$	2
Scaled hinge loss	$\max\left\{0, 1-\frac{y\langle w,x\rangle}{0.9\gamma_0(1-\xi_0)}\right\}$	$\frac{1}{0.9\gamma_0(1-\xi_0)}$

Table 5.3.1: Loss functions we consider in Theorem 1, with their expressions and the associated Lipschitz constants

5.3.2 Private labelled sample complexity analysis

In this section, we present the theoretical guarantees of PILLAR for Semi-Private learning of linear halfspaces. We prove that for binary cross entropy loss and hinge loss defined in Table 5.3.1, PILLAR is (ϵ, δ)-DP with respect to the private dataset and achieve high accuracy in learning linear halfspaces with relatively small number of private labelled data samples. Please refer to Section D.1.2 for the proof of Theorem 1.

Theorem 1. *Let $k \le d \in \mathbb{N}, \gamma_0 \in (0,1)$, and $\xi_0 \in (0,1)$. Consider the family of distributions $\mathcal{D}_{\gamma_0,\xi_0}$ which consists of all (γ, ξ_k)-large margin low rank distributions over $\mathcal{X}_d \times \mathcal{Y}$, where $\gamma \ge \gamma_0$ and $\xi_k \le \xi_0$. For any $\alpha \in (0,1), \beta \in (0, 1/4)$, $\epsilon \in (0, 1/\sqrt{k})$, and $\delta \in (0,1)$, PILLAR with scaled hinge loss or cross entropy loss, is an $(\alpha, \beta, \epsilon, \delta)$-semi-private learner for linear halfspaces $\mathcal{H}^d$ on $\mathcal{D}_{\gamma_0,\xi_0}$ with sample complexity*

$$n^U = O\left(\frac{\log 2/\beta}{(1-\xi_0)^2\Delta_k^2}\right), n^L = \widetilde{O}\left(\frac{L_\ell\sqrt{k}}{\alpha\epsilon}\right)$$

where Δ_k denote the gap between the k^{th} and the $k+1^{\text{th}}$ eigenvalue of the population covariance matrix, and L_ℓ is the Lipschitz coefficient of the loss function ℓ[6].

Table 5.3.1 provides a summary of two loss functions and the associated Lipschitz coefficients. Notably, the Lipschitz coefficient $L_\ell = \frac{1.1}{\gamma_0(1-\xi_0)}$ for the scaled hinge loss varies with the distributional parameters γ_0 and ξ_0. In contrast, for cross-entropy loss, L_ℓ remains fixed at 2. Hence, PILLAR with scaled hinge loss is inherently designed to better harness the large margin property of the distribution with large γ_0 and small ξ_0. On the other hand, PILLAR with cross entropy loss reflects the experiments more closely.

As discussed in Section 5.3.1, the feature representations of images, obtained from pre-trained neural networks, usually satisfy the properties of large-margin low rank distributions (Figure 5.3.1). Thus, in practical implementation, the private and public datasets refer to private and public

[6]Note that $\widetilde{O}$ neglects the logarithmic terms associated with $\frac{1}{\delta}$ and $\frac{1}{\beta}$.

representations, as shown in Figure 5.2.1. Note that while Theorem 1 only guarantees (ϵ, δ)-DP on the set of private representations (see Figure 5.2.1), this guarantee can also extend to (ϵ, δ)-DP on the private labelled image dataset. See Section D.1.2 for more details.

As a concrete instance of the application of Theorem 1, we formally define a family of distributions based on gaussian mixtures, referred to as large margin Gaussian mixture distributions, in Section D.1.5. For this family of distributions, we demonstrate through Theorem 1 that PILLAR significantly reduces the private sample complexity from $O(\sqrt{d})$ to $O(1)$.

5.3.3 Distribution shift between private and public datasets

PILLAR also provides theoretical guarantees when the private and public representations come from similar, but not identical distributions. In this case, private sample complexity also depends on the Total Variation (TV) distance, say η between the two distributions. An informal theorem is presented below in Theorem 2, while the formal result can be found in Section D.1.4.

Theorem 2. *Let $k, d, \gamma_0, \xi_0, \mathcal{D}_{\gamma_0,\xi_k}$, $\alpha, \beta, \epsilon, n^L, n^U$ and δ be defined as in Theorem 1. Additionally, consider any $\eta \in \left[0, 9(1-\xi_0)\Delta_k/140\right)$. Then PILLAR with scaled hinge loss satisfies the same guarantees as Theorem 1 with $1/L_\ell = \gamma_0\left(1 - \xi_0 - \frac{140\eta}{9\Delta_k}\right)$ as long as the distributions of the private and public datasets are within η Total variation.*

5.3.4 Comparison with existing theoretical results and discussion

Existing works have offered a variety of techniques for achieving dimension-independent sample complexity. In the following, we review these works and compare them with our approach.

Generic private algorithms [Bassily et al., 2014b] proposed the Noisy SGD algorithm $\mathcal{A}_{\text{Noisy-SGD}}$ that can privately learn linear halfspaces with margin γ on a private labelled dataset of size $O(\sqrt{d}/\alpha\epsilon\gamma)$. Recently, [Li et al., 2022b] showed that DP-SGD, a slightly adapted version of $\mathcal{A}_{\text{Noisy-SGD}}$, can achieve a dimension independent error bound under a low-dimensionality assumption termed as Restricted Lipschitz Continuity (RLC), which is more restrictive than our low-rank separability assumption. Similar results were showed in [Song et al., 2021]. However, these methods cannot utilise public unlabelled data. [Nasr et al., 2023] leverages public data for gradient clipping in DP-SGD. However, their method does not achieve dimension-independent error bound. The generic semi-private learner in [Alon et al., 2019] leverages unlabelled data to reduce the infinite hypothesis class to a finite α-net and applies exponential mechanism [McSherry

and Talwar, 2007] to achieve $(\epsilon, 0)$-DP. Nonetheless, it is not computationally efficient and still requires a dimension-dependent labelled sample complexity $O\left(d/\alpha\epsilon\right)$.

Dimension reduction based private algorithms Perhaps, most relevant to our work, [Nguyen et al., 2020] applies Johnson-Lindenstrauss (JL) transformation in the input space to reduce the dimension of a linear halfspace with margin γ from d to $O\left(1/\gamma\right)$ while preserving the margin in the lower-dimensional space. Private learning in the transformed low-dimensional space requires $O\left(1/\alpha\epsilon\gamma^2\right)$ labelled samples. Our algorithm removes the quadratic dependence on the inverse of the margin but pays the price of requiring the linear separator to align with the top few principal components of the data. Specifically, the benefit in private sample complexity is significant when $k = o\left(\log(n)/\gamma^2\right)$, which is a realistic condition as k is often independent of n and γ is usually small.

Another approach to circumvent the dependency on the dimension is to apply dimension reduction techniques directly to the gradients. For smooth loss functions with ρ-Lipschitz and G-bounded gradients, [Zhou et al., 2021] showed that applying PCA in the gradient space of DP-SGD [Abadi et al., 2016] achieves dimension-independent labelled sample complexity $O\left(\frac{k\rho G^2}{\alpha\epsilon} + \frac{\rho^2 G^4 \log d}{\alpha}\right)$. However, this algorithm is computationally costly as it applies PCA in every gradient-descent step to a matrix whose size scales with the number of parameters. [Kasiviswanathan, 2021] proposed a computationally efficient method by applying JL transformation in the gradient space. While their method can eliminate the linear dependence of DP-SGD on dimension when the parameter space is the ℓ_1-ball, it leads to no improvement for parameter space being the ℓ_2-ball as in our setting. Gradient Embedding Perturbation (GEP) by [Yu et al., 2021a] is another computationally efficient method that exploit the low-dimensionality of the gradient space with public unlabelled data. However, their analysis yields dimension independent guarantees only when a strict low-rank assumption of the gradient space is satisfied. Similar assumptions were leveraged by [Kairouz et al., 2020] who proposed a private adaptive gradient method to achieve dimension independent error bounds. Their final error bounds are very similar to [Song et al., 2021]. We compare the assumptions in more detail in Section D.1.6.

Private PCA (DP-PCA) Another natural algorithm is to first project the private labelled data to its top k principal components estimated using DP-PCA on both the private and the public data, and then apply DP-SGD to learn a linear classifier in the k-dimensional space [Abadi et al., 2016]. However, estimating the top principal components using DP-PCA on $O(n^U + n^L)$ samples in Theorem 1 introduces an irreducible error of $\Omega\left(\min\left\{\gamma_0^2 d, \frac{d}{\alpha\sqrt{k}}\right\}\right)$ in the estimated space (Theorem 5.4 of [Liu et al., 2022a]), making the lower-dimensional space linearly inseparable for large d.

Hence, the classification error of any linear classifier in the low dimensional space does not converge to zero using the same amount of data required for PILLAR.

Importantly, we compare against these algorithms in our experiments and show a consistent improvement, often by a wide margin, on a variety of datasets.

Non-private learning and dimensionality reduction It is interesting to note that our algorithm may not lead to a similar improvement in the non-private case. We show a dimension-independent Rademacher-based labelled sample complexity bound for non-private learning of linear halfspaces. We use a non-private version of Algorithm 1 by replacing Noisy-SGD with Gradient Descent using the same loss function. As before, for any $\gamma_0 \in (0,1), \xi_0 \in (0,1)$, let $\mathcal{D}_{\gamma_0,\xi_0}$ be the family of distributions consisting of all (γ, ξ_k)-large margin low rank distributions with $\gamma \geq \gamma_0$ and $\xi_k \leq \xi_0$.

Proposition 3 (Non-DP learning)**.** *For any $\alpha, \beta \in (0, 1/4)$, and distribution $D \in \mathcal{D}_{\gamma_0,\xi_0}$, given a labelled dataset of size $\tilde{O}\left(1/\zeta\alpha^2\right)$ and unlabelled dataset of size $O\left(\log\frac{2}{\beta}/(\gamma_0\Delta_k)^2\right)$, the non-private version of $\mathcal{A}(k,\zeta)$ produces a linear classifier $\hat{w}$ such that with probability $1-\beta$*

$$\mathbb{P}_D\left[y\langle\hat{w}, x\rangle < 0\right] < \alpha,$$

where $\zeta = \gamma_0(1-\xi_0)$.

The result follows directly from the uniform convergence of linear halfspaces with Rademacher complexity. For example, refer to Theorem 1 in [Awasthi et al., 2020]. The labelled sample complexity in the above result shows that non-private algorithms do not significantly benefit from decreasing dimensionality[7]. We find this trend to be true in all our experiments in Figure 5.4.1 and 5.4.2.

In summary, our algorithm is computationally efficient and under certain (realistic) assumptions on the data, yields dimension independent private sample complexity. We also show through a wide variety of experiments in the following sections that the results transfer to practice in both common benchmarks as well as many newly designed challenging settings.

5.4 Results on Standard Image Classification Benchmarks

In this section, we report performance of PILLAR on two standard benchmarks (CIFAR-10 and CIFAR-100 [Krizhevsky, 2009]) for private image classification. We demonstrate that in this setting, PILLAR outperforms all the competing methods. The improvement is especially remarkable

[7]However, this bound uses a standard Rademacher complexity result and may be lose. A tighter complexity bound may yield some dependence on the projected dimension.

for low ϵ values where there is a significant margin for improvement. For moderate values of ϵ, the improvement is more modest.

5.4.1 Experimental setting

The resolution difference between ImageNet-1K and CIFAR images can negatively impact the performance of training a linear classifier on pre-trained features. To mitigate this issue, we pre-process the CIFAR images using the ImageNet-1K transformation pipeline, which increases their resolution and leads to significantly improved performance. This technique is consistently applied throughout the paper whenever there is a notable resolution disparity between the pre-training and private datasets. For further details and discussions on pre-training at different resolutions, please refer to Section D.2.2.

We diverge from previous studies in the literature, such as those conducted by [De et al., 2022, Tramer and Boneh, 2021, Kurakin et al., 2022], by not exclusively focusing on values of $\epsilon > 1$. While a moderately large ϵ can be insightful for assessing the effectiveness of privately training deep neural networks with acceptable levels of accuracy, it is important to acknowledge that a large value of ϵ can result in loose privacy guarantees and consequently lack of willingness to share data [Nanayakkara et al., 2023] . The seminal work of [Dwork, 2011] emphasizes that reasonable values of ϵ are expected to be less than 1. Moreover, [Yeom et al., 2018] and [Nasr et al., 2021] have already highlighted that $\epsilon > 1$ leads to loose upper bounds on the success probability of membership inference attacks. Finally, several recent deployments of DP have use values of ϵ smaller than 1[8]. Consequently, we focus on $\epsilon \in \{0.1, 0.3, 0.7, \infty\}$, where $\epsilon = \infty$ corresponds to the public training of the linear classifier. Nevertheless, for completeness and consistency with the current literature, we also present additional results for higher $\epsilon = \{1, 2\}$ in Section D.2.3.

5.4.2 Comparison with Existing Methods

We now compare the performance of PILLAR against several baselines that also leverage either public data or dimensionality reduction or both. We use the same PRV accountant for all methods [Gopi et al., 2021a]. For a comprehensive discussion on implementation details and the cross-validation ranges for hyper-parameters across all methods, refer to Section D.2.7.

[8]https://desfontain.es/privacy/real-world-differential-privacy.html

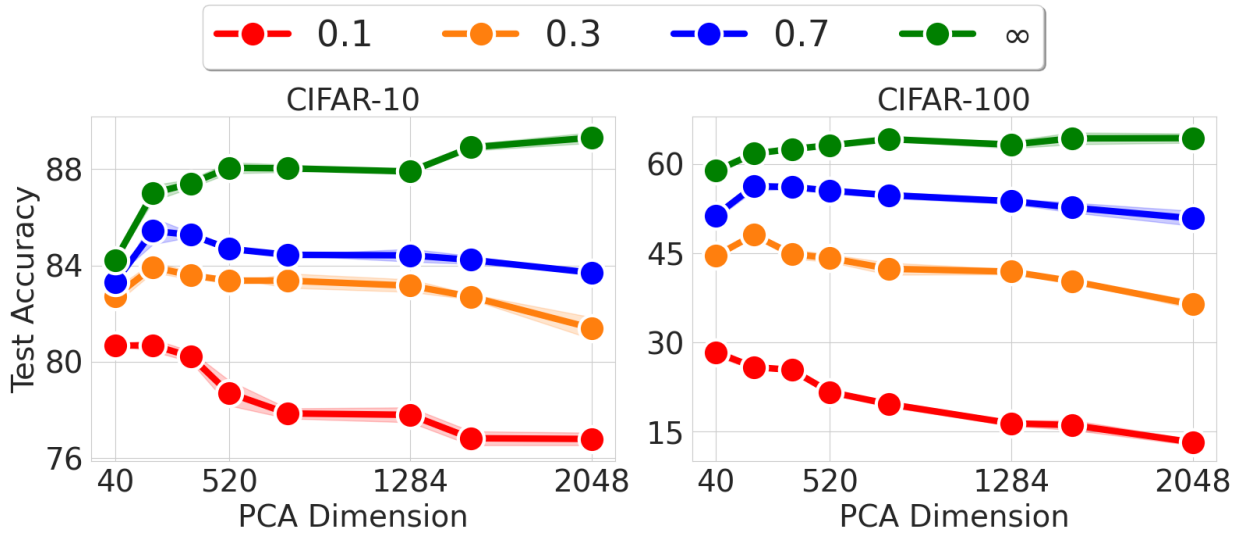


Figure 5.4.1: DP training of linear classifier on SL pre-trained feature using the PRV accountant. For non-DP training ($\epsilon = \infty$), accuracy increases as dimension increases; opposite occurs for DP training ($\epsilon = \{0.1, 0.3, 0.7\}$). For results on additional feature-extractors see Section D.2.8.

	Public Data	SL Pre-training				BYOL Pre-training		
Datasets		CIFAR10	CIFAR100	Flower-16	GTSRB	Dermnet	PCAM	Pneumonia
ϵ		0.1 0.3 0.7	0.1 0.3 0.7	0.1 0.3 0.7	0.1 0.3 0.7	0.1 0.3 0.7	0.1 0.3 0.7	0.1 0.3 0.7
DP-SGD	None	76.8 81.4 83.7	13.2 36.4 50.9	46.2 72.2 82.4	36.0 46.8 59.6	14.4 22.4 28.0	80.1 81.4 81.6	71.3 73.6 79.2
DP-PCA	None	72.1 77.5 81.2	10.2 34.9 48.3	46.2 69.6 76.1	35.1 50.0 58.0	15.4 22.9 27.6	80.1 81.9 81.1	66.7 68.3 79.3
JL	None	76.1 82.1 84.1	13.7 37.6 51.3	43.5 70.4 80.9	36.3 53.3 62.1	14.3 22.3 28.1	78.3 78.7 79.7	65.2 70.0 76.9
GEP	Unlabelled	80.1 83.2 84.5	12.4 41.2 45.2	59.1 78.5 82.8	38.7 58.2 61.2	18.3 24.6 27.7	79.1 82.0 81.7	75.9 78.5 82.9
AdaDPS	Labelled	66.3 80.9 83.2	13.0 33.2 39.4	30.2 69.4 75.9	24.1 49.1 54.4	8.2 21.1 24.6	78.2 79.3 81.4	76.4 74.2 81.3
OURS	Unlabelled	**81.2 84.0 85.5**	**28.3 48.0 53.9**	**67.3 81.8 85.1**	**46.3 59.1 66.0**	**19.5 26.4 29.1**	**82.6 82.6 82.7**	**83.4 84.3 85.7**

Table 5.4.1: Empirical comparison of PILLAR (OURS) against several baselines with different assumptions about the availability of public data. For the first four datasets (CIFAR-10, CIFAR-100, Flower-16, GTSRB), we use a SL pre-trained feature extractor, as it yields the best performance. For the last three datasets (Dermnet, PCAM, Pneumonia) we use a BYOL pre-trained feature extractor. In all cases, PILLAR outperforms all baselines under several levels of tightness of the privacy constraints ($\epsilon = \{0.1, 0.3, 0.7\}$). Baselines are implemented with the official, publicly available implementation when available. We use the PRV accountant. See Section D.2.7 for more details.

Baselines We consider the following baselines:

- *DP-SGD* [Abadi et al., 2016, Li et al., 2022b]: Trains a linear classifier privately using DP-SGD on the pre-trained features.
- *JL* [Nguyen et al., 2020]: Applies a Johnson-Lindenstrauss (JL) transformation (without utilizing public data) to reduce the dimensionality of the features. We cross-validate various target dimensionalities and report the results for the most accurate one.

- *AdaDPS* [Li et al., 2022a]: Utilizes the public *labeled* data to compute the pre-conditioning matrix for adaptive optimization algorithms. Since our algorithm does not require access to labels for the public data, we consider this comparison; nevertheless we report their performance.

- *GEP* [Yu et al., 2021a]: Employs the public *unlabeled* data to decompose the private gradients into a low-dimensional embedding and a residual component, subsequently perturbing them with noise of different magnitudes.

- *DP-PCA* [Abadi et al., 2016] applies a step of DP-PCA (which consumes a fraction of the privacy budget) to compute the PCA components and then trains a linear classifier. We consider using $1\%, 25\%, 50\%$ of the privacy budget and report the results for the best choice.

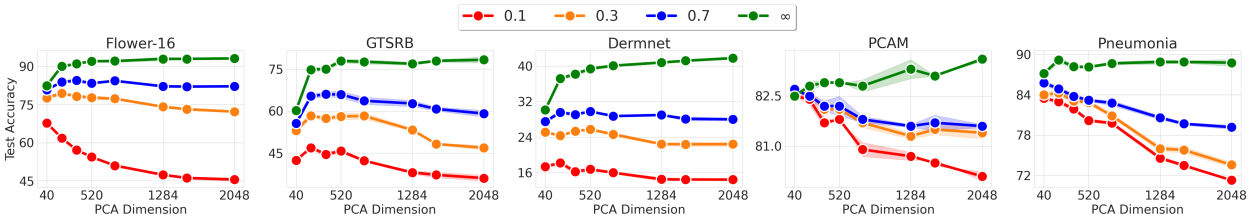


Figure 5.4.2: Test Accuracy of DP classification on Flower-16, GTSRB, Dermnet, PCAM, and Pneumonia for best pre-training algorithm (SL pre-training for Flower-16 and GTSRB and BYOL for the remaining.). For results on additional feature-extractors refer to Section D.2.8.

Whenever public data is utilized, we employ 10% of the training data as public and remaining data as private. The official implementations of AdaDPS and GEP are used for our comparisons. Compared to baselines like AdaDPS and GEP, PILLAR introduces only one hyperparameter (the dimensionality k), making it less computationally expensive to cross-validate (as discussed in Section D.2.2). It is also extremely simple to implement, and therefore less prone to bugs that may invalidate the privacy guarantees.

In Section D.2.4, we discuss PATE [Papernot et al., 2017, 2018] and the reasons for not including it in our comparisons. For a detailed comparison with the work of [De et al., 2022], including the use of a different feature extractor to ensure a fair evaluation, we refer to Section D.2.2, where we demonstrate that our method is competitive, if not superior, while enjoying significantly more computational efficiency.

Results In Table 5.4.1, we compare our approach with other methods in the literature. Our results suggest that reducing dimensionality by using the JL transformation can only marginally ($\leq 1\%$ for both CIFAR-10 and CIFAR-100) improve over DP-SGD and sometimes even perform worse than DP-SGD. This may be attributed to the higher sample size required for the JL lemma to provide meaningful guarantees. Similarly, employing public data to pre-condition an adaptive optimizer does not result in improved performance for AdaDPS in most settings. The most competitive baseline is often GEP, however *PILLAR consistently outperforms all of them often with large margins.* For instance, consider the challenging setting of CIFAR-100 with $\epsilon = 0.1$. The performance of DP-trained classifiers is particularly low on this dataset because there are only 500 samples for each class. DP-SGD only achieves 13.2% accuracy for $\epsilon = 0.1$ whereas non-private accuracy is more than 80%. In this case no baseline yields performance significantly superior to DP-SGD except PILLAR, which is accurate by more than a factor of two. For $\epsilon = 0.3$, PILLAR outperforms the strongest baseline, GEP, by 6.8%. For $\epsilon = 0.7$, DP-SGD is again the strongest-baseline, and we outperform it by 3.0%.

5.4.3 Reducing dimension of projection k helps private learning

In Figure 5.4.1, we present the test accuracy of private and non-private trianing on CIFAR-10 and CIFAR-100 as the dimensionality of projection (PCA dimension) varies, with an initial embedding dimension of $k = 2048$. The principal components are computed on a public, unlabelled dataset that constitutes 10% of the full dataset, as allowed by Semi-Private Learning in Definition 4. Our results demonstrate that private training benefits from decreasing dimensionality, while non-private training either suffers in performance or remains stagnant. For example, using the SL feature extractor at $\epsilon = 0.1$ on CIFAR-10, the test accuracy of private training reaches 81.21% when $k = 40$, compared to 76.9% without dimensionality reduction. Similarly, for CIFAR-100 with the SL feature extractor at $\epsilon = 0.7$, the accuracy drops from 53.98% at $k = 200$ to 50.83% for the full dimension.

This observed dichotomy between private and non-private learning in terms of test accuracy and projection dimension aligns with Theorem 1 and Proposition 3. Theorem 1 indicates that the private test accuracy improves as the projection dimension decreases, as depicted in Figure 5.4.1. For non-private training with moderately large dimension, ($k \geq 520$), the test accuracy remains largely constant. We discuss this theoretically in Proposition 3. The decrease in non-private accuracy for very small values of k is attributed to the increasing approximation error (i.e. how well can the best classifier in k dimensions represent the ground truth). This difference in behaviour between private and non-private learning for decreasing k values consistently holds in all our experiments and is one of the interesting observations of this paper. While we have demonstrated the effectiveness

of our algorithm on the CIFAR-10 and CIFAR-100 benchmarks, as discussed in Section 5.5, we acknowledge that this evaluation setting may not fully reflect the actual objectives of private learning.

5.5 Experimental Results Beyond Standard Benchmarks

In line with concurrent work [Tramèr et al., 2022], we raise concerns regarding the current trend of utilizing pre-trained feature extractors for differentially private training [De et al., 2022, Tramer and Boneh, 2021]. It is common practice to evaluate differentially private algorithms for image classification by pre-training on ImageNet-1K and performing private fine-tuning on CIFAR datasets [De et al., 2022, Tramer and Boneh, 2021]. However, we argue that this approach may not yield generalisable insights for privacy-sensitive scenarios. Both ImageNet and CIFAR datasets primarily consist of everyday objects, and the label sets of ImageNet are partially included within CIFAR. Such a scenario is unrealistic for many privacy-sensitive applications, such as medical, finance, and satellite data, where a large publicly available pre-training dataset with similar characteristics to the private data may not be accessible.

Moreover, public datasets are typically large-scale and easily scraped from the web, whereas private data is often collected on a smaller scale and subject to legal and competitive constraints, making it difficult to combine with other private datasets. Additionally, labeling private data, particularly in domains such as medical or biochemical datasets, can be costly. Therefore, evaluating the performance of privacy-preserving algorithms requires examining their robustness with respect to small dataset sizes. In order to address these considerations, we assess the performance of our algorithm on five additional datasets that exhibit varying degrees of distribution shift compared to the pre-training set, as described in Section 5.5.1. Furthermore, we also demonstrate the robustness of our algorithm to minor distribution shifts between public unlabeled and private labeled data. In Section 5.5.2, we show our algorithm is also robust to both small-sized private labeled datasets and public unlabeled datasets.

5.5.1 Effectiveness under Distribution Shift

Distribution Shift between Pre-Training and Private Data We consider private datasets that exhibit varying levels of dissimilarity compared to the ImageNet pre-training dataset: Flower-16 [Flo, 2021], GTSRB [Houben et al., 2013], Pneumonia [Kermany et al., 2018], a fraction (12.5%) of PCAM [Veeling et al., 2018], and DermNet [Der, 2019]. In Figure 5.1.1, we provide visual samples from each of these datasets. Flower-16 and GTSRB have minimal overlap with ImageNet-1K, with only one class in Flower-16 and 43 traffic signs aggregated into a single label in ImageNet-1K. The

Pneumonia, PCAM, and DermNet datasets do not share any classes with ImageNet-1K. We also observe that, given a fixed pre-training distribution and model, different training procedures can have a different impact in the utility of the extracted features for each downstream classification task. Therefore, for each dataset we report the best performance produced by the most useful pre-training algorithm. Results for all the 5 pre-training strategies we consider and a discussion of how to choose them is relegated to Section D.2.8.

From Table 5.4.1, we can see PILLAR outperforms all the considered baselines for all the ϵ values on all datasets. Before providing a more detailed discussion of the results, we would like to emphasize that no baseline consistently achieves the best performance across all these settings, in contrast to PILLAR, which proves to be a more consistent and widely applicable algorithm. On Flower-16, PILLAR achieves remarkable improvements. For $\epsilon = 0.1$, it outperforms the strongest baseline (GEP) by 8.2%. Similarly, on GTSRB we attain improvements of 3.9% over the runner-up (JL) for $\epsilon = 0.7$ and 8.4% with respect to GEP for $\epsilon = 0.1$. In the case of PCAM, although the relatively large training set size and the simplicity of the binary classification problem allows all classifiers to produce moderately high levels of accuracy (approximately 80%), our method is the only one to maintain an accuracy of approximately 82.6% across all the considered ϵ values, thus alleviating the utility degradation incurred by imposing tighter privacy constraints. In contrast, the Pneumonia dataset is a binary classification dataset with significantly less training data. In this case, competing techniques incur a significantly larger utility cost. For $\epsilon = 0.1$, the strongest baseline (AdaDPS) achieves 76.4%, while our method achieves 83.4%. *In summary, PILLAR consistently achieves the highest performance, often by a large margin, among all baselines for a wide range of datasets.*

In Figure 5.4.2, we demonstrate that reducing the dimensionality of the pre-trained models enhances differentially private training, irrespective of the private dataset used. Dimensionality reduction has a more pronounced effect on performance when tighter privacy constraints are imposed. It is worth noting that using dimensionality reduction can significantly degrade performance for non-DP training, similar to what we observed in CIFAR-10 and CIFAR-100.

When to use labels in pre-training We also investigate the impact of different pre-training strategies on DP test accuracy. In our experiments, we have observed that some pre-trained models are more effective than others for specific datasets. To measure the maximum attainable accuracy with a publicly trained classifier, we compute the drop in performance, observed by training a DP classifier on BYOL pre-trained features, and the drop in performance for SL pre-trained features. We then plot the fractional reduction for both BYOL and SL across all the datasets

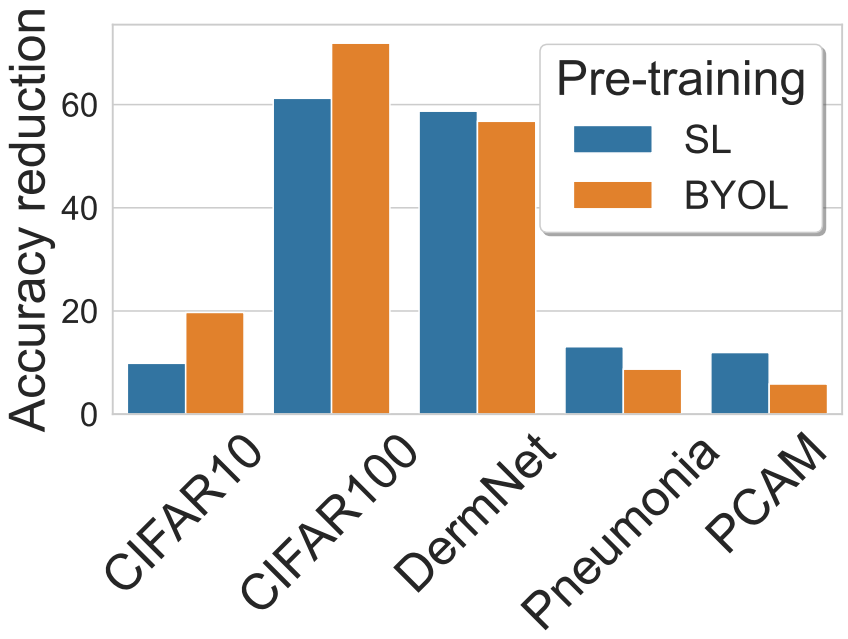


Figure 5.5.1: Comparing the difference between the maximum attainable test accuracy with a publicly trained linear classifier and a DP trained linear classifier between using SL and BYOL pre-trained networks for different datasets. SL suffers a smaller drop in accuracy is more useful when the fine-tuning dataset contains daily-life objects and semantically overlap with ImageNet-1K, BYOL performs better otherwise otherwise.

for $\epsilon = 0.1$ in Figure 5.5.1. In Figure D.2.4 we compare the relative reduction in performance when using Semi-supervised pre-training and BYOL pre-training. We find that datasets with daily-life objects and semantic overlap with ImageNet-1K benefit more from leveraging SL features and thus have a smaller reduction in accuracy for SL features compared to BYOL features. In contrast, datasets with little label overlap with ImageNet-1K benefit more from BYOL features, consistent with findings by [Shi et al., 2023].

	CIFAR10		CIFAR100	
Pre-training / PCA Data	SL	BYOL	SL	BYOL
In-distribution	81.21	72.33	28.3	19.98
CIFAR-10v1	81.18	73.24	28.19	19.61

Table 5.5.1: Distribution Shift between public (PCA) and private data: Comparison between using the same amount of in-distribution data (i.e. 10% of CIFAR-10 and CIFAR-100 respectively) and CIFAR-10v1 for computing the PCA projection ($\epsilon = 0.1$).

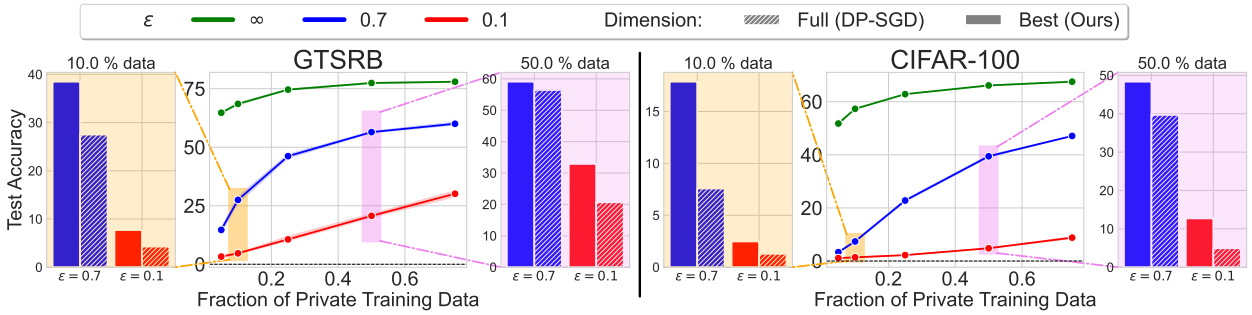


Figure 5.5.2: For the GTSRB and CIFAR-100 datasets, in the central panel we report how the test accuracy varies as the amount of available private training data decreases (fraction of available data in $\{0.05, 0.1, 0.25, 0.5, 0.75\}$) for $\epsilon = 0.1$ and 0.7. We then select the cases in which 10% and 50% of the samples are available (left orange and right pink panels, respectively) and compare how PILLAR (solid bars) behaves with respect to DP-SGD (dashed bars). As it can be seen, PILLAR can alleviate the utility degradation caused by the reduced availability of private training data.

Distribution Shift between S^U and S^L We demonstrate the effectiveness of our algorithm even when the public unlabeled data (used for computing the PCA projection matrix) is sourced from a slightly different distribution than the private labeled dataset. Specifically, we utilize the CIFAR-10v1 [Recht et al., 2018b] dataset and present the results in Table 5.5.1.

Notably, CIFAR-10v1 consists of only 2000 samples (4% of the training data), yet the results for both CIFAR-10 and CIFAR-100 remain essentially unchanged. This finding indicates that the data used to compute the PCA projection matrix does not necessarily have to originate from the same distribution as the private data and underscores that large amounts of public data are not required for our method to be effective.

5.5.2 Effectiveness in Low-Data Regimes

In privacy-critical settings such as medical contexts, there is often a limited availability of training data. For instance, the DermNet and pneumonia datasets contain only 12,000 and 3,400 training data points, respectively, which is significantly smaller compared to datasets like CIFAR-10 with 50,000 samples. To examine the impact of reduced data (both private labeled and public unlabeled) on privacy, in this section we conduct ablations using varying fractions of public and private training data.

Less public unlabelled data We demonstrate the robustness of our algorithm to reduced amounts of public unlabeled data used to compute the Principal Components.

PCA Data \ Pre-training	CIFAR10		GTSRB	
	SL	BYOL	SL	BYOL
1%	79.93	72.27	45.59	35.91
5%	81.02	72.33	45.64	35.88
10%	81.21	72.33	46.32	35.97

Table 5.5.2: Varying amounts of public (PCA) data: Performance of PILLAR with varying amounts of public (in distribution) data for computing the PCA projection ($\epsilon = 0.1$). The amount of public data is presented as a fraction of the whole available dataset.

In Table 5.5.2, we show the results of this ablation. As it can be seen, reducing the available public data does not yield dramatic variations in performance under the tightest privacy guarantees we consider ($\epsilon = 0.1$). For instance, for CIFAR-10 and GTSRB using a BYOL trained feature extractor, we observe the performance does not vary at all when the amount of available public data is reduced from 10% to 5% and 1%. For a SL trained feature extractor, we observe the performance only marginally decreases. For GTSRB, the performance reduces only by 0.93% when passing from 10% to 1% available public data, and of 1.28% on CIFAR-10 in the same setting.

Less private labelled data In Figure 5.5.2, we present the performance of private and public training using different percentages of labeled private training data for CIFAR-100 and GTSRB. Our results indicate that under stringent privacy constraints $\left(\epsilon \in \{0, 7, 0.1\}\right)$, the performance of DP training, without dimensionality reduction (DP-SGD), is considerably low. Conversely, even with a small percentage of training data, non-DP training demonstrates relatively high performance. By applying our algorithm in this scenario, we achieve significant performance improvements compared to using the full-dimensional embeddings. For instance, applying PCA with with $k = 40$ dimensions enhances the accuracy of our proposed algorithm from 7.53% to 18.3% on 10% of CIFAR-100, with $\epsilon = 0.7$ using the SL feature extractor. Similar improvements are also shown for GTSRB: when 10% of the data is available, the test accuracy improves from 27.3% to 38.4% for $\epsilon = 0.7$. To a smaller extent, improvements can be also observed when $\epsilon = 0.1$.

5.6 Conclusion

In this paper, we consider the setting of semi-private learning where the learner has access to public unlabelled data in addition to private labelled data. This is a realistic setting in many circumstances e.g. where some people choose to make their data public. Under this setting, we proposed a new algorithm to learn linear halfspaces. Our algorithm uses a mix of PCA on unlabelled data and DP

training on private data. Under reasonable theoretical assumptions, we have shown the proposed algorithm is (ϵ, δ)-DP and provably reduces the sample complexity. In practical applications, we performed an extensive set of experiments that show the proposed technique is effective when tight privacy constraints are imposed, even in low-data regimes and with a significant distribution shift between the pre-training and private distribution. In particular our algorithm consistently outperforms existing methods, often by a wide margin.

6

Extracting Training Data from Document-Based VQA Models

Contents

Abstract

Vision-Language Models (VLMs) have made remarkable progress in document-based Visual Question Answering (i.e., responding to queries about the contents of an input document provided as an image). In this work, we show these models can memorize responses for training samples and regurgitate them even when the relevant visual information has been removed. This includes Personal Identifiable Information (PII) repeated *once* in the training set, indicating these models could divulge memorised sensitive information and therefore pose a privacy risk. We quantitatively measure the extractability of information in controlled experiments and differentiate between cases where it arises from generalization capabilities or from memorization. We further investigate the factors that influence memorization across multiple state-of-the-art models and propose an effective heuristic countermeasure that empirically prevents the extractability of PII.

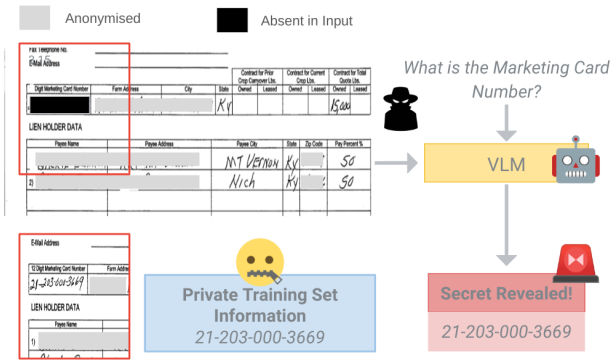


Figure 6.1.1: A malicious user may prompt a Vision-Language Model (VLM) to reveal secret information about a victim by generating a copy of the original document with the secret information missing (black box). If the secret was part of the training question-answer pairs, the VLM may respond correctly. For ethical reasons, we anonymize (grey boxes) personal information of a DocVQA [Mathew et al., 2021] sample on which the attack is successful for the Donut model [Kim et al., 2022]. The answer is repeated *only once* in the whole training set, yet it is memorized.

6.1 Introduction

Document-Based Visual Question Answering [Mathew et al., 2021]—the task of answering questions about the content of documents presented as visual inputs—has witnessed remarkable advancements in recent years, with modern Vision-Language Models (VLMs) gaining the ability to comprehend textual information exclusively from visual cues and provide accurate responses [Davis et al., 2022, Lee et al., 2023, Kim et al., 2022, Chen et al., 2023b,a, GPT].

However, our paper exposes a concerning behavior of these models: even when the answer to a question is explicitly removed from the input image and is unique or sporadically repeated across the training set, the VLM can still provide the correct response. This ability, which we refer to as *extractability* of the answers given some input context, indicates that the VLM may have either memorized the answer from a specific training sample [Feldman, 2019, Carlini et al., 2023a, Lukasik et al., 2023] or learned a distributional shortcut that allows to infer it from spurious features [Jabri et al., 2016, Niu et al., 2021, Goyal et al., 2017, Dancette et al., 2021, Tito et al., 2023]. We show that, in some cases, sensitive information can be extracted even when it appears only in a single training sample (see Figure 6.1.1). In order to fix this unintended behaviour of the models, we introduce a simple mitigation strategy that reduces the amount of extractable PIIs to zero.

In this study, we investigate this phenomenon across three state-of-the-art Document-Based VQA models: Donut [Kim et al., 2022], Pix2Struct [Lee et al., 2023] and PALI-3 [Chen et al., 2023b]). We evaluate their behaviour on the popular Document Visual Question Answering (DocVQA) dataset [Mathew et al., 2021], which consists of a public collection of pages from

industrial documents accompanied by questions and answers for a purely extractive purpose (i.e., the task only necessitates reading the document without any additional reasoning). We propose a series of controlled experiments on in-distribution canaries, enabling us to address the following key questions:

- **What type of training information can be extracted from Document-Based VQA systems?** In Section 6.4 we show that, among the extractable answers, some are only present once in the training set. In some cases, extractable information is PII.
- **Can we distinguish between extractable answers arising from generalization and memorization?** In Section 6.4, we propose an efficient technique to attribute extractability to either memorization or generalization, and find that each phenomenon is responsible for some of the data we extract.
- **How do different modalities, contextual information and training conditions influence extractability?** In Section 6.5, we highlight two key factors that favour extractability: (low) image resolution at training time, and access to the exact training question. In contrast, we find that access to partial information about training images is less important for extractability.
- **Are there effective countermeasures?** In Section 6.6, we evaluate multiple heuristic defenses. We show that training a model to *abstain* from responding when the answer is not visually present in an input effectively mitigates extraction of PIIs.

6.2 Related Work

The concerning phenomenon we observe in Figure 6.1.1 can be seen as an extension to the VQA setting of the notion of training data *extraction* that has been observed in generative models for text [Carlini et al., 2021, 2023a, Kandpal et al., 2022] and images [Carlini et al., 2023b, Somepalli et al., 2023b]. These works primarily focus on showcasing the ability to extract near-exact copies of entire training samples from a model. In contrast, we focus on *partial* extraction of information from a VQA model and aim to distinguish between extraction attempts that succeed due to the memorization or generalization capabilities of the considered models. To provide context for our definitions and experimental setup, we start with a concise overview of relevant literature.

Training data extraction from generative models. Large Language Models (LLMs) can memorize and regurgitate training data [Carlini et al., 2021, 2023a, Chen et al., 2020a], even when no overfitting occurs (on average) [Tirumala et al., 2022].

Similarly, text-to-image generators like Stable Diffusion can reproduce training data when prompted with captions seen during training [Somepalli et al., 2023a,b, Carlini et al., 2023b]. For both text and image generators, the ability to extract a sample appears to depend heavily on the number of *duplicates* of that sample in the training set [Carlini et al., 2023a], even though some uniquely-occurring samples can also be extracted [Carlini et al., 2021].

While no prior work has (to our knowledge) studied whether private training samples can be extracted from VQA systems, some studies have shown that language models can learn to infer sensitive information such as gender or nationality of a person from other contextual clues or distributional shortcuts [Plant et al., 2022], and that VQA systems can memorize information shared across many training samples [Tito et al., 2023]. These works thus exploit the model's legitimate generalization properties rather than the memorization notion we analyse in this work. (For further discussion about distributional shortcuts, refer to Section E.5.2).

Defining memorization. Disentangling memorization and generalization is a challenging task. A widely accepted definition is the *counterfactual* notion proposed by Feldman [2019], which defines memorization as the difference in performance of a model on some sample, comparing the cases in which a sample is in the training set or not. Unfortunately, empirically measuring this counterfactual score is expensive, as it requires training a large number of models, including and excluding the training sample in question [Lukasik et al., 2023, Feldman and Zhang, 2020, Zhang et al., 2021b]. In our paper, we follow a more efficient heuristic adopted by prior works, where counterfactual memorization is estimated by comparing the performance of just two models, one trained on a dataset containing the considered sample and one not containing it [Carlini et al., 2021, Guo et al., 2023].

6.3 Experimental Setting

Document-based visual question answering. Given an input image representing a document I and a question about its content Q whose correct answer is a, the goal of a Document-Based VQA model f is to produce an answer $\hat{a} = f(I, Q)$ such that $\hat{a} = a$. This is done by training the model on a dataset $\mathcal{D}^{tr} = \{(I_i, Q_i, a_i)\}_{i=1}^{N}$ to maximize the likelihood of the correct response a_i given the input image-question pair (I_i, Q_i). To simplify notation and improve readability, unless referring to specific samples is crucial for clarity, we often suppress the sample index i. For a thorough literature review about these systems, refer to Section E.5.1.

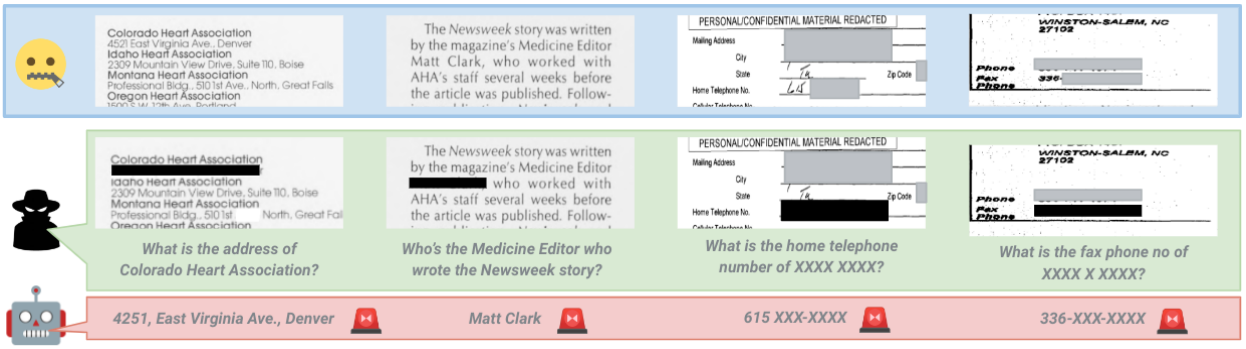


Figure 6.3.1: Four examples of Personally Identifying Information (PII) extractable by Donut (first two samples from left) and Pix2Struct-L (last two samples from right). A malicious user may query the model to reveal the PII by using a scan of the document from which the PII has been removed (black in the image). We anonymize personal information using gray boxes.

Dataset. We focus on the DocVQA dataset [Mathew et al., 2021], which contains images of real-world documents with diverse formats (e.g., letters, advertisements, reports, tickets etc.). We focus on this dataset for two reasons: (1) It is representative of privacy-sensitive tasks, and contains multiple forms of PII (see Section E.4); (2) it contains questions that are purely extractive [Mathew et al., 2022], meaning the answer is always explicitly written in the document. This makes it easier to automatically detect and eliminate parts of the input image that are necessary to answer a question, which forms the basis of our memorization test. This process would be harder for datasets that require abstract reasoning or external knowledge to answer questions.

Models. We consider three end-to-end state-of-the-art systems capable of directly processing the input image document, comprehending its contents, and producing a relevant response: **1) Donut** [Kim et al., 2022], among the first end-to-end Document-Based VQA systems that achieves high performance without using Optical Character Recognition (OCR). It is first pre-trained on synthetic documents, and then fine-tuned on DocVQA. **2) Pix2Struct** [Lee et al., 2023], a specialized model available in two versions: Base (282M parameters) and Large (1.3B parameters). It is pre-trained to perform semantic parsing of a 80M subset of the C4 corpus [Raffel et al., 2019] and then fine-tuned on DocVQA. **3) PaLI-3** [Chen et al., 2023b], a foundation model of 5B parameters, pre-trained on a web-scale multilingual image-text dataset, and fine-tuned on DocVQA.

Each of the models is fine-tuned on DocVQA using the training procedure outlined by the respective authors. To guard against overfitting, we perform early stopping based on the validation loss. This ensures that all the models we evaluate can generalize to previously unseen data, making them representative of practical deployed VQA systems. While training at the maximum resolution

possible is generally recommended to achieve better performance [Kim et al., 2022, Lee et al., 2023, Chen et al., 2023b], lower resolutions might also be adopted in some settings to accelerate training, especially for the largest models. We train each model multiple times with different image resolutions, to analyze the effect of this design choice on memorization.

Defining and Quantifying Extractability Drawing inspiration from [Carlini et al., 2023a], we introduce a definition of extractability that is suitable for the Document-Based VQA task.

Definition 1. ***Extractability of the answer*** *a* ***from a partial context*** (I^{-a}, Q) *Given a model* f *and a sample* $(I, Q, a) \in \mathcal{D}$*, we say it is an extractable sample if the correct answer a is obtained from the partial context* (I^{-a}, Q)*, i.e.,* $f(I^{-a}, Q) = a$*, where* I^{-a} *is a copy of the image I from which the correct answer a has been removed.*

We obtain the partial image I^{-a} by using the OCR outputs of Tesseract [Smith, 2007] included in the dataset: we identify the bounding boxes associated with all occurrences of the answer a within the document and replace it by a blank white box (we use black in the visualizations for readability). With this methodology, it is easy to identify some sensitive samples that are effectively extractable from the training set. In Figure 6.3.1, we show a few of the several cases in which it is possible to extract PII that is repeated *only once or twice* across the whole training set containing about 40K samples.

However, precisely quantifying the amount of extractable samples requires some care. Notably, due to occasional failures of the OCR system and the matching procedure to find the answer a within a document, some successful extractions are false positives (i.e., the correct answer is still in the input document). To account for this, we manually curate a smaller set of training samples (or *canaries*) $\mathcal{D}^C$. We select about 5400 canary answers (corresponding to about 1200 unique images) at random. We then manually inspect each of them and filter out all cases in which the answer removal procedure has failed. We also filter out samples for which the answer could be easily inferred from the context (e.g., predicting an intermediate value in a sequence of numbers, or predicting the total amount given a list of values), leaving us with 4654 samples, The obtained set of canaries contains a substantial amount of PIIs, whose distribution with respect to the most relevant classes of PIIs is reported in Figure E.4.1 in Section E.4.

6.4 Extractability and Memorization

In this section, we quantify the extent to which malicious users who are aware of the original training question and possess an incomplete copy of the training document can prompt the Document-based VQA systems to successfully retrieve the information they seek.

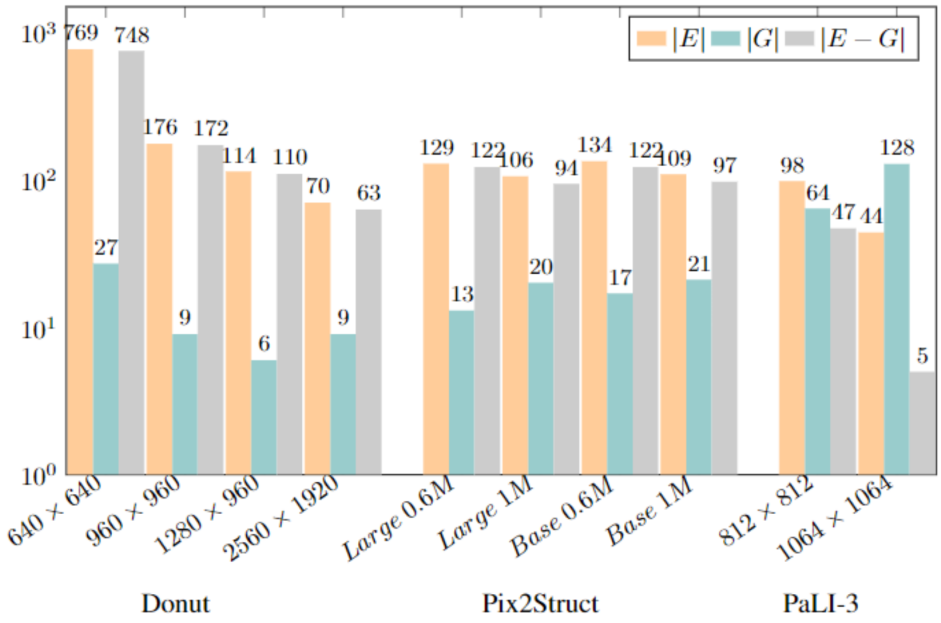


Figure 6.4.1: Extractability of answers for an attacker prompting the model with the original image from which the answer has been removed $I_i^{-a_i}$ and the original training question Q_i. The Y-axis is in logscale, therefore it overemphasizes the magnitued of lower values. PaLI-3 exhibits the lowest amount of extractable information in M.

Let us consider a model f that has been trained on $\mathcal{D}^{tr}$ including the canaries. We indicate with E the set of samples in $\mathcal{D}^C$ that are extractable from context for f. In Figure 6.4.1, we report the amount of extractable samples $|E|$, where $|.|$ indicates the cardinality of the set. As it can be seen, all the considered models extract a non-zero amount of answers from the canaries set. However, it is unclear whether the models are extracting some information because they have memorized it or because the partial context provided is already sufficient for a well-trained VQA system to respond correctly. For this reason, we propose a simple procedure to roughly estimate which samples in E are extractable due to memorization or generalization.

6.4.1 A Simple Baseline for Disentangling Memorization and Generalization

In order to determine whether the extractable answers are effectively memorized, in a similar vein to [Carlini et al., 2023a, Guo et al., 2023], we introduce a generalization baseline f_G. The idea is to compare the answers E extractable from f to the answers G that are extractable from

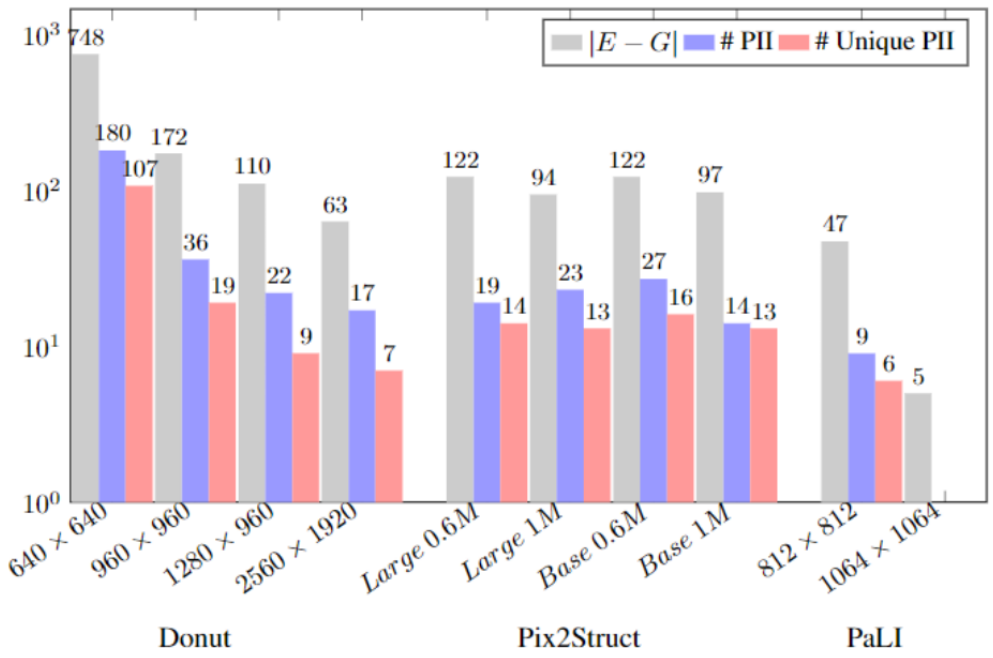


Figure 6.4.2: Amount of samples in M that are PII, and amount of samples that are unique PIIs when querying the model with (I^{-a}, Q).

a model f_G that has never seen $\mathcal{D}^C$ at training time (by removing it from the training set[1], i.e. $\mathcal{D}^{tr} - \mathcal{D}^C$), and which can therefore extract the correct answers due to legitimate generalization capabilities (or chance). If an answer is extractable from f but not from f_G, this suggests that the answer was memorized at training time, and cannot simply be recovered from context. We thus quantify the amount of extractable memorized information as the amount of answers extractable from f but not f_G: in other terms, $|M| = |E - G|$.

Result: In Figure 6.4.1 we report $|E|, |M|$ and $|G|$ for all the considered models. In Figure 6.4.2, we also report the amount of unique PIIs that are memorized. These PIIs mostly represent individuals names, sensitive locations (like travel destinations), and serial numbers of tickets or products. For both Donut and Pix2Struct, a substantial amount of examples extractable by f are not extractable by the generalization baseline and are likely memorized. In contrast, for PaLI-3 trained at a high resolution, most extractable answers appear due to generalization alone, and not memorization.

As shown in Figure 6.4.2, the highest resolution variants of Donut and Pix2Struct can extract PIIs and especially unique PIIs, but the highest resolution variant of PaLI-3 does not. From these results, we can identify two factors that have a strong impact on the amount of memorized samples:

1) Training resolution: Given a fixed model architecture, the resolution at which the model

[1]Notice that removing the canaries set from the training set does not yield a difference in generalization performance.

is trained is inversely proportional to the amount of memorized samples. Intuitively, the lower the resolution, the harder it is for a model to actually read the answers from the image and the easier it is for it to minimise the loss by memorization. For instance, while at the highest resolution for Donut $|M| = 63$, as the training resolution decreases, $|M|$ grows to 109, 168 and to an extremely high level of 756 for the lowest training resolution.

2) Pretraining: Manually inspecting the samples extractable by the generalization baseline, we observe that for Donut and Pix2Struct, these contain highly repeated answers (e.g., page, table and figure numbers) or frequently repeated names of organizations (e.g., ITC and AHA). For PaLI-3, we instead observe that, besides trivial answers like the ones extracted for Donut and Pix2Struct, the generalization baseline correctly responds to questions whose answer relies on general knowledge (e.g., the meaning of ambiguous acronyms that can be resolved considering the topic of the input document, properties of chemical substances or general geographical notions). This is attributable to the web-scale pretraining. The lower amount of samples in M may also indicate that a better pre-trained model may rely less on memorization even at relatively low training resolutions due to their better generalization abilities: indeed, of all the models, PaLI-3 produces the best generalization performance on the test set (87.6 ANLS compared to 76.6 and 67.5 of the best Pix2Struct and Donut variants, respectively).

6.4.2 Extractable Memorization and Simplicity Scores

The method proposed in the previous section may incorrectly identify some extractable answers as memorized due to the randomness of the training process. To show our attribution technique mostly identifies memorized samples, we leverage a modified version of the memorization and simplicity metrics developed in [Feldman, 2019, Zhang et al., 2021a].

Memorization and simplicity scores. Let $\mathcal{A}$ be stochastic training algorithm. For each sample $(I_i, Q_i, a_i) \in \mathcal{D}^C$, we would like to estimate the Memorization score [Feldman, 2019]:

$$\begin{aligned} \mathcal{M}(\mathcal{A}, \mathcal{D}^{tr}, i) = P_{f\sim\mathcal{A}(\mathcal{D}^{tr})}[f(I_i, Q_i) = a_i] - \\ P_{f\sim\mathcal{A}(\mathcal{D}^{tr-i})}[f(I_i, Q_i) = a_i] \end{aligned} \tag{6.1}$$

where $\mathcal{D}^{tr-i}$ indicates $\mathcal{D}^{tr}$ from which sample i has been removed. This score quantifies the difference between the probability that a model produces a correct prediction on a canary given the model has seen it at training time or not.

A score of 1 indicates the model can predict correctly on an input sample exclusively if it has seen it at training time. A score of 0 indicates that it has the same probability to produce a

correct prediction whether the sample was or not in the training set. Note that the memorization score says nothing about the model's accuracy on a sample (e.g., both a model that is always right or always wrong exhibits low memorization). To account for this [Zhang et al., 2021a] proposed a simplicity score $\mathcal{S}(\mathcal{A}, \mathcal{D}^{tr}, i)$ that sums the first and second terms of Equation equation 6.1. This allows to distinguish cases where a model fails to memorize a sample because it is hard to answer even when trained on (low simplicity), or because the answer is easy to produce even when not trained on (high simplicity).

Extractable memorization and simplicity. These two scores do not quite reflect the property we are interested: they inform us about the correctness of a model on an input sample (I, Q), and not about the ability to answer a question given a partial context (I^{-a}, Q). We thus adapt the memorization and simplicity scores accordingly, to consider the probability of a successful extraction:

$$\begin{aligned}\mathcal{M}_E(\mathcal{A}, \mathcal{D}^{tr}, i) = P_{f\sim\mathcal{A}(\mathcal{D}^{tr})}[f(I_i^{-a_i}, Q_i) = a_i] - \\ P_{f\sim\mathcal{A}(\mathcal{D}^{tr-i})}[f(I_i^{-a_i}, Q_i) = a_i]\end{aligned} \tag{6.2}$$

We call Equation equation 6.2 the Extractable Memorization score, and refer to the first term as the in-sample extractability and to the second as the out-sample extractability. Similarly, we define an Extractable Simplicity score $\mathcal{S}_E(\mathcal{A}, \mathcal{D}^{tr}, i)$ as the summation of the two terms.

Empirical estimation. Analogously to [Feldman, 2019, Lukasik et al., 2023], we compute empirical estimates $\hat{\mathcal{M}}_E$ and $\hat{\mathcal{S}}_E$ of $\mathcal{M}_E$ and $\mathcal{S}_E$ by training on random splits S^k of the training set that omit or maintain at random samples from the canary set $\mathcal{D}^C$. We produce a total of K splits, and define the indices of the splits containing a sample i as $K_{in} = \{k : (I_i, Q_i, a_i) \in S^k\}$ and $K_{out} = \{k : (I_i, Q_i, a_i) \notin S^k\}$. We then compute the in-sample and out-sample extractability scores as $\frac{1}{|K_{in}|}\sum_{k\in K_{in}} \mathbb{1}(a_i = f_{S^k}(I_i^{-a_i}, Q_i))$ and $\frac{1}{|K_{out}|}\sum_{k\in K_{out}} \mathbb{1}(a_i = f_{S^k}(I_i^{-a_i}, Q_i))$. Given that training Document-Based VQA systems is extremely expensive, we follow the sampling procedure in [Carlini et al., 2022b] in order to produce $K = 50$ splits such that each canary is in or out of a split exactly 25 times.

Experimental results. In Figure 6.4.3 we plot 2D histograms of the memorization and simplicity scores, $\hat{\mathcal{M}}_E$ and $\hat{\mathcal{S}}_E$. As it can be seen, the vast majority of the samples are not extractable at all, so we have $\hat{\mathcal{M}}_E = \hat{\mathcal{S}}_E = 0$. Some fraction of the training canaries are counterfactually extractable though, i.e., $\hat{\mathcal{M}}_E \gg 0$. To determine whether the technique proposed in Section 6.4.1 is actually identifying memorised samples, we now plot the Extractable Memorization and Simplicity scores

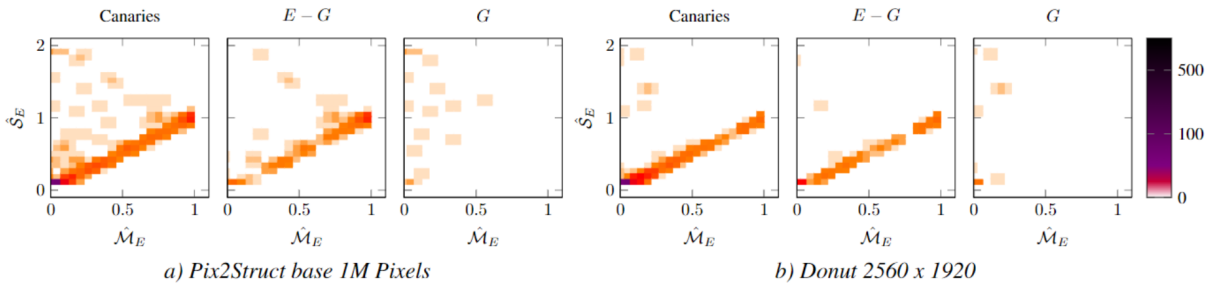


Figure 6.4.3: Distributions of the $\hat{\mathcal{M}}_E$ and $\hat{\mathcal{S}}_E$ scores for all the canaries, $E - G$ and G for both Pix2Struct base 1M Pixels (three panels on the left) and Donut 2560 x 1920 (three panels on the right). Samples in $E - G$ have high memorization scores, while samples in G do not.

of samples $E - G$ that were extractable only from the original model f, as well as the "control" samples G that were extractable by the generalization baseline f_G. As expected, samples in G have low memorization scores $\hat{\mathcal{M}}_E$: these answers can be extracted whether we train on them or not. In contrast, samples in $E - G$ have memorization scores $\hat{\mathcal{M}}_E$ that vary between 0 and 1. Most of the samples are close to the line $\hat{\mathcal{S}}_E = \hat{\mathcal{M}}_E$, indicating that the in-sample extractability is the only term contributing to $\hat{\mathcal{M}}_E$ (i.e., a model must see a sample at training time in order to extract it, and cannot extract it due to generalization only).

6.5 Ablations on the Extraction Context

So far, we studied the extractability of an answer a assuming knowledge of all other parts of an input. We now relax this assumption to both gain further insights into the factors influencing extractability, and, in some cases, to simulate more realistic attack scenarios in which perfect knowledge of the context (I^{-a}, Q) is not available. Indeed, while perfect knowledge of the context is unlikely in many cases, it is possible for an attacker to craft an approximation of the context (e.g., because the information they are seeking is contained in documents with a known or fixed structure, like driving licences or forms available online).

Before delving in the results, we point out that just like modifying the way a LLM is prompted can modify its output significantly, changing the way the VLMs are prompted changes which samples are extractable. For this reason, in few cases, the amount of extractable samples may increase with respect to the baseline scenario we considered so far, especially for cases in which the generalization baseline is weakened by the reduced information contained in the approximation of the context.

6.5.1 No Text in the Image

For LLMs, prior work has shown that prompting a model with the prefix of a memorized string is a reliable way of extracting data [Carlini et al., 2023a, Tirumala et al., 2022]. Yet, for Document-Based VQA systems it is unclear whether the models actually needs to read any surrounding text in a document in order to recall the answer. For this reason, we study the case in which *all* text is removed from the image I. If the model can still respond correctly, it indicates the model is relying on the question and non-textual features (e.g., layout, presence of icons or images etc.) in order to regurgitate the answer. This experiment also represents a practical threat model where the attacker knows the layout of a document (e.g., because it is a form available online or a document with a fixed structure like driving licences or ID cards) but has little to no knowledge about its contents.

Results: Figure 6.5.1 shows that in case of Donut and Pix2Struct, the absence of text in the image significantly reduces the ability of the model to return the correct answer. In case of Donut the amount of samples in M is 26. Pix2Struct shows a similar decrease from about 94 to 27. The amount of PIIs returned is also significantly reduced, and consisting mostly of highly repeated PIIs (more than 6 times). In the case of PaLI-3, we also observe the model responds correctly to answers requiring general knowledge (e.g., the name of chemical substances from their symbols contained in the questions, names of animal species portrayed in pictures contained in the document). The increase in the amount of extractable answers may be related to the fact that, when the extraction fails, a typical pattern is for the model to read another part of the document. When no text is present, it is easier for the model to retrieve the information from the general knowledge it acquired at pre-training time. For PaLI-3, no PII is extracted.

Reliance on surrounding text: The lack of any text in the document significantly reduces the ability to extract unique PIIs.

6.5.2 Imperfect Knowledge of the Training Question

To understand whether the model is memorizing an association between the exact question Q and answer a, we measure whether we can extract the answer when the question is paraphrased. We create paraphrases Q' of Q and extract the answers using (I^{-a}, Q'). To this end, we use PaLM2 [Anil et al., 2023] to create a paraphrased question for each canary question. An example of paraphrase is the following: if the question Q is"What is the address shown in the document?", then the paraphrase Q' can be "What is the street name and city shown in the document?". This experiment also reflects the setting in which the attacker does not know the exact phrasing of the training question Q and approximates it with their own words.

Results: Figure 6.5.1 shows that the number of extracted answers significantly drops, but is still non-negligible. For both Pix2Struct and Donut we observe several unique PIIs are extractable (e.g., names of individuals, serial numbers of tickets and travel destinations). The extractability increases in the case of PaLI-3, but is again related to questions probing general knowledge and reveal no PII.

Robustness to paraphrasing of Q: Uncertainty about the exact phrasing of a question that queries PII does not prevent extraction of sensitive information, but can reduce the amount of extractable samples.

6.5.3 Robustness to Image Perturbations

An attacker may be able to craft a document similar to the one originally used for training, but the scanning procedure naturally induces some small visual differences that may influence the extractability of the answers (e.g. brightness changes, small rotations or translations). For this reason we consider the case in which the original context I^{-a} is perturbed with augmentations that reflect plausible differences that may incur between the training and adversarially crafted document scans. For this purpose, we consider the following augmentations: 1) brightness change: we increase ($\times 1.3, \times 2$) or decrease ($\times 0.8, \times 0.5$) the brightness of the document; 2) small rotations: we randomly rotate by ± 5 or ± 10 degrees; 3) small translations: we randomly shift the image by ± 20 and ± 100 pixels along both axes.

Results: In Figure 6.5.1, we can see that brightness changes can indeed reduce the amount of extractable information, but the amount of extractable samples is still significantly high. In most cases, the stronger the change in brightness, the less the answer is extractable. However, a substantial amount of samples remains extractable, especially with respect to the context perturbations considered in the previous sections. Rotating or translating the image has a stronger adverse effect on the extractability of answers, indicating that spatial information plays a more important role for extractability than the intensity information. Notice, the amount of extractable samples under image perturbations is significantly larger than the amount extractable when the question is paraphrased, indicating that precise knowledge of the question Q is more important for an extraction attack than precise knowledge of the original scan I^{-a}. This also suggests that extractability is more likely to be triggered in the presence of the training question than in presence of the input image I^{-a}.

Robustness to Image Perturbations The amount of extractable samples is relatively robust to brightness perturbations and less to spatial transformations. An adversary does not need to reproduce a perfect copy of the original training image to extract the answer.

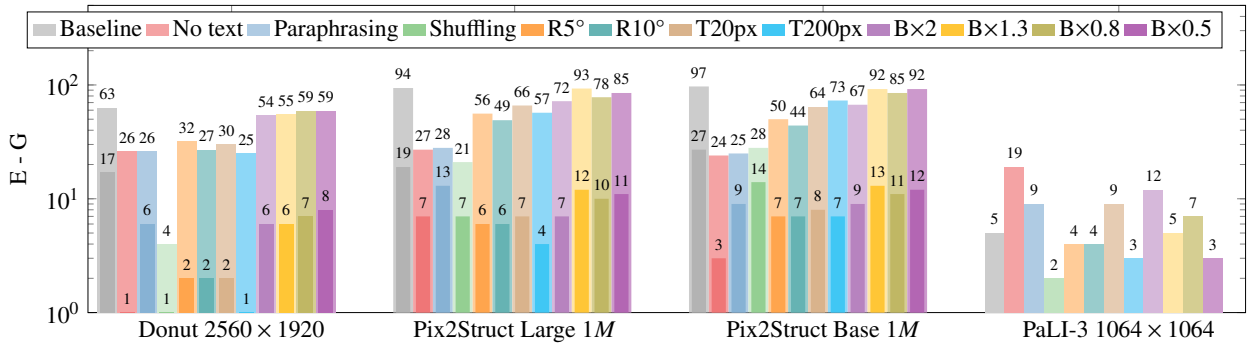


Figure 6.5.1: Extractability of answers when the context does not contain the text (No Text), the question is paraphrased (Paraphrasing), or not related to the image but the model still responds correctly (Shuffling), the image undergoes rotations (R5* and R10*), translations (T20px, T100px) and when brightness is changed by a mutliplicative factor (B×2, 1.3, 0.8 or 0.5). Darker colors indicate the number of PII samples that are extractable. Y-axis is in logscale. Across all deployable models, PaLI-3 exhibits the lowest amount of extractable information.

6.5.4 Permuting Modalities

Document-Based VQA systems contain both a visual component and a language component, each of which are fine-tuned on the training data. Extensive evidence has been provided that each of these components can memorise training data *in isolation* [Feldman, 2019, Lukasik et al., 2023, Carlini et al., 2022b, 2019]. Therefore an interesting question is whether it is possible for a multimodal model to extract the answers independently of one of the two input modalities. For this reason, we consider two experiments that randomise the relationship between the two input modalities.

Extractability based on questions only. At inference time, we feed the model a partial image with an unrelated question $(I_j^{-a_j}, Q_i)$, where $i \neq j$ and there is no training sample with question Q_i applied to image I_j, and the correct answer to question Q_i does not appear in the text of image I_j. This experiment evaluates the ability of the model to respond solely based on the question and reflects the case in which the attacker does not know the image I_i at all.[2]

Results: In the setting where we try to extract the original answer a_i, as visible in the Shuffling column in Figure 6.5.1, we can extract only 4 answers in case of Donut, and 21 in case of Pix2Struct. Among all the samples in M, we can also find some sensitive samples containing area codes, names of individuals and dates in which the documents were issued. The sensitive samples are also repeated only once or at most twice in the model's training set. While 2 answers can be extracted for PaLI-3, no PII was extracted.

[2]We have also tried replacing the input image with constant intensity value set to black, white or the average value of I_i. No answer was extractable in this case, perhaps because such images are too far out-of-distribution.

Δ ANLS / $\lvert M \rvert$	PR	AR	ITP	EB (Ours)
Donut	-3.4 / 38	-3.1 / 34	-12.5 / 26	**+1.2 / 2**
Pix2Struct-B	-2.9 / 40	-1.9 / 35	-12.9 / 28	**+3.4 / 0**
Pix2Struct-L	-2.6 / 37	-2.0 / 33	-13.8 / 25	**+2.1 / 0**
PaLI-3	-3.7 / 4	-3.2 / 3	-8.1 / 9	**+1.5 / 0**

Table 6.6.1: Variation of ANLS (utility metric for DocVQA) and amount of extractable samples in M for various countermeasures with respect to the standard training procedure.

Extractability based on images only. As in the previous experiment, we provide the model with a partial input image and an unrelated question that does not contain an answer within the image. We then measure whether we can extract an answer to one of the questions that was asked about this image during training. We find no extractable answers in this setting, which suggests that the question plays a more predominant role in the extraction.

Dependency of extractability on modalities In few cases, the model can leverage the language component alone to extract sensitive answers. If the training answers are not present in the image modality and the question was not seen at training time for a specific document, the image alone is not sufficient to extract any memorized answer.

6.6 Defenses

To conclude our study, we consider various mitigation strategies and measure their impact on memorization and generalization capabilities of the models (by computing the ANLS [Mathew et al., 2021] on a held-out test set):

- **Inference Time Paraphrasing (ITP)**, similar to [Somepalli et al., 2023a] we consider its effectiveness as a defense strategy.
- **Prepending/Appending a Random String (PR/AR)** Inspired by [Somepalli et al., 2023a], we perturb the question by prepending or appending a short 6-digit random string to the question.
- **Extraction Blocking (EB)** For each original sample (I, Q, a), we suggest adding to the training set a corresponding sample $(I^{-a}, Q, \texttt{'ANSWER NOT PRESENT'})$. This approach is similar in spirit to the intuition behind the V-CSS part of the algorithm proposed in [Chen et al., 2020a] to improve the grounding of VQA systems.

Results: We observe that although ITP and PR/AR can reduce the amount of extractable information, they also yield a substantial drop in ANLS on a held-out validation set. Therefore they can only be implemented as mitigation strategies if the practitioners are willing to pay a cost

in terms of performance. On the other hand, we observe EB to be extremely effective, reducing to 0 the amount of extractable samples for most models. Furthermore, although we apply the technique by augmenting the original training set using the context (I^{-a}, Q), it is also generalizes to adversaries that query the model with the approaches considered in Section 6.5 (see Table E.3.1), while producing an increase in the ANLS (in a similar way V-CSS does in [Chen et al., 2020a]).

6.7 Conclusion

In this study we have analysed the memorization abilities of three recent Document-Based VQA systems. We have shown these models can memorize information that is unique or sporadically repeated across the training set and it can be extracted when the model is prompted with incomplete context. We have introduced an extension of the Counterfactual Memorization and Simplicity scores that reveals that the memorized information identified by our attribution method is indeed also memorized according to these more computationally expensive scores. We have analysed the influence of the context on the extractability of samples, and studied the effectiveness of a few heuristic techniques, one of which results in a reduction of the amount of extractable samples and improves the test performance.

7
Conclusion

7.1 Summary

In this integrated thesis, we worked on solving some important challenges implied in the design of more trustworthy and privacy preserving deep learning systems. Here we summarise the main contributions of each of the presented works:

- **RegMixup: Mixup as a Regularizer Can Surprisingly Improve Accuracy and Out Distribution Robustness**. The paper finds Mixup reduces the performance at the task of Out-of-Distribution detection, and proposes a simple fix that improves the in-distribution and covariate-shift accuracy, while enhancing calibration and Out-of-Distribution detection abilities.

- **An Impartial Take to the CNN vs Transformer Robustness Contest**. The paper challenges previous literature and finds there's no strong evidence that Visual Transformers are more robust or reliable than CNNs through an extensive set of experiments, while finding they are subject to similar biases.

- **Not Just Pretty Pictures: Toward Interventional Data Augmentation Using Text-To-Image Generators**. The paper studies how modern T2I generators can be used to approximate interventions. We find this technique to be extremely effective to generate data that can be used to train more robust classifiers.

- **Kessler: A machine learning library for spacecraft collision avoidance**. The paper proposes the first ML library providing a solution to the problem of reliably estimating the probability of collision of satellites using Bayesian Deep Learning and Probabilistic Programming tools.

- **PILLAR: How to make semi-private learning more effective**. The paper finds a simple and effective dimensionality reduction technique that allows to significantly increase the accuracy of semi-privately learned linear classifiers.

- **Extracting Training Data From Document-Based VQA Models**. The paper identifies a way to prompt Document-Based Visual Question Answering systems in order to extract important personal information contained in the training set. We study the factors affecting the extractability of this information and propose a simple countermeasure .

Our works all aim at either improving existing methods or furthering our understanding of existing methods and techniques in key areas of Uncertainty Estimation, Domain Generalization and Privacy Preserving Machine Learning.

Appendices

A

Appendix of Chapter 2

Contents

A.1 Experimental Details

A.1.1 Code-base

For fair comparisons, when training on C10 and C100, we developed our own code base for all the approaches (except SNGP, DUQ and AugMix) and performed an extensive hyperparameter search to obtain the strongest possible baselines.

We would like to highlight that it was not easy to make a few recent state-of-the-art approaches work in situations different from the ones they reported in their papers as these approaches mostly required non-trivial changes to the architectures and additional sensitive hyperparametes. We also observed that their performances did not easily translate to new situations. Below we highlight few of these issues we faced and the measures we took for comparisons.

For DUQ, the original paper did not perform large scale experiments similar to ours. Unfortunately, we could not manage to make their code work on C100 as the training procedure seemed to be *unstable*. For this reason, *wherever possible*, we borrowed the numbers for DUQ from the SNGP paper. Please note that the authors of SNGP performed non-trivial modifications to the original DUQ methodology to make it work on C100.

For SNGP, we used the publicly available code following exactly the same procedure as mentioned in their original paper. The code *diverges slightly* from the procedure described in their paper, hence the slight differences in the performance. The only modification we performed to the official code-base was to make the inference procedure consistent with the one described in the paper: indeed, in their code they implement a mean-field approximation to estimate the predictive distribution [Lu et al., 2020], while in their paper they use Monte Carlo Integration with a number of samples equal to the number of members in the ensembles they use as a baseline, which provides better calibration. The rationale is that we could not find an obvious way to tune the mean-field approximation hyperparameters to improve at the same time both the calibration and OOD detection performance (indeed, *the mean-field approximation imposes a trade-off between calibration and OOD detection performance*). Additionally, since the standard KFAC-LLLA uses the same Monte Carlo Integration procedure, we opted for the latter for a fair comparison. For the SNGP RN50 experiments, we tried running the official implementation on C10 and C100, but we could not make SNGP converge to SOTA accuracy values. The authors of SNGP did not provide experiment results on C10 and C100 on RN50. Hence we decided *not to report* these experiments for SNGP.

For the KFAC-LLLA we leverage the official repository[1] [Hobbhahn et al., 2021] and the Backpack library [Dangel et al., 2020] for the computation of the Kronecker-Factored Hessian.

[1] `https://github.com/19219181113/LB_for_BNNs`

For AugMix, we used their code base and the exact training procedure. AugMix seems to be sensitive to hyperparameters of the training procedure as we could could not get the considered architectures to converge to acceptable accuracy levels under the training regime we used for **all** other baselines. Even with the recipes provided in the AugMix paper, we could not get it to converge to competitive accuracy levels when using RN50 on C10 and C100 hence we decided *not to report* these experiments for AugMix.

A.1.2 Optimization

For **C10 and C100** training, we use SGD with Nesterov momentum 0.9 for 350 epochs and a weight decay of 5×10^{-4}. For WRN, we apply a dropout $p = 0.1$ at train time. For all our experiments we set the batch size to 128[2]. At training time, we apply standard augmentations random crop and `horizontal flip` similar to [Liu et al., 2020a]). The data is appropriately normalized before being fed to the network both at train and test time.

For **ImageNet-1K** training, we use SGD with momentum for 100 epochs, learning rate 0.1, cosine learning scheduler, weight decay of 1×10^{-4}, batch size 128 and image size 224×224. We use `color jitter`, `random horizontal flip` and `random crop` for augmentation. We leverage the `timm` library for training [Wightman, 2019b] all the considered methods with Automatic Mixed Precision to accelerate the training.

A.1.3 Hyperparameters

- For DNN-SN and DNN-SRN the spectral norm clamping factor (maximum spectral norm of each linear mapping) $c \in \{0.5, 0.75, 1.0\}$ and the target of stable rank $r \in \{0.3, 0.5, 0.7, 0.9\}$ (as $r = 1$ for SRN is the same as applying SN with $c = 1.0$). Refer to miyato2018spectral and [Sanyal et al., 2020] for details about these hyperparameters.

- For Mixup, we consider a wide range of Beta distribution hyperparameter $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5, 1, 5, 10, 20\}$.

- For RegMixup we consider the Beta distribution hyperparameters to be $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5, 1, 5, 10, 15, 20, 30\}$.

- For KFAC-LLLA we take 1000 samples from the distribution. Although the number might seem quite high, we could not notice significant improvements using a lower number

[2]For SNGP and DUQ, we use the hyperparameters suggested in their original papers.

of samples. We tuned the prior variance σ_0 needed for the computation of the Laplace approximation minimising the ECE on the validation set. We also tried using the theoretical value $\sigma_0 = 1/\tau$ [Kristiadi et al., 2020], where τ represents the weight decay, but it produced inferior results with respect to our cross-validation procedure.

- For Deep Ensembles we use 5 members.
- When temperature scaling is applied, the temperature T is tuned on the validation set, minimising the ECE (we considered values ranging from 0.1 to 10, with a step size of 0.001). For Deep Ensembles, we first compute the mean of the logits, then scale it by the temperature parameter before passing it through the softmax.

All the cross-validated hyperparameters are reported in Table A.1.1. The cross-validation is performed with stratified-sampling on a 90/10 split of the training set to maximise accuracy[3] on C10 and C100. For ImageNet we split the test set using the same proportion. It is important to observe that:

- Cross-validating hyperparameters based solely on the ECE can prefer models with lower accuracy but better calibration. However, a method improving calibration should avoid degrading accuracy.
- Hyperparameters should not be cross-validated based on CS experiments and OOD detection metrics as they these datasets should be unknown during the training and hyperparameter selection procedure as well.

A.2 Existing Uncertainty Measures

There are various uncertainty measures and there is no clear understanding on which one would be more reliable. In our experiments we considered the following metrics and chose the one best suited for each method in order to create the strongest possible baselines. Let K denote the number of classes, $\mathbf{p}_i$ the probability of i-th class, and $\mathbf{s}_i$ the `logit` of i-th class. Then, these uncertainty measures can be defined as:

- **Entropy**: $H(\mathbf{p}(\mathbf{x})) = -\sum_{i=1}^{K} \mathbf{p}_i \log \mathbf{p}_i$.
- **Dempster-Shafer** [Sensoy et al., 2018]: $\texttt{DS}(\mathbf{x}) = {}^{K}/_{K+\sum_{i=1}^{K} \exp(\mathbf{s}_i)}$.

[3]Except for the σ_0 of the KFAC-LLLA, as we could not observe significant differences in Accuracy between hyperparameters optimising the accuracy and ECE

Training Set Architecture	Hyp	C10		C100		ImageNet
		WRN	R50	WRN	R50	
DNN	T	1.32	1.51	1.33	1.42	1.19
DNN-SN	c	0.5	0.5	0.5	0.5	-
	T	1.42	1.51	1.21	1.42	-
DNN-SR	r	0.3	0.3	0.3	0.3	-
	T	1.33	1.41	1.22	1.42	-
DE	T	1.31	1.42	1.11	1.21	1.29
SNGP	T	1.41	-	1.52	-	-
Mixup	α	0.3	0.3	0.3	0.3	0.1
	T	0.73	0.82	1.09	1.21	1.06
RegMixup	α	20	20	10	10	10
	T	1.12	1.31	1.23	1.21	1.14
KFAC-LLLA	#samples	1000	1000	1000	1000	
	σ_0	1	0.6	4	0.1	-

Table A.1.1: Cross-validated hyperparameters. Note, T and σ_0 are cross-validated by minimizing the ECE. All other hyperparameters have been tuned to maximise the accuracy.

- **Energy**: $E(\mathbf{x}) = -\log \sum_{i=1}^{K} \exp(\mathbf{s}_i)$ (ignoring the temperature parameter). This metric was used in [Liu et al., 2020c] for OOD.
- **Maximum Probability Score**: $\texttt{MPS}(\mathbf{x}) = \max_i \mathbf{p}_i$.
- **Feature Space Density Estimation** (FSDE): Assuming that the features of each class follow a Gaussian distribution, there are several ways one can estimate the *belief* of a test sample belonging to in-distribution data and treat it as a measure of uncertainty. One such approach is to compute the Mahalanobis score $\arg\min_{i \in y} (\phi(\mathbf{x}) - \mu_i)^T \Sigma_i^{-1} (\phi(\mathbf{x}) - \mu_i)$, where μ_i and Σ_i are class-wise mean and the covariance matrices of the *train* data, and $\phi(\mathbf{x})$ is the feature vector.

In the main paper, we report the OOD detection performance using the `DS` *score (as it provided slightly improved performance in most cases), except when it damages the performance of a method (e.g. Mixup) or when it does not yield improvements (e.g. KFAC-LLLA). In these situations we use the entropy as the uncertainty measure.*

Remarks regarding various metrics: We would like to highlight a few important observations that we made regarding these metrics. **(1)** `DS` **and** E **are equivalent** as they are both decreasing functions of $\sum_{i=1}^{K} \exp(\mathbf{s}_i)$, and since log does not modify the monotonicity, both will provide the

Methods	Clean AdaECE (↓)	CIFAR-10-C AdaECE (↓)	CIFAR 10.1 AdaECE (↓)	CIFAR 10.2 AdaECE (↓)	Methods	Clean AdaECE (↓)	CIFAR-100-C AdaECE (↓)
C10 R50					**C100 R50**		
DNN	3.02	17.30	7.39	12.24	DNN	9.47	25.17
Mixup	2.87	**11.35**	**4.05**	**7.72**	Mixup	7.47	21.52
RegMixup (**Ours**)	1.40	11.51	4.15	8.23	RegMixup (**Ours**)	3.92	13.68
DNN-SN	2.90	17.40	7.68	12.30	DNN-SN	9.44	25.04
DNN-SRN	2.82	17.17	7.57	11.97	DNN-SRN	9.59	25.50
KFAC-LLLA	**0.76**	11.52	6.20	9.17	KFAC-LLLA	**1.49**	**12.18**
DE (5×)	2.10	13.99	6.23	10.33	DE (5×)	6.50	19.76
C10 WRN					**C100 WRN**		
DNN	2.27	15.92	6.00	11.00	DNN	5.30	17.38
Mixup	2.23	**7.93**	7.22	**6.58**	Mixup	3.60	16.54
RegMixup (**Ours**)	**0.67**	8.36	**3.02**	7.03	RegMixup (**Ours**)	2.47	10.49
DNN-SN	2.21	15.55	5.49	10.82	DNN-SN	4.97	16.35
DNN-SRN	2.23	15.11	5.47	10.36	DNN-SRN	5.05	15.71
SNGP	1.51	11.33	5.59	10.85	SNGP	5.65	10.89
KFAC-LLLA	1.12	11.67	3.87	9.98	KFAC-LLLA	**2.30**	**8.99**
AugMix	1.89	**5.77**	4.10	9.61	AugMix	5.23	13.67
DE (5×)	1.74	13.52	4.33	9.44	DE (5×)	3.92	13.47

Table A.3.1: CIFAR calibration performance (%) without temperature scaling

same ordering of a set of samples. Hence, will give the same AUROC values. **(2)** We observed DS and H to perform similarly to each other except in a few situations where DS provided slightly better results. **(3)** MPS, in many situations, was slightly worse. **(4)** We found Gaussian assumption based density estimation to be **unreliable**. Though it provided extremely competitive results for C10 experiments, sometimes slightly better than the DS based scores, it performed very poorly on C100. We found this score to be highly unstable as it involves large matrix inversions. We applied the well-known tricks such as perturbing the diagonal elements and the low-rank approximation with high variance-ratio, but the results were sensitive to such stabilization and there is no clear way to cross-validate these hyperparameters.

A.3 Calibration Metrics without Temperature Scaling

For completeness, we report the calibration metrics over all the methods and considered datasets without the temperature scaling [Guo et al., 2017] in Tables A.3.1 and A.3.2. Details about the cross-validation procedure used when temperature scaling is applied is provided in Section A.1.1.

	IND	Covariate Shift			
	ImageNet-1K (Test)	**ImageNet-R**	**ImageNet-A**	**ImageNet-V2**	**ImageNet-Sk**
	AdaECE (↓)	**AdaECE** (↓)	**AdaECE** (↓)	**AdaECE** (↓)	**AdaECE** (↓)
DNN	4.90	20.48	52.30	9.58	22.94
Mixup	**2.28**	**14.70**	47.41	6.46	**18.26**
RegMixup (**our**)	3.06	17.42	**45.65**	7.34	20.85
AugMix	4.28	19.13	51.35	**3.94**	21.25
DE (5×)	3.61	17.32	51.64	7.94	19.35

Table A.3.2: ImageNet calibration performance (%) without temperature scaling.

A.4 Bayesian at Test Time: Last Layer Laplace Approximation

A structural problem of using MLE logistic regression is that the produced uncertainties depend on the decision boundary. On the other hand, replacing the MLE logistic regression with a Bayesian logistic regression and estimating the predictive posterior employing a Laplace approximation allows to produce better uncertainties [Kristiadi et al., 2020]. However, a Bayesian training either requires a modification in the architecture [Liu et al., 2020a] or makes the inference procedure very expensive [Kingma et al., 2015, Gal and Ghahramani, 2016a]. Since the objective is to utilize the standard MLE training of neural networks, the idea of Kronecker-Factored Last Layer Laplace Approximation [Kristiadi et al., 2020] is making the network **Bayesian at test time** with almost no additional cost.

Let $\mathbf{w}$ be the parameters of the of the last layer of a neural network, then we seek to obtain the posterior only over $\mathbf{w}$. Let $p(\mathbf{w}|\mathbf{x})$ be the posterior, then the predictive distribution can be written as:

$$p(y = k|\mathbf{x}, \mathcal{D}) = \int \texttt{softmax}(\mathbf{s}_k) p(\mathbf{w}|\mathcal{D}) d\mathbf{w}, \tag{A.1}$$

where, $\mathbf{s}$ is the logit vector and $\texttt{softmax}(\mathbf{s}_k)$ is the k-th index of the $\texttt{softmax}$ output of the network.

The Laplace approximation assumes that the posterior $p(\mathbf{s}|\mathcal{D}) \sim \mathcal{N}(\mathbf{s}|\mu, \Sigma)$, where μ is a mode of the posterior $p(\mathbf{w}|\mathcal{D})$ (found via standard optimization algorithms for NNs) and Σ is the inverse of the Hessian $\mathbf{H}^{-1} = -(\nabla^2 \log p(\mathbf{w}|\mathcal{D})|_\mu)^{-1}$. For the formulations and definitions, including the variants with the terms associated to the bias, we refer to [Kristiadi et al., 2020].

For our experiments, we obtain Σ using the Kronecker-factored (KF) approximation [Ritter et al., 2018]. Broadly speaking, the KF approximation allows to reduce the computational complexity of computing the Hessian by factorizing the inverse of the Hessian as $\mathbf{H}^{-1} \approx \mathbf{V}^{-1} \otimes \mathbf{U}^{-1}$, then the covariance of the posterior evaluated at a point $\mathbf{x}$ takes following form $\Sigma = (\phi(\mathbf{x})^T \mathbf{V} \phi(\mathbf{x}))\mathbf{U}$. This procedure can be easily implemented using the Backpack library [Dangel et al., 2020] to compute $\mathbf{V}$

and $\mathbf{U}$ by performing a single pass over the training set after the end of the training, as detailed in the Section of [Kristiadi et al., 2020] and clearly exemplified in the code-base of [Hobbhahn et al., 2021].

Let Σ_k be the covariance matrix of the posterior over the last linear layer parameters for the k-th class obtained using the Laplace approximation around μ, then, given an input $\mathbf{x}$, we obtain $\sigma_k = \phi(\mathbf{x})^\top \Sigma_k \phi(\mathbf{x})$ representing the variance of k-th logit $\mathbf{s}_k$. Once we obtain the covariance matrix, the Monte Carlo approximation of the predictive distribution (equation (A.1)) is obtained as:

$$\tilde{p} = \frac{1}{m} \sum_{i=1}^{m} \texttt{softmax}(\mathbf{s}(i)), \tag{A.2}$$

where, m logit vectors $\mathbf{s}(i)$ are sampled from a distribution with mean $\mathbf{s}$ and a covariance matrix (depending on the approximation used). Lu et. al [Lu et al., 2020] showed that similar performance can be achieved via the mean-field approximation which provides an approximate closed form solution of the integration in equation (A.1) involving the re-scaling of the logits and then taking the softmax of the re-scaled logit. The re-scaling is defined as follows:

$$\tilde{\mathbf{s}}_k = \frac{\mathbf{s}_k}{\sqrt{1 + \lambda \sigma_k^2}} \tag{A.3}$$

Note, the scaling of the k-th logit depends on its variance (obtained using the Laplace approximation) and a hyperparameter λ. This approximation is efficient in the sense that it does not require multiple samples as required in the MC approximation (which can become expensive as the number of classes and samples grow). In our experiments, we use the MC approximation, since we could not find an obvious way to fine-tune λ. Additionally, we observe that the mean-field approximation imposes a trade-off between calibration and OOD detection performance. Increasing λ, indeed, flattens the softmax distribution and improves OOD detection scores; although, as a consequence, harms calibration by making the network underconfident.

A.5 Additional Insights: RegMixup encourages compact and separated clusters

Here we provide additional experiments to show that RegMixup encourages more compact and separated clusters in the feature space We use the well known Fisher criterion [Bishop, 2006, Chapter 4] to quantify the compactness and separatedness of the feature clusters.

Fisher Criterion: Let C_k denotes the indices of samples for k-th class. Then, the overall *within-class* covariance matrix is computed as $\mathbf{S}_W = \sum_{k=1}^{K} \mathbf{S}_k$, where $\mathbf{S}_k = \sum_{n \in C_k} (\phi(\mathbf{x}_n) - \mu_k)(\phi(\mathbf{x}_n) -$

$\mu_k)^\top, \mu_k = \sum_{n \in C_k} \frac{\phi(\mathbf{x}_n)}{N_k}$, and $\phi(\mathbf{x}_n)$ denote the feature vector. Similarly, the *between-class* covariance matrix can be computed as $\mathbf{S}_B = \sum_{k=1}^{K} N_k(\mu_k - \mu)(\mu_k - \mu)^\top$, where $\mu = \frac{1}{N}\sum_{k=1}^{K} N_k \mu_k$, and N_k is the number of samples in k-th class. Then, the Fisher criterion is defined as $\alpha = \texttt{trace}(\mathbf{S}_W^{-1}\mathbf{S}_B)$.

Note, α would be high when within-class covariance is small and between-class covariance is high, thus, *a high value of α is desirable*. In Figure A.5.1, we compute α over the C10 dataset with varying degrees of domain-shift. As the amount of corruption increases, α gradually decreases for all the models. However, *RegMixup consistently provides the best α* in most cases.

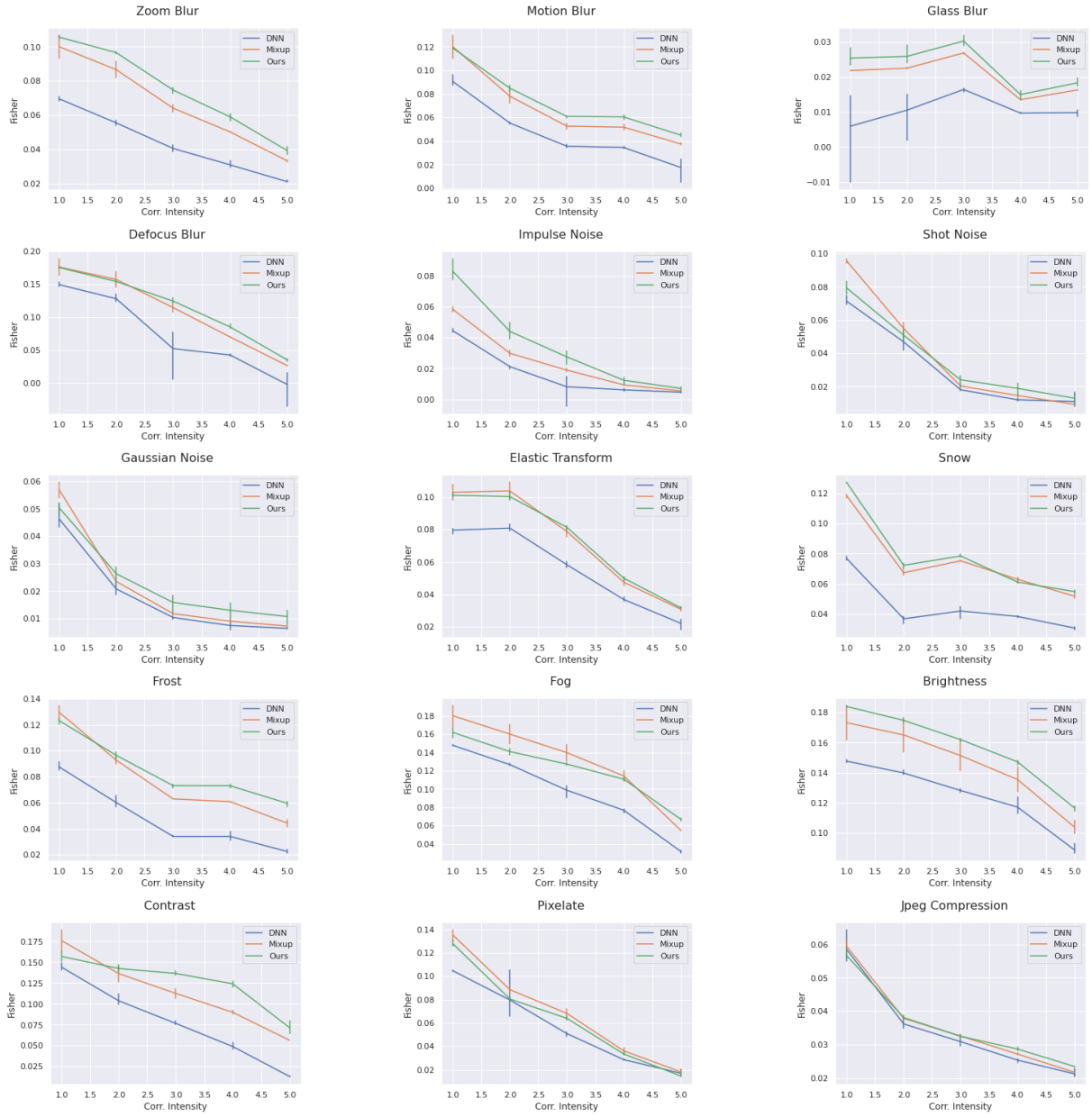


Figure A.5.1: Fisher criterion for all the corruptions and intensity values of CIFAR-10-C (WRN28-10).

B

Appendix of Chapter 3

Contents

	Clean Data			Domain-Shift												OOD
	ImageNet-1K (Test)			ImageNet-R			ImageNet-A			ImageNet-V2			ImageNet-Sk			ImageNet-O
	Acc (↑)	ECE (↓)	AdaECE (↓)	Acc (↑)	ECE (↓)	AdaECE (↓)	Acc (↑)	ECE (↓)	AdaECE (↓)	Acc (↑)	ECE (↓)	AdaECE (↓)	Acc (↑)	ECE (↓)	AdaECE (↓)	AUROC (↑)
BiT-R50x1	80.05	1.58	1.76	38.98	10.01	10.01	26.89	19.73	19.67	67.98	1.75	1.74	24.72	18.39	18.39	67.01
BiT-R50x3	83.59	2.65	2.51	47.25	8.51	8.51	46.72	11.66	11.63	72.36	6.30	6.08	32.81	19.00	19.00	77.99
BiT-R101x1	82.04	1.16	1.06	43.65	7.49	7.49	38.32	15.79	15.72	70.97	4.33	4.29	29.10	18.28	18.28	73.62
BiT-R101x3	84.19	3.78	3.72	50.14	9.10	9.10	53.12	10.90	10.93	73.36	7.85	7.71	36.29	21.15	21.15	80.44
BiT-R152x2	84.17	2.96	2.71	51.02	8.51	8.51	52.97	10.37	10.13	73.46	6.30	6.12	36.96	19.22	19.22	80.72
BiT-R152x4	84.49	6.28	6.26	54.06	11.50	11.50	58.52	12.17	12.14	74.36	10.94	10.94	41.17	25.47	25.47	85.58
ConvNeXt-B	85.53	2.87	2.82	62.46	2.57	2.51	52.63	8.28	8.31	75.43	2.91	2.78	48.62	8.87	8.86	85.72
ConvNeXt-L	86.29	2.27	2.34	64.57	3.00	3.08	58.23	7.57	7.26	76.77	3.72	3.85	50.06	10.31	10.31	89.07
ConvNeXt-XL	86.58	2.40	2.29	66.01	2.92	2.90	61.11	7.54	7.21	77.20	4.00	4.24	52.67	11.15	11.15	90.04
ViT-B/16	77.85	1.39	1.38	43.09	5.28	5.28	23.31	23.51	23.51	65.94	4.67	4.53	18.33	12.74	12.74	79.93
ViT-L/16	84.33	1.72	1.70	61.75	2.88	2.88	46.36	12.55	12.39	74.15	5.52	5.43	46.21	10.56	10.56	90.63
Swin-B	84.81	8.52	8.52	59.81	2.11	2.14	49.88	8.57	8.40	75.07	5.11	5.06	45.43	7.50	7.50	83.94
Swin-L	85.95	5.65	5.65	64.44	2.29	2.19	58.96	6.82	6.83	76.49	3.24	3.02	49.06	8.73	8.72	87.66
Fine-tuned at resolution 384×384																
ConvNeXt-B-384	86.51	3.16	3.15	64.12	3.36	3.47	63.25	7.70	7.58	77.03	2.49	2.65	50.31	7.84	7.84	87.11
ConvNeXt-L-384	87.14	2.39	2.38	66.09	3.27	3.16	66.52	7.01	6.90	77.97	3.51	3.31	51.68	9.60	9.60	90.45
ConvNeXt-XL-384	87.45	2.37	2.49	67.24	3.22	3.35	69.59	7.28	7.29	78.34	3.03	2.87	53.80	8.69	8.67	91.12
ViT-B16-384	79.43	1.53	1.60	40.62	6.49	6.49	33.63	17.46	17.46	68.37	4.45	4.45	14.54	15.75	15.75	81.75
ViT-L16-384	85.80	2.09	1.93	63.26	3.31	3.31	63.07	6.11	5.86	76.47	5.29	5.25	46.10	12.38	12.38	92.42
SWIN-B-384	86.29	6.78	6.78	63.41	2.29	2.28	62.20	6.57	6.52	76.65	3.80	3.83	48.43	8.43	8.43	86.46
SWIN-L-384	87.01	6.58	6.58	66.40	3.40	3.50	67.92	7.37	7.29	77.51	3.89	3.79	50.29	7.62	7.62	89.25

Table B.1.1: Analogous of Tables 3.3.3 and Table 3.3.4 but using the prepocessing pipeline suggested suggested by the timm library for each model. The conclusions of the main paper do not change.

B.1 Additional experimental details

B.1.1 About the evaluation metrics

All metrics are reported in percentage terms. The out-of-distribution detection metrics leverage the Dempster-Shafer uncertainty metric [Sensoy et al., 2018], as we find it to be the most effective for the task. For the misclassification detectiontasks, we use the confidence score (i.e. the maximum probability of the softmax) as uncertainty metric, as we find it to be the most effective for the task.

B.1.2 The impact of the input preprocessing pipeline

For the results reported in the main paper, we apply the standard ImageNet-1K test pre-processing pipeline: we first rescale the image at resolution 256×256 then extract the center crop of 224×224 and normalise with respect to the mean and standard deviation of the training set.

It should be noticed that the timm library suggests using a different pre-processing pipeline for each method. We do not follow this procedure for the results in the main paper as fine-tuning the test pre-processing pipeline hyperparameters would require a cross-validation procedure to not overfit the test set and we want to have a fair comparison using the same evaluation procedure for all models. We report the results applying the timm-proposed preprocessing pipelines in Tables B.1.1 and B.1.2. All the conclusions drawn in the main paper about ConvNeXts, ViTs and SwinTranformers do not change. The only case in which altering the pipeline dramatically changes the performance is on BiT models. With respect to the performance with the default pre-processing pipeline, BiT models become:

	Clean Data	Domain-Shift			
	ImageNet-1K (Test)	**ImageNet-A**	**ImageNet-R**	**ImageNet-SK**	**ImageNet-V2**
	PRR (↑)				
BiT-R50x1	72.48	23.31	-25.84	56.68	68.08
BiT-R50x3	73.41	-19.39	-8.67	62.41	67.39
BiT-R101x1	74.04	16.70	-22.27	60.64	68.12
BiT-R101x3	73.39	15.32	**-8.64**	62.54	66.61
BiT-R152x2	73.24	48.97	-20.76	61.36	66.81
BiT-R152x4	71.82	23.89	-35.15	62.15	64.54
ConvNeXt-B	73.43	16.03	-39.91	67.48	69.84
ConvNeXt-L	73.48	40.56	-23.60	69.04	69.50
ConvNeXt-XL	**74.36**	35.96	-19.32	69.29	70.07
ViT-B16	74.12	**49.46**	-33.53	64.59	70.89
ViT-L16	76.24	5.92	-31.09	**69.70**	**72.61**
Swin-B	71.99	17.10	-16.70	63.98	67.61
Swin-L	72.70	-10.78	-25.43	63.83	68.91
	Fine-tuned at resolution 384×384				
ConvNeXt-B-384	74.06	36.79	-22.39	67.37	68.75
ConvNeXt-L-384	74.12	32.74	**-10.88**	68.47	69.16
ConvNeXt-XL-384	74.71	55.16	-12.21	69.05	70.27
ViT-B/16-384	74.35	46.47	-32.97	66.18	71.13
ViT-L/16-384	**76.89**	-9.48	-20.06	**69.41**	**72.76**
Swin-B-384	72.53	**69.93**	-42.21	63.73	67.58
Swin-L-384	71.73	27.26	-17.02	63.04	66.71

Table B.1.2: Analogous of Table 3.3.5 but using the preprocessing pipeline suggested by the timm library for each model. The conclusions of the main paper do not change.

- way more accurate on in-distribution data. For instance, BiT-R152×4's accuracy jumps from 78.16 to 84.49.

- way more accurate on covariate shifted inputs. Particularly remarkable is the improvement when exposed to ImageNet-A. For instance, the accuracy of BiT-R50×1 jumps from 10.97 to 38.98 (which renders the smallest BiT model better performing than ViT-B/16!). Similarly, larger capacity BiT models can outperform ViT-L/16 and BiT-R152×4 is as competitive as the top-performing transformer (Swin-L). It is important to recall that ImageNet-A samples were selected to produce low accuracy on ResNets. This selection bias obviously makes comparisons between ResNets and any other architecture unfair. However, already changing the pre-processing pipeline at test time is enough to significantly weaken the adversarial effectiveness of the selection process on ResNet inspired architectures. Similarly, on other data-shift datasets, BiTs become extremely more competitive, and can outperform or be almost comparable to smallest transformer variants in many cases.

- significantly better at out-of-distribution detection (e.g. the minimum gap between BiT models and ViT-L/16 passes from almost 11% to less than 7%)
- significantly more calibrated on both in-domain and covariate shifted inputs. (e.g. the ECE is approximately halved in most cases on in-distribution data)
- on in-domain misclassification detection, the performance significantly increases. Similarly it increases (in most cases) on ImageNet-R, ImageNet-SK and ImageNet-V2. On ImageNet-A the performance decreases. This is another interesting case in which the calibration and misclassification detection provide complementary information: while the calibration error decreases on ImageNet-A, the misclassification detection performance gets worse, indicating the problem of being overconfidently wrong becomes more pronounced.

It should also be noticed that variants fine-tuned at resolution 348×384 exist (see the lower parts of Table B.1.1 and Table B.1.2). These variants generally outperform the variants fine-tuned at lower resolution in terms of accuracy, but generally exhibit worse uncertainty properties. The final conclusions of our paper do not change when considering these variants. Since we could not find BiT checkpoints fine-tuned at this resolution in the timm library, we reported the performance for models fine-tuned at 224×224 to have a fair comparison.

B.1.3 The impact of pre-training

It would be interesting to study the robustness and reliability of models without pre-training on ImageNet-21K. Unfortunately, checkpoints training solely on ImageNet-1K are often not included in the timm library or in general not publicly available, mostly becauuse some of the models considered do not produce good performance if trained from scratch on ImageNet-1K. For completeness, we report the performance results on ConvNeXt-B/L and Swin-B in Tables B.1.3 and B.1.4 . Notice, in this case the out-of-distribution detection results are reported using the negative confidence score as a form of uncertainty, as we find it to be the most effective in this case.

As it can be seen in Table B.1.3, ConvNeXt-B is typically more accurate and better calibrated than Swin-B except on ImageNet-A and ImageNet-V2 (where Swin-L is more calibrated). Swin-B produces better out-of-distribution detection performance. As it can be seen in Table B.1.4, ConvNeXt-L outperforms all other models at misclassification detection except in 1 case. However, we cannot draw conclusions from these two tables given the lack of comparison with other strong Transformer architectures and CNNs.

What we can evaluate is the difference occuring between the case without and with pretraining:

	Clean Data			Domain-Shift												OOD
	ImageNet-1K (Test)			ImageNet-R			ImageNet-A			ImageNet-V2			ImageNet-Sk			ImageNet-O
	Acc (↑)	ECE (↓)	AdaECE (↓)	Acc (↑)	ECE (↓)	AdaECE (↓)	Acc (↑)	ECE (↓)	AdaECE (↓)	Acc (↑)	ECE (↓)	AdaECE (↓)	Acc (↑)	ECE (↓)	AdaECE (↓)	AUROC (↑)
ConvNeXt-B	83.73	3.33	3.43	51.72	8.18	8.14	35.79	22.55	22.51	73.69	5.55	6.30	38.27	22.78	22.78	62.64
ConvNeXt-L	84.16	3.86	3.95	53.93	8.50	8.47	40.54	21.42	21.40	74.01	5.74	6.47	40.14	23.40	23.40	62.68
Swin-B	83.08	5.08	5.01	47.20	8.72	8.71	34.39	20.43	20.45	72.10	5.29	4.99	32.62	22.83	22.83	64.01

Table B.1.3: Analogous of Table B.1.1, but checkpoints are not pre-trainined on ImageNet-21K.

	Clean Data	Domain-Shift			
	ImageNet-1K (Test)	ImageNet-A	ImageNet-R	ImageNet-SK	ImageNet-V2
	PRR (↑)				
ConvNeXt-B	70.45	-7.13	-11.43	65.16	65.31
ConvNeXt-L	69.71	36.21	-30.27	64.04	65.93
Swin-B	68.20	34.16	-2.43	59.58	63.18

Table B.1.4: Analogous of Table B.1.2, but checkpoints are not pre-trained on ImageNet-21K.

- in all cases, in-distribution accuracy is significantly improved by pre-training
- for ConvNeXt models, the lack of pre-training harms calibration on in-domain data and under covariate shift. For Swin-B, the lack of pretraining improves the calibration on in-distribution data, but harms it under covariate shift.
- the lack of pretraining significantly damages the out-of-distribution detection performance of all models
- the lack of pretraining harms the misclassification detection performance except in the case of ImageNet-R for ConvNeXt-B and Swin-B. Swin-B performance drops more significantly than ConvNeXt-L models without pretraining.

B.2 Further discussion about the practice of comparing models based on parameter count

In this section we provide additional examples explaining why the parameter count is not really representative of the ability of a model to capture better approximations of the function underlying the relationship between inputs and outputs that generalise better.

One might wonder whether ways to quantify this aspect of a model exist. For this reason, we resort to known complexity measures in the literature and show these are no better than parameter count for the purpose of comparing models belonging to different families of architectures, and advocate for the need of measures that allow to compare models independently of their kinship.

B.2.1 Additional examples of why parameter count is not a proxy for generalization

Firstly, it is important to observe that all the considered models have significantly more parameters than the number of training samples (even when considering ImageNet-21K as training set). Therefore, from a theoretical point of view, all the considered models can interpolate the training set. These models differ in the way the learning procedures can leverage the data and the available parameters to learn solutions that generalise better. How overparametrization is related to the extraordinary generalization properties of Neural Networks is still an open area of research, and out of the scope of this paper.

Secondly, consider the following examples:

- consider BiT-R152×2 and BiT-R152×4. It is evident that although the latter has about 4 times the number of parameters of the former, the performance improvements observed in our tables are often marginal. This implies the training procedure is not capable of leveraging the additional number of parameters to boost the performance. It could be interesting if future literature investigated how much BiT-R152×4 can be pruned before it looses its advantage over BiT-R152×2.

- consider ConvNeXt-B and BiT-R152×4. Although the first contains almost 10 times less parameters than the latter, and both rely on convolutional inductive biases, ConvNeXt-B significantly outperforms BiT-R152×4 almost always. This comparison shows that parameter count is not representative of the generalisation properties of a model even when comparing models sharing convolutional inductive biases. Several other design choices that are often neglected in existing literature comparing Transformers and CNNs (e.g. quantity and types of activations or normalization layers, kernel sizes, proportions between the block sizes etc.) can greatly influence the ability of a model to produce robust and reliable predictions.

B.2.2 Can complexity measures do better than parameter count?

A natural question that could arise from these observations is whether it is possible to find a measure that quantifies the generalization properties of a model as a function of the value of its parametrs. The value of the parameters will obviously depend on the training data, the training procedure and the architecture, as opposed to the parameter count metric. This is still an open area of research, and a recent study collected and compared the most popular complexity measures [Jiang* et al., 2020].

We now consider a couple of the most popular complexity measures and show how they cannot be used to compare the generalization properties of models belonging to different families, and therefore, for this purpose, are no more useful than parameter count. We consider the following two measures:

- Path-Norm [Neyshabur et al., 2017], defined as:

$$PN = \sum_i f_{w^2}(\mathbf{1})[i]$$

  where f_{w^2} represents a network whose parameters have been squared, $\mathbf{1}$ indicates an input (of adequate shape, in this case we apply the same shape of ImageNet inputs) for which each entry is set to 1, $f_{w^2}(.)[i]$ represents the logit associated to class i.

- (logarithm of) Spec-Fro [Neyshabur et al., 2018], defined as:

$$\log SF = \log \prod_{i=1}^{L} \|\mathbf{W}_i\|_2^2 \sum_{i=1}^{L} \text{srank}(\mathbf{W}_i)$$

  where L is the total number of layers in the network, $\mathbf{W}_i$ represents the weight matrix of the i-th layer, $\text{srank}(\mathbf{W}_i)$ represents the stable rank of $\mathbf{W}_i$, i.e. $\text{srank}(\mathbf{W}_i) = \|\mathbf{W}_i\|_F^2 / \|\mathbf{W}_i\|_2^2$. We take the logarithm for numerical stability reasons.

Consider Table B.2.1. It is clear that both metrics cannot be used to compare models belonging to different families (e.g. the BiT values of Path-Norm and Spec-Fro are evidently at another scale with respect to those of all the other models; also, no inter-family consistent sorting based on generalization on the in-domain test set seems to emerge). Also for a same architecture, the behaviour of the metrics is inconsistent when comparing models pre-trained on ImageNet-21K and then fine-tuned on ImageNet-1K with respect to those trained only on ImageNet-1K. For instance, in the case of ConvNeXt the metrics remain almost unchanged, while for Swin-B the change is dramatic. The metrics also produce inconsistent behaviours within a family, for instance they do not sort based on generalisation properties the ViT-B and L models with patch sizes 16 and 32. For these reasons, for the purposes of this analysis, these metrics are no more useful than the parameter count. Future research should address this issue.

B.3 Samples of the ImageNet-9 and Cue-Conflict dataset

In Figure B.3.1 and B.3.2 we show some samples from the ImageNet-9 mixed same and mixed random splits. In Figure B.3.3 we show some samples from the Cue-Conflict dataset.

	# params	Path-Norm	log-Spec-Fro
BiT-R50×1	25	71.90	101.179
BiT-R50×3	217	211.21	103.10
BiT-R101×1	44	75.43	197.44
BiT-R101×3	387	224.96	199.62
BiT-R152×2	232	151.50	295.13
BiT-R152×4	936	298.22	296.47
ConvNeXt-B	88	0.51	2.23
ConvNeXt-L	196	0.75	55.60
ConvNeXt-XL	348	-0.28	80.75
ViT-B/16	86	0.34	46.77
ViT-L/16	304	-0.44	118.72
ViT-B/32	88	0.17	46.20
ViT-L/32	306	0.86	118.17
Swin-B	87	0.04	34.11
Swin-L	195	-0.95	84.72
Trained on ImageNet-1K only			
ConvNeXt-B	88	0.50	2.22
ConvNeXt-L	196	0.76	55.60
Swin-B	87	-314.96	381.30

Table B.2.1: Path-Norm and Spec-Fro Complexity measures for each of the considered models (checkpoints pre-trained on ImageNet-21K and fine-tuned on ImageNet-1K, except for the bottom part of the table

Figure B.3.1: A few samples from the ImageNet-9 mixed same split, in which the foreground of a class is mixed with a background from the same class.

Figure B.3.2: A few samples from the ImageNet-9 mixed random split, in which the foreground of a class is mixed with a background from another class.

Figure B.3.3: A few samples from the Cue-Conflict dataset, in which style transfer is used to alter the textures of an image using images from other classes as style sources.

B.4 The AUROC is agnostic to data imbalance and positive class choice

We now provide some trivial proofs about the properties of the AUROC in the setting of out-of-distribution detection.

Consider a threshold-based binary classifier $f_T(H(x))$ that assigns label 1 (out-of-distribution) if $H(x)$ (an uncertainty measure computed over the input x) is above a threshold T, 0 otherwise(in-distribution). If the choice of the positive class changes (1 means confident prediction, 0 means uncertain prediction), one can still use a thresholded classifier (that we will call $f'_T(C(x))$) that assigns label 1 (in-distribution) if $C(x)$ (a confidence measure computed over the input x) is above a threshold T, 0 otherwise (out-of-distribution). It is important to observe that given a strictly monotonically decreasing function g, one can get an uncertainty measure from a confidence measure or viceversa via composition with g.

As typical for binary classification, let us define TP as the number of true positives, FN as the number of false negatives, FP as the number of false positives, TP as the number of true positives for classifier f_T. We recall that the total number of positive (negative) samples can be written as $P = TP + FN$ ($N = TN + FP$), respectively. We also recall that in the context of evaluating out-of-distribution detection, the sizes of the test in-distribution and out-of-distribution sets are fixed, so that P and N are constant (because the positive classed has been assigned to out-of-distribution), and also $k = P/N$ is constant. Let us add an apex to indicate the same quantities for classifier f'_T (i.e. if in-distribution samples are assigned the positive class)

Let us clearly define the AUROC. To plot the ROC curve, one varies the the classification threshold $T \in [0,1]$ and plots the corresponding $(FPR(T), TPR(T))$, the function is monotonically increasing (not strictly). Let us denote with TPR_i with $i = 0, 1, \ldots, M$ the ordered set of points

corresponding to a discontinuity in the stair function. Ordered means:

$$TPR_{i-1} < TPR_i < TPR_{i+1} \iff FPR(TPR_{i-1}) < FPR(TPR_i) < FPR(TPR_{i+1}) \tag{B.1}$$

By convention set $TPR_0 = 0$ and $TPR_M = 1$. For a lighter notation, let us denote $TPR_i = h_i$ and $FPR_i = b_i$. Then the area under the ROC curve is:

$$AUROC = \sum_{i=1}^{M} (b_i - b_{i-1}) h_i$$

.

Theorem 1. *The AUROC in the aforementioned out-of-distribution detection evaluation setting is invariant to the choice of the positive class.*

Proof. (Sketch) Assume we start from the classifier $f_T(H(x))$ without loss of generality. If the label of all the test samples x are flipped, if a strictly monotonically decreasing function g is applied to the uncertainty measures $H(x)$ to obtain $C(x) = g(H(x))$, one can apply the classifier $f'_T(C(x))$. Then the following relationships hold: (1) $TN = TP'$, (2)$FN = FP'$, (3)$FP = FN'$, (4)$TP = TN'$. These imply (5) $TPR = TP/(TP + FN) = TN'/(TN' + FP') = TNR' = 1 - FPR'$ and similarly (6) $FPR = FP/(FP + TN) = FN'/(FN' + TP') = FNR' = 1 - TPR'$. We refer to this set of relationships as the "mapping implied by the swap of the positive class". We recall that the ROC function is a staircase function and its integral is computed as the sum of the rectangles composing the stair. Given

$$AUROC = \sum_{i=1}^{M} (b_i - b_{i-1}) h_i$$

let's similarly denote

$$AUROC' = \sum_{i=1}^{M} (b'_i - b'_{i-1}) h'_i$$

after the swap of the positive class. Imposing the same ordering constraint in the (b', h') plane, the relationship we found before becomes $b'_i = 1 - h_{M-i}$ and $h'_i = 1 - b_{M-i}$. Replacing these values in $AUROC'$ we obtain

$$AUROC' = \sum_{i=1}^{M} (1 - h_{M-i} - (1 - h_{M-i+1}))(1 - b_{M-i}) = \sum_{i=1}^{M} (h_{M-i+1} - h_{M-i})(1 - b_{M-i}) = AUROC \tag{B.2}$$

The last equality is obvious with geometric reasoning: while the $AUROC$ partitions the ROC stair with vertical rectangles (one for each step) and summates the area of the rectangles obtained

this way, $AUROC'$ partitions the same ROC stair using horizontal lines (one for each step) and summates the area of these rectangles (which is obviously the same). This is further validated by reasoning geometrically on the mapping of the swap of variables: this mapping moves the origin of the original space to $(1,1)$, rotates the coordinate axes of 90° anti-clockwise and flips what used to be the x axis orientation.

A similar reasoning applies if starting from a classifier $f'_T(C(x))$. □

Theorem 2. *The AUROC in the aforementioned out-of-distribution detection evaluation setting is invariant to class rebalancing by sampling multiple times the minority class.*

Proof. (Sketch) Let us assume, without loss of generality, that $k = P/N > 1$ and for the sole purpose of making the aforementioned rebalancing strategy feasible k is integer[1]. Then TPR is unchanged ($TPR = TP/P = TP/(FP+FN)$, and the FPR is unchanged too ($FPR = kFP/(kFP+kTN)$). Therefore the AUROC is unchanged. A similar reasoning holds for $k = N/P > 1$ and k integer.

□

The latter theorem implicitly shows that the AUROC is unaffected by the class imbalance. Indeed, if a class imbalance exists, a strategy to rebalance the two classes exists, and both the imbalanced and rebalanced sets will have the same AUROC.

[1]The value of k can be rational, but then the rebalancing strategy should resample also samples form the non-minority class to equate the number of positive and negative samples, but a similar theorem can be proved with a few more passages

C
Appendix of Chapter 4

Contents

C.1 Experiment Implementation

C.1.1 General Setup

Due to the speed limitation of generative models[1], we pre-generate all the augmentation images. For each image in the training set, we randomly selected k text prompts from the templates (see Section C.10). In the Single Domain Generalization experiments, we choose $k = 3$ (for PACS and OfficeHome) and $k = 5$ (for NICO++ and DomainNet) prompts for each image (i.e., one prompt from each target domain), whereas for the weakening spurious correlation experiments, we choose $k = 4$ prompts for each image to randomize the correlation between the causal and spurious features. Then for each prompt **one** image will be generated and saved as a corresponding augmented version of the original image. At training time, for each training image in the batch, one of its augmented versions will be randomly selected from the k pre-generated intervened samples

On efficiency, we note that, given a dataset with N traing samples, the generated interventional data will have a size of $N \times k$, where k ranges from $3 \sim 5$ (depending on the experiment). As such, the number of generated samples is generally low (given N is often of a few thousands) with respect to the amount of samples generated by baseline augmentation techniques (which is $N \times e$ where e represents the number of training epochs, and typically[2] $e >> k$). Although baselines produce more augmentations of the same image, our technique requires fewer augmentations per image to attain superior performance as `SDEdit` can intentionally target specific types of interventions.

We report a few statistics about the training, validation and test set sizes as well as the number of classes for each dataset in Table C.8.2. We use the model checkpoint of the last epoch to measure the test accuracy. For the experiments, as typical in the literature, we use pre-trained models on ImageNet for the backbones. To reproduce the experiment, we make part of our implementation available in the following anonymous repository.

C.1.2 Single Domain Generalization

We set up the experiment under the standard Single Domain Generalization paradigm. For **PACS**, **Office-Home**, **NICO++**, and **DomainNet**, we train a model for each single domain and evaluate it on the remaining unseen domains to measure the test accuracy. For the first two datasets, we generate **three** augmented samples each one of them corresponds to one target domain. For the latter two, we similarly generate **five**. For all the datasets, we use an image size of

[1] Significant progress in generation speed has been performed from the first versions of Stable Diffusion to the one we have been using in this paper. Accelerating diffusion models is an active area of research

[2] E.g., in our experiments, $e = 50$.

224×224. The full experiment results with expanded test accuracy one each test domain for PACS/OfficeHome/NICO++/DomainNet are shown in Table C.1.1/Table C.1.3/Table C.1.5/ Table C.1.7 for ResNet-18; Table C.1.2/Table C.1.4 / Table C.1.6/Table C.1.8 for ResNet-50. The visualised comparison between traditional data augmentation and generative model-based image editing as well as comparison among different types of conditional generation strategy is shown in Figure C.1.1 and Figure C.1.2, respectively. An overall comparison between different editing techniques is also presented in Table C.1.9.

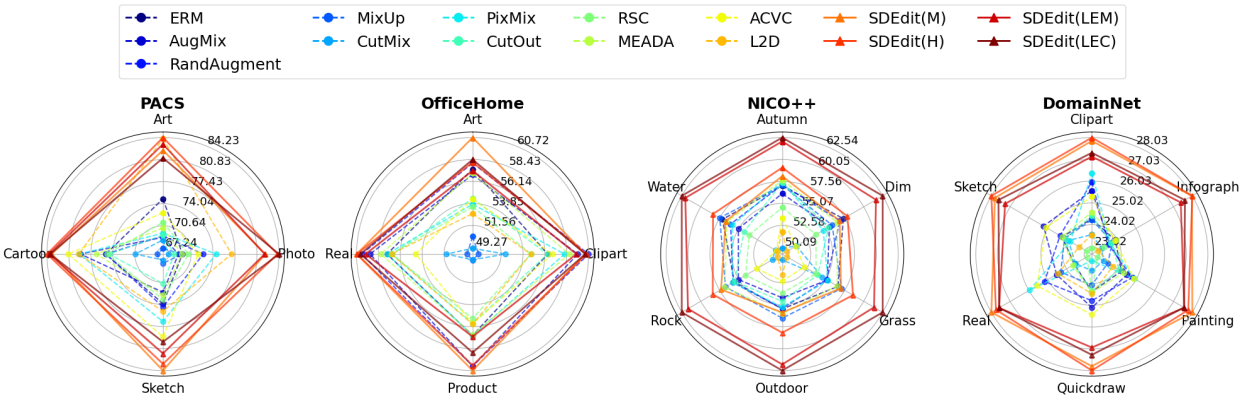


Figure C.1.1: Single Domain Generalization (SDG) Performance results in comparison with baselines (dashed lines) and OURS (solid lines) using ResNet-18.

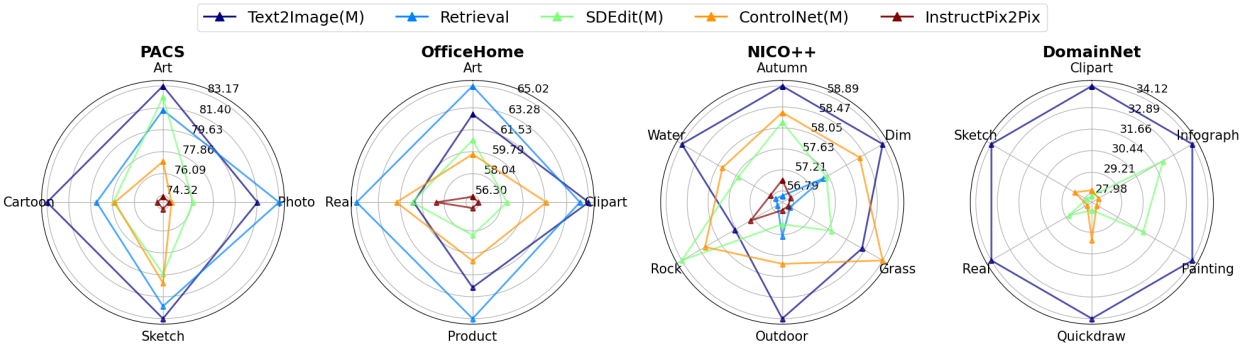


Figure C.1.2: Comparison Between Different Condition Generation Strategy using ResNet-18.

C.1.3 Effect of Accessing Multiple Source Domain

We also present further investigation on the effect of accessing multiple source domains, potentially including the target test domain. We experimented on three settings: (a) MDG: classifier trained on all but the target domain, which is the standard set-up for multi-domain generalization, where

Table C.1.1: Single Domain Generalization (SDG) PACS result with ResNet-18.

	Art	Photo	Sketch	Cartoon	Average
ERM	74.8	39.67	48.12	72.37	58.74
MixUp	67.14	39.57	33.24	63.27	50.81
CutMix	68.46	36.5	31.99	67.2	51.04
AugMix	68.88	38.75	43.89	76.86	57.09
RandAugment	69.07	44.48	49.36	72.31	58.8
CutOut	69.19	37.77	40.72	71.77	54.86
RSC	71.18	41.04	46.56	72.17	57.74
MEADA	70.32	39.55	44.94	74.03	57.21
PixMix	69.49	47.5	54.72	77.06	62.19
L2D	84.07	51.06	50.94	77.12	65.8
ACVC	72.65	43.33	60.35	78.98	63.83
VQGAN-CLIP(M)	78.09	54.38	53.78	77.76	66.00
Retrieval	81.22	75.49	83.36	83.24	80.83
SDEdit(M)	82.27	58.87	72.76	81.93	73.96
SDEdit(H)	84.23	61.7	70.31	82.74	74.75
SDEdit(LEC)	81.08	62.04	62.19	82.18	71.87
SDEdit(LEM)	83.21	58.74	66.45	82.37	72.69
Text2Image(LEM)	80.59	68.82	83.58	85.27	79.56
Text2Image(M)	83.17	71.31	87.42	87.12	82.26
ControlNet(M)	77.07	54.64	75.78	81.81	72.32
Textual Inversion	78.57	67.67	68.66	83.9	74.7
InstructPix2Pix	76.08	56.22	50.79	78.39	65.37

Table C.1.2: Single Domain Generalization (SDG) PACS result with ResNet-50. Columns are Single source domains; accuracies are the average test accuracy of the three remaining target domains when training using the indicated source domain (best accuracies are in bold).

	Art	Photo	Sketch	Cartoon	Average
ERM	74.44	48.78	50.89	73.74	61.96
MixUp	66.31	42.98	45.64	77.76	58.17
CutMix	72.53	40.03	44.72	76.72	58.5
AugMix	75.8	51.32	49.99	81.42	64.63
RandAugment	71.38	46.8	55.95	76.33	62.61
CutOut	76.67	42.69	48.93	75.2	60.87
RSC	73.15	53.47	51.11	80.58	64.58
MEADA	73.72	48.78	59.81	73.84	64.04
PixMix	77.33	55.58	52.42	83.15	67.12
L2D	77.33	58.41	58.14	81.7	68.89
ACVC	79.63	52.76	58.13	81.4	67.98
SDEdit(M) ×	81.21	57.54	80.60	84.76	76.03
SDEdit(O-M)	82.59	65.44	79.3	83.64	77.74
Retrieval	82.36	76.24	87.0	86.01	82.9
SDEdit(M)	82.67	62.94	73.78	86.33	76.43
SDEdit(H)	83.68	64.22	78.95	84.63	77.87
SDEdit(LEC)	82.69	59.48	77.76	85.57	76.38
SDEdit(LEM)	84.11	59.1	73.39	86.0	75.65
Text2Image(LEM)	82.11	68.08	87.55	87.71	81.36
Text2Image(M)	84.15	72.9	90.51	87.34	83.72
ControlNet(M)	75.65	56.47	81.83	84.01	74.49
Textual Inversion	76.15	68.36	76.66	87.89	77.27
InstructPix2Pix	76.87	54.47	54.7	81.25	66.82

multiple domains of source data are used for training and one unseen domain is used for testing; (b) All: classifier trained on all the domains, (c) Target: classifier trained only on the target domain. As shown in Table C.1.10, While ERM(Target/All) achieves almost perfect performance, this is expected as the training source domain includes the target test domain. Note that the accuracy in the table below is not directly comparable to SDG in the main paper since our main setting is Single Domain Generalization (SDG), where we have a single source domain for training, and the accuracy reported is the average test accuracy on multiple unseen test domains. However, here we have a single unseen target domain for testing. To provide a direct comparison between ERM and SDEdit, we experiment with SDEdit under the same setting in MDG with a minimal prompt. We demonstrate under MDG setting SDEdit also leads to significant performance improvement in all unseen test domains.

Table C.1.3: SDG OfficeHome result with ResNet-18.

	Art	Clipart	Product	Real	Average
ERM	57.43	50.83	48.9	58.68	53.96
MixUp	50.41	43.19	41.24	51.89	46.68
CutMix	49.17	46.15	41.2	53.64	47.54
AugMix	56.86	54.12	52.02	60.12	55.78
RandAugment	58.07	55.32	52.02	60.82	56.56
CutOut	54.36	50.79	47.68	58.24	52.77
RSC	53.51	48.98	47.16	58.3	51.99
MEADA	57.0	53.2	48.81	59.21	54.55
PixMix	53.77	52.68	48.91	58.68	53.51
L2D	52.79	48.97	47.75	58.31	51.95
ACVC	54.3	51.32	47.69	56.25	52.39
Retrieval	65.02	63.55	60.51	64.32	63.35
SDEdit(M)	60.72	54.95	52.47	61.26	57.35
SDEdit(H)	58.15	55.12	51.94	61.24	56.61
SDEdit(LEC)	58.43	54.96	50.64	60.93	56.24
SDEdit(LEM)	57.27	53.97	49.02	60.5	55.19
Text2Image(LEM)	59.8	61.88	55.13	58.24	58.76
Text2Image(M)	62.77	64.57	57.51	61.2	61.51
ControlNet(M)	59.59	59.58	54.94	62.14	59.06
InstructPix2Pix	56.2	51.58	49.85	59.96	54.4

Table C.1.4: SDG OfficeHome result with ResNet-50.

	Art	Clipart	Product	Real	Average
ERM	63.62	61.32	56.85	65.99	61.94
MixUp	63.46	59.2	54.97	64.21	60.46
CutMix	59.3	54.45	51.9	63.0	57.16
AugMix	63.99	61.11	58.88	66.44	62.6
RandAugment	64.92	61.38	59.34	66.42	63.02
CutOut	62.15	58.24	55.77	63.98	60.03
RSC	60.91	56.86	54.21	64.41	59.1
MEADA	64.48	61.6	57.34	64.89	62.08
PixMix	63.54	60.34	57.29	64.54	61.43
L2D	60.79	55.01	54.76	62.93	58.37
ACVC	62.33	57.76	55.59	64.02	59.92
Retrieval	71.15	71.46	67.21	70.73	70.14
SDEdit(M)	67.08	64.48	60.01	67.06	64.66
SDEdit(H)	64.55	63.05	58.99	66.48	63.27
SDEdit(LEC)	65.96	63.12	58.6	66.03	63.43
SDEdit(LEM)	64.86	62.88	58.46	66.37	63.14
Text2Image(LEM)	65.72	68.8	62.61	64.09	65.31
Text2Image(M)	68.6	71.11	63.83	66.98	67.63
ControlNet(M)	66.52	66.65	62.12	66.47	65.44
InstructPix2Pix	63.34	60.39	58.43	67.13	62.32

Table C.1.5: SDG NICO++ Result with ResNet-18.

	autumn	dim	grass	outdoor	rock	water	Average
ERM	57.07	60.95	62.4	61.82	58.52	65.04	60.97
RandAugment	57.19	60.51	61.23	61.77	58.67	64.08	60.57
AugMix	56.19	59.18	61.29	60.72	58.1	63.16	59.77
MixUp	57.15	59.52	62.77	62.71	59.47	65.36	61.16
CutOut	57.42	59.07	60.33	61.07	58.48	62.5	59.81
PixMix	57.55	58.38	61.36	61.62	58.68	63.85	60.24
RSC	54.61	57.47	60.14	60.25	57.32	61.86	58.61
ACVC	53.43	54.91	58.94	59.07	56.11	58.67	56.85
MEADA	57.7	60.17	62.32	62.27	59.53	64.52	61.09
L2D	51.88	53.79	57.15	58.48	53.92	58.55	55.63
Retrieval	56.69	60.31	61.58	62.51	58.06	63.57	60.45
SDEdit(M)	58.17	60.48	62.72	62.16	59.95	64.66	61.36
SDEdit(H)	59.14	61.48	63.95	64.14	60.84	66.15	62.62
SDEdit(LEC)	62.54	65.97	67.01	67.85	64.15	69.85	66.23
SDEdit(LEM)	62.11	65.11	66.12	67.25	63.49	69.43	65.59
Text2Image(M)	58.89	63.79	63.56	64.85	58.9	66.3	62.72
ControlNet(M)	58.36	62.43	64.13	63.29	59.49	65.12	62.14
InstructPix2Pix	57.01	58.36	61.53	61.76	58.59	63.73	60.16

C.2 Weaken Spurious Correlation

In the three considered cases, the reliance on the spurious correlation is measured as: (1) **ImageNet-9** (Background Bias): **Gap**, as defined in [Xiao et al., 2021], is the difference between the accuracies measured on the test sets `mixed same` and `mixed rand`. (2) **CCS Dataset** (Texture Bias): **Texture Bias**, as defined in [Geirhos et al., 2019], is the number of correct texture classifications over sum of the true positive texture and shape classifications. In the test CCS dataset, each image is synthesized with a texture and subject from different classes (i.e texture: elephant, class: cat). The true positive texture classification is the percentage of cases in which the model predicts the texture label correctly; similarly, true positive shape classification is the percentage of correctly classified shape labels. (3) **CelebA-sub** (Demographic Bias): **RandGap** and **FlipGap** represent the accuracy gap between **I.I.D** distribution to `rand` and `flip` respectively. The purpose is to measure the reliance on the spurious feature both for the average case (i.e., randomizing the spurious correlation in the

Table C.1.6: SDG NICO++ Result with ResNet-50.

	autumn	dim	grass	outdoor	rock	water	Average
ERM	66.74	70.37	72.05	71.3	66.58	72.64	69.95
RandAugment	67.23	71.43	70.81	70.62	66.47	72.71	69.88
AugMix	66.18	69.21	70.03	70.22	65.51	71.72	68.81
MixUp	67.6	70.3	72.47	72.26	67.12	74.01	70.63
CutMix	62.82	67.6	69.39	69.01	63.59	69.78	67.03
CutOut	66.76	69.34	70.13	70.13	66.67	72.33	69.23
PixMix	66.99	68.75	69.57	71.72	67.1	72.75	69.48
RSC	63.96	67.69	68.48	69.21	63.96	70.94	67.37
ACVC	63.74	67.48	67.73	68.71	63.89	69.95	66.92
MEADA	67.47	69.99	71.72	71.31	65.76	73.06	69.89
L2D	61.81	64.44	66.78	66.67	63.42	68.02	65.19
SDEdit(M) ×	67.90	71.42	72.61	72.32	67.10	73.79	70.90
Retrieval	67.39	72.16	71.53	71.83	66.19	73.45	70.42
SDEdit(M)	68.28	71.42	72.68	72.31	67.95	74.07	71.12
SDEdit(H)	69.13	72.13	73.64	73.33	68.46	75.03	71.95
SDEdit(LEC)	70.21	74.68	75.05	75.54	69.74	76.89	73.69
SDEdit(LEM)	70.36	74.4	74.48	75.09	69.83	77.47	73.61
Text2Image(M)	68.14	72.67	72.63	73.77	66.88	75.17	71.54
ControlNet(M)	67.69	72.64	73.19	72.84	67.73	74.92	71.5
InstructPix2Pix	66.86	70.35	72.02	71.95	67.13	73.31	70.27

test set) and in the worst case (i.e., the test set flips the spurious correlation). For all the three dataset each original image sample will have **four** pre-generated augmented samples. The comparison with all the baselines is in with ResNet-18 Table C.2.1, Table C.2.3, and Table C.2.2.

C.2.1 Additional Experiment on Cifar-10-C

We conduct further experiments with the Cifar-10-C dataset. We adopt a similar setting as RRSF, where we train on Cifar-10 and test on Cifar-10-C. In the evaluation, domain shifts were organised into distinct groups for clarity. The following classifications were made:

- Blurring Effects: defocus blur, gaussian blur, glass blur, motion blur, zoom blur
- Noise Variations: gaussian noise, impulse noise, shot noise, speckle noise

Table C.1.7: SDG DomainNet Result with ResNet-18.

	clipart	infograph	painting	quickdraw	real	sketch	Average
ACVC	25.31	19.86	25.23	8.0	27.49	26.84	22.12
AugMix	25.58	19.09	24.74	7.41	26.41	27.03	21.71
CutMix	23.56	17.83	23.0	4.33	25.36	25.04	19.85
CutOut	24.44	19.27	24.16	6.03	25.45	25.31	20.78
ERM	24.29	19.93	24.32	6.08	25.42	25.54	20.93
L2D	23.55	17.26	23.69	6.24	26.33	24.17	20.21
MEADA	24.6	20.06	24.5	6.17	25.52	25.56	21.07
MixUp	24.25	19.46	23.31	5.51	26.18	25.34	20.68
PixMix	26.39	19.18	25.28	3.49	27.9	24.89	21.19
RSC	22.92	18.21	22.52	6.11	24.72	23.59	19.68
RandAugment	25.99	18.88	25.12	6.83	27.08	25.71	21.60
SDEdit(H)	28.03	31.68	29.27	12.66	29.78	31.22	27.11
SDEdit(LEC)	27.33	30.56	28.96	11.35	29.6	30.65	26.41
SDEdit(LEM)	27.16	29.92	28.97	10.74	29.6	30.12	26.08
SDEdit(M)	27.88	31.62	29.57	12.3	30.03	30.94	27.06
Text2Image(M)	34.12	35.32	31.68	36.13	33.21	36.43	34.48
ControlNet(M)	28.19	23.40	27.59	18.81	29.28	31.62	26.48

- Compression Artifacts: JPEG compression
- Image Transformations: brightness, contrast, elastic transform, pixelate, saturate

As shown in Table C.2.4, SDEdit still demonstrate effectiveness under various parametric domain shift. Although the generalization performance is inferior to other parametric augmentation methods, it can be used in a combined manner.

C.2.2 Hyperparameters

Training Hyperparameter For all the other baselines, we use the value as proposed in their original papers or official implementation.

Generator Hyperparameter For the two types of generative models, we use the hyperparameters for each dataset as shown in Table C.2.6. Hyperparameters are tuned based on human judgement of few-shot image manipulation quality, without downstream task accuracy-based evaluation. Unspecified hyperparameters are set to their default value. For Stable Diffusion, We

Table C.1.8: SDG DomainNet Result with ResNet-50.

	clipart	infograph	painting	quickdraw	real	sketch	Average
ACVC	29.84	26.72	29.86	8.96	31.88	31.47	26.46
AugMix	30.04	26.1	29.48	8.92	31.07	31.61	26.20
CutMix	28.9	24.29	27.92	5.99	29.97	29.75	24.47
CutOut	28.98	25.8	28.71	6.6	29.96	29.36	24.90
ERM	29.06	27.07	28.87	6.92	29.85	29.8	25.26
L2D	28.15	23.85	28.61	7.12	31.25	29.53	24.75
MEADA	29.09	26.77	28.81	6.81	30.06	30.05	25.26
MixUp	29.34	26.89	29.17	6.46	30.66	30.42	25.50
PixMix	30.77	26.96	29.95	3.68	32.94	28.87	25.53
RSC	26.89	24.12	26.48	5.79	28.7	27.96	23.32
RandAugment	30.28	26.51	29.96	8.31	31.82	30.14	26.17
SDEdit(H)	32.89	37.95	33.99	15.89	34.45	35.77	31.82
SDEdit(LEC)	32.35	37.14	33.83	15.62	34.32	35.39	31.44
SDEdit(LEM)	31.84	36.75	33.72	13.88	34.19	35.24	30.94
SDEdit(M)	33.09	38.13	33.99	15.86	34.64	35.94	31.94
InstructPix2Pix	30.63	27.29	30.04	14.70	32.27	30.99	27.65
ControlNet(M)	32.66	30.72	31.93	14.92	33.66	36.15	30.01
Text2Image(M)	39.05	41.28	36.78	48.422	38.18	40.89	40.77

Table C.1.9: Comparison Between Editing and Condition Strategies.

	PACS	OfficeHome	NICO	DomainNet
SDEdit(M)	76.43	64.66	71.12	31.94
Text2Image(M)	83.72	67.63	71.54	40.77
ControlNet(M)	74.49	65.44	71.50	30.01
InstructPix2Pix	66.82	62.32	70.27	27.65
Retrieval	82.90	70.14	70.42	31.07

Table C.1.10: Impact of Assessing Multiple Real/Synthetic Domain.

	Art	Photo	Sketch	Cartoon	Average
ERM (Target)	99.65	99.94	99.64	99.66	99.72
ERM (All)	99.71	99.70	99.84	99.60	99.74
ERM (MDG)	80.01	96.28	73.86	76.28	81.61
OURS(MDG)	87.5	95.75	79.21	85.2	86.91

Table C.2.1: ImageNet-9 result with ResNet-18

	I.I.D. Test	Mixed Rand	Mixed Same	**Gap** (↓)
ERM	95.16	73.54	86.02	12.48
MixUp	94.62	67.63	83.91	16.28
CutMix	95.36	65.21	84.77	19.56
AugMix	95.16	74.73	87.72	12.99
RandAugment	96.69	78.20	90.44	12.25
CutOut	95.46	71.10	85.41	14.31
RSC	94.12	74.72	84.39	9.68
MEADA	95.56	74.74	87.43	12.69
PixMix	97.04	79.76	91.96	12.20
ACVC	93.97	76.38	88.16	11.77
L2D	92.84	73.04	84.10	11.06
SDEdit(H)	91.85	73.33	82.96	9.63
Text2Image(H)	90.12	69.63	75.8	6.17
ControlNet(H)	91.85	75.19	85.68	10.49
InstructPix2Pix	92.84	78.89	88.15	9.26
Retrieval	91.6	73.83	80.12	6.29

Table C.2.2: Texture result with ResNet-18

	I.I.D. Test	Random	**Texture Bias** (↓)
ERM	81.75	18.77	72.45
MixUp	77.36	19.23	71.69
CutMix	79.96	15.64	77.16
AugMix	82.2	20.08	72.42
RandAugment	83.09	18.9	72.2
CutOut	81.85	17.81	74.19
RSC	79.9	20.48	71.11
MEADA	81.97	19.5	71.14
PixMix	80.91	26.86	64.64
ACVC	81.13	29.33	59.25
L2D	80.06	23.55	62.12
SDEdit(H)	85.94	31.48	55.46
Text2Image(H)	86.44	35.23	51.21
ControlNet(H)	84.13	21.88	62.58
InstructPix2Pix	79.75	26.17	53.58
Retrieval	85.85	33.91	51.94

use "`Runmyml/stable-diffusion-v1-5`" pre-trained model. The training hyperparameters for setting are specified as shown in Table C.2.5

C.3 Prompting Strategies

Hwere we detail how the prompts were obtained.

- ***Domain expert (H)***: a collection of 1-8 simple "handcrafted" prompts per image domain (e.g., "an ink pen sketch of a(n) `class`"), authored by a human given only the domain descriptions provided by the respective benchmarks, without looking at any samples from the target

Table C.2.3: CelebA-sub result with ResNet-18

	I.I.D. Test	Flip	Random	**FlipGap** (↓)	**RandGap** (↓)
ERM	99.44	77.16	88.48	22.28	11.32
MixUp	99.16	79.4	88.86	19.76	9.46
CutMix	99.24	74.82	86.92	24.42	12.1
AugMix	99.56	76.42	87.82	23.14	11.4
RandAugment	99.04	77.62	88.96	21.42	11.34
CutOut	99.48	78.24	89.72	21.24	11.48
RSC	99.52	81.7	91.8	17.82	10.1
MEADA	99.48	77.24	89.08	22.24	11.84
ACVC	99.16	79.5	89.58	19.66	10.08
PixMix	99.32	76.62	88.34	22.7	11.72
L2D	99.12	81.96	90.96	17.16	9.0
Retrieval	98.6	77.9	89.3	20.7	11.4
SDEdit(H)	99.2	86.6	92.5	12.6	5.9
Text2Image(H)	98.8	90.0	93.6	8.8	3.6
ControlNet(H)	99.2	89.3	93.5	9.9	4.2
InstructPix2Pix	99.2	86.9	93.5	12.3	6.6

	Blurring Avg.	Noise Avg.	Compression Avg.	Transformations Avg.	Overall Avg.
ERM	72.04	74.01	75.71	73.93	73.55
MixUp	73.79	76.22	77.46	75.72	75.48
PixMix	75.84	79.41	81.39	79.32	78.63
SDEdit(M)	74.28	75.71	77.10	75.02	75.11

Table C.2.4: Average performance of algorithms across grouped domain shifts.

domain.

- ***Language enhancement (LE)***: following [He et al., 2023], we use the T5 language model [Raffel et al., 2020] fine-tuned on CommonGen [Lin et al., 2020][3] to generate 1-8 prompts using only the domain and class labels as inputs. Two strategies, Conservative (LE_C) and Moderate (LE_M), are used: LE_C deterministically generates consistent, high-probability outputs; and LE_M is built to balance prompt diversity with quality. For both strategies, we use a T5 [Raffel et al., 2020] model that is pre-trained on both unsupervised language

[3] `https://huggingface.co/mrm8488/t5-base-finetuned-common_gen`

Table C.2.5: Training Hyperparameters

	PACS	OfficeHome	NICO++	ImageNet-9	Texture	CelebA-sub
Epoch	50	50	50	30	30	30
Batch size	64	64	64	64	64	64
Warmup Epoch	5	5	5	5	5	5
Warmup Type	sigmoid	sigmoid	sigmoid	sigmoid	sigmoid	sigmoid
Weight Decay	5e-4	5e-4	5e-4	5e-4	5e-4	5e-4
Nesterov	True	True	True	True	True	True
Learning rate	1e-3	3e-3	3e-3	1e-3	1e-3	1e-3
Scheduler	Step	Step	Step	Step	Step	Step
Decay Step	45	45	45	27	27	27
Learning rate decay	0.1	0.3	0.3	0.1	0.1	0.1

Table C.2.6: Generator hyperparameters for each dataset

	PACS	OfficeHome	NICO++	DomainNet	ImageNet-9	CelebA	Texture
Inference Step	30	30	30	30	30	30	30
Image Strength	0.75	0.75	0.75	0.75	0.75	0.75	0.75
Guidance Scale	2.0	2.0	2.0	2.0	2.0	2.0	2.0
Sampler	UniPC	UniPC	UniPC	UniPC	UniPC	UniPC	UniPC

modeling of web text and supervised text-to-text language modeling tasks[4], then fine-tuned on CommonGen[5] [Lin et al., 2020]. (We refer to this model as T5_{CG}.) CommonGen is a constrained-generation task whose objective is to generate a sentence describing a commonplace scenario that contains all words[6] provided in an input word set. For example, given the words {dog, frisbee, catch, throw}, an acceptable output is "The dog catches the frisbee when the boy throws it." [Lin et al., 2020] We always provide T5_{CG} with a text input containing only a domain label and class label; for example, given a PACS image with domain `sketch` and class `elephant`, we simply feed T5_{CG} "sketch elephant" as input. For n number of prompts we will use to generate images, in LE_{C}, we simply use beam search decoding to generate prompts with $4n$ beams and select the top-n highest probability beams. In LE_{M}, we use a conjunction of top-k and top-p (nucleus) sampling, with $k = 50$ and $p = 0.95$, returning n sampled prompts. We experimented with other decoding configurations, but found

[4]Pre-trained model (not directly used in experiments): `https://huggingface.co/t5-base`

[5]Fine-tuned model used in experiments: `https://huggingface.co/mrm8488/t5-base-finetuned-common_gen`

[6]Synonyms and inflected forms are also allowed (e.g., given input "eat", outputs containing "consume" or "eaten" are valid).

Figure C.4.1: CLIP Filtering Examples. The most-similar (top) and least-similar (bottom) eight images according to their average percentile rank of CLIP similarity scores computed with respect to the provided prompts.

that increasing prompt diversity (e.g., by increasing k, lowering p, or increasing temperature) consistently came at the cost of prompt quality.

- ***Textual Inversion*** [Gal et al., 2022]: Given a set of images that share a common feature (e.g., belonging to the same class), this method learns an embedding in the text space that represents that feature. This embedding can be used to condition the generative process, thereby enhancing the generator's capability to reproduce that feature. Due to the computational cost associated with the additional training phase required by this approach, we limit its application to PACS. As shown in Table C.1.1 and Table C.1.2, Texual Inversion achieves 74.70% and 77.27% average accuracy for SDG. While outperforming all baseline methods, it is inferior to other relatively low-cost generative model-based strategies.

In order to yield the best IDA performance from a given T2I model, future work might consider strategies for directly optimising prompts or utilizing human-in-the-loop prompt "debugging", as we discuss in Section C.5 and C.6 (respectively).

C.4 CLIP Filtering Details

For each image in the generated dataset, we compute its CLIP similarity with respect to both prompts. Since the distributions of similarity scores can differ in scale and location, we cannot simply average the two scores in order to quantify how well a sample represents a class and a domain. Therefore, we sort the scores to produce two rankings and associate each image to the average of the percentile rank with respect to both prompts. We then discard a fixed amount of images with the lowest average percentile rank. (See Section C.4.1 for an example of top- and bottom-ranked images.) After filtering out the worst 10%, 25%, or 50% of synthetic images, we train our classifier on the remaining data. The results are displayed in Figure 4.4.3. We find filtering to not yield consistent improvements across all the considered cases.

C.4.1 CLIP Filtering Examples

Figure C.4.1 displays the best-matching (top) and worst-matching (bottom) synthetic images generated with `SDEdit` using LE_M of class `dog` and domain `cartoon` according to their average percentile rank of CLIP similarity scores with the prompts "an image of a dog" and "a cartoon". In

general, we observe that the images on the top do indeed appear to be cartoons and contain dogs (if somewhat disfigured in a few cases); whereas it seems that most of the images on the bottom either resemble *photos* of dogs (images 2, 4, 5, 6, and 8) or cartoons (images 1, 3, and 7), but do not generally seem to match both the target domain and the correct class.

C.5 Fully automated applications

We examine a basic implementation of a fully-automated augmentation pipeline in the language enhancement (**LE**) experiments described above, finding that it sometimes achieved performance on-par with or exceeding that of the expert-handcrafted prompts. However, this language model is optimised to generate simple sentences describing commonplace scenarios (see Section C.3), not image-generation prompts. Thus, it is possible that fine-tuning language models to generate prompts that are better optimised for downstream T2I generators may yield superior results to expert-handcrafted prompts in many scenarios, making this approach a promising direction for future work. Another approach to improve fully-automated prompting involves continuous prompt optimisation (also known as "prompt tuning" or "soft prompting"). Recently, these methods have been shown to outperform human-interpretable prompts for a variety of natural-language [Li and Liang, 2021, Liu et al., 2021a, Min et al., 2022, Khashabi et al., 2022] and vision-and-language [Gal et al., 2022, Zhou et al., 2022] tasks. These methods are not directly applicable to domain generalization because they require labelled samples to learn continuous prompts; but we suggest that they may be a promising fully-automated approach to domain adaptation tasks[7] where prompt interpretability is not necessary (cf. [Khashabi et al., 2022]).

C.6 Human-in-the-Loop Applications

C.6.1 Prompt interpretability enables human-in-the-loop debugging

Specifying interventions with natural language makes it possible to flexibly specify the type of manipulations desired. In the future, we expect practitioners could iteratively improve the collection of prompts to achieve improved performance.

We begin with a small set of handcrafted prompts (the ones used for the results reported in the main paper) and observe a decrease in texture bias of **5.44%**. Reasoning on the task at hand and the

[7] I.e., where an unlabeled sample of the target domain is available to facilitate learning of the environmental features of the target domain [Ghifary et al., 2016].

desired effect of the augmentations, we expand the prompts set to cover a broader range of textures to further decrease texture bias by an additional **2.84%** (see Section C.10 and Table C.2.2).

More generally, it is possible to "debug" augmentations by directly analysing prompts and modifying them to better reflect the desired intervention (which is possible with zero exposure to the target domain, or before augmented images are even generated). For example, the top prompts generated by **LE$_{\mathbf{M}}$** for OfficeHome's `art` domain and `computer` class include "art on a computer", "a man is working on a computer with a piece of art on it", etc., indicating that **LE$_{\mathbf{M}}$** generated prompts describing scenarios where both the class and domain label refer to individual objects in a visual scene. In Section C.6.1, we describe a few simple steps that can automatically filter out many such prompts[8], illustrating the flexibility of the natural-language augmentation interface.

LE$_{\mathrm{M}}$ is prone to generating prompts that treat OfficeHome [Venkateswara et al., 2017] domain labels as objects, not as visual domains or styles. Fortunately, the interpretability of natural-language prompts that makes it possible for us to diagnose this problem also enables us to filter out many such prompts. One approach is to map domain labels to the visual conditions they denote: for example, the `Product` label may be replaced with "white background", `Real World` with "photograph", etc. However, this solution requires some knowledge about test domains, which may not always be available. Alternatively, image-related keywords like "image", "depict", or "style" can be included in the input issued to T5$_{\mathrm{CG}}$, and outputs which do not place these additional terms in the same minimal noun phrase as the domain label can be removed (e.g., "an artistic depiction of a computer" or "a product image of a candle" would be kept, whereas "art depicted on a computer" or "a product and an image of a candle" would be excluded)[9]. While both of these strategies require limited human oversight to successfully "debug" prompts, more sophisticated fully-automated augmentation pipelines might learn to make such changes on their own, e.g., by integrating downstream image classifier accuracy as feedback to fine-tune prompt-generation models.

C.6.2 Other human-in-the-loop applications

The usage of T2I generators to approximate interventions facilitates a variety of novel use cases. For example, consider a "human-in-the-loop" (HITL) application context, where humans are available to provide interactive feedback to a model. In the HITL *active learning* paradigm, human experts perform the role of "oracles" that a model may "query" to provide labels of highly uncertain or novel

[8]However, as our LE experiments are explicitly intended to operate fully autonomously (i.e., with no human intervention or supervision), we do not carry out a full-scale "debugged" version of this experiment – all reported OfficeHome results are from the "buggy" prompts.

[9]For clarity, T5$_{\mathrm{CG}}$ can replace input terms with inflected forms in generated prompts, e.g., allowing input terms "art" and "depict" to occur as "artistic" and "depiction" (respectively) in outputs.

inputs [Mosqueira-Rey et al., 2022]. In contrast, the "human-in-the-loop debugging" paradigm elaborated above implements the *interactive machine teaching* paradigm [Ramos et al., 2020], which treats human collaborators as *teachers* that may provide interactive feedback to update the "curriculum"[10] of images used to train a model. For example, a human collaborator may observe that a model tends to perform poorly in the context of a given target domain, or that generated images do not capture some important stylistic properties of the domain. In response, they may easily compose or revise image-generation prompts with explicit reference to important features of the target domain. Critically, our approach allows human teachers to directly update visual curricula using natural language, providing models with feedback in much the same terms as one would a human student. We believe that the intuitiveness and efficiency of this approach makes it a promising approach to domain generalization, shifting the burden of the problem from human domain knowledge to natural language and thus enabling human collaborators to interactively instruct models without prerequisite domain expertise.

In particular, we argue that this benefit is particularly salient in the context of test-driven software engineering practice. Rather than blindly assuming that the performance on application-independent benchmarks will transfer to application-specific cases, engineers need to extensively document (often through natural language) the potential use cases and test conditions. The ability to directly specify these criteria via natural-language augmentations, or even directly reuse the documentation to generate training data, could be invaluable for controlling, predicting, and understanding the behavior of vision models in real-world applications.

C.7 Computational Expense

Although the inference speed of generative models has greatly improved over time, we found that SD is still too slow to generate synthetic data on-the-fly during training, so we pre-generate and store augmented data to amortise the generation cost when experimenting on different architectures and training procedures. For each sample in $\mathcal{D}^S$, we randomly selected k text prompts, and for each prompt, **one** augmented image was generated and stored. At training time, for each training image in the batch, one of its augmented versions will be randomly selected from the k pre-generated intervened samples. The general statistics of computational expense of each type of generative model on an NVIDIA A40 GPU and generator with hyperparameters specified for OfficeHome experiment are as follows: Stable Diffusion 1.5 took up ~ 8GB of VRAM (for inference – we do not

[10]Note that, in our case, a curriculum is defined in terms of the domains from which training examples are drawn, not the order in which they are presented (cf. [Bengio et al., 2009]).

Table C.7.1: Quantitative Comparison on Computation Time

	ERM	AugMix	RandAugment	MixUp	CutMix	RSC	L2D	ACVC	MEADA	OURS (online)
Time (s)	14.2	33.1	42.7	27.3	28.4	18.0	41.1	127.8	92.2	21.2

Table C.8.1: Inpainting Result on ImageNet-9

	in	mixed rand	mixed same	gap
ERM	95.06	71.85	83.58	11.73
SDEdit(H)	95.06	77.65	85.8	8.15
Inpaint(H)	**96.05**	**80.62**	**87.16**	**6.54**

compute gradients for any experiments) and required $\sim$ 0.5 seconds per sample generated on average. In addition to our qualitative assessments, we have conducted a quantitative comparison of various data generation methods, focusing on the time efficiency aspect. Specifically, we measured the time required to complete an epoch on the PACS dataset using a ResNet18 model. The results, detailed in Table C.7.1, reveal that the online augmentation speed of our method is on par with other parametric data augmentation methods and notably faster than learning-based methods. It's important to note that while the offline generation time for our method is approximately 4 hours on a single A40 GPU, this process is a one-time requirement and can be performed offline. Consequently, once the data is generated, it can be reused multiple times, thereby offsetting the initial time investment.

C.8 Further Experiments on Generative Models

C.8.1 Manipulating only the environmental features is important

It is important to observe that the T2I generator can manipulate not only the environmental features but also the class-related ones. When the manipulated class-related features still resemble those of the original training set, the issue is alleviated. However, it is important for future generators to allow stronger control over which features are manipulated and which not through language. In some cases, a potential solution could be to provide a mask that indicates which are the environmental features to be manipulated. To exemplify the importance of controlling mainly the environmental variables, we show that, when the inpainting capabilities of Stable Diffusion can leverage ground-truth background masks to preserve the foreground area, this further improves the performance of our method on ImageNet-9 as shown in Table C.8.1

Table C.8.2: Dataset Statistics

	No.train	No.validation	No.test	No.classes
PACS	8977	1014	9991	7
Officehome	14032	1556	15588	65
NICO++	61289	7661	15322	60
DomainNet	410657	18000	157918	345
ImageNet-9	2835	405	810	9
Texture	9600	1600	1280	16
CelebA-sub	5000	500	1000	2

C.9 Augmentation Samples

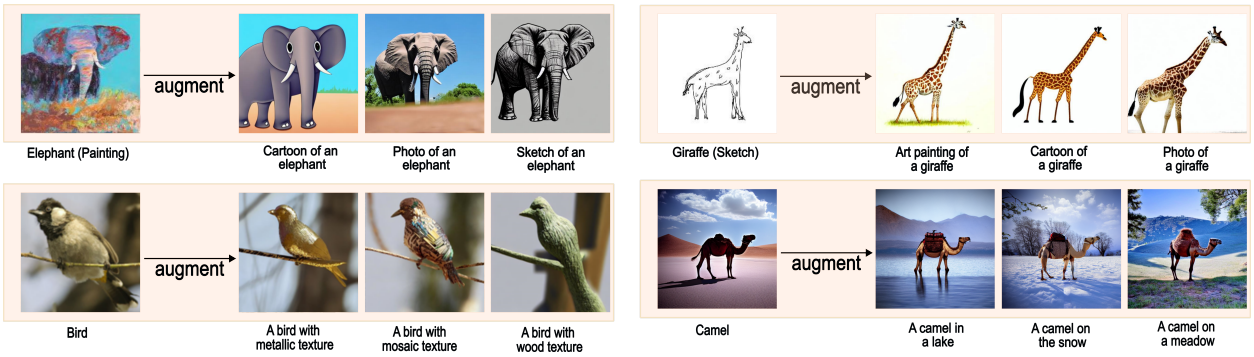


Figure C.9.1: Interventional samples generated by Stable Diffusion. For each group of four images, the leftmost image is the original image, and the three images on the right are augmented samples with text prompts indicated.

C.9.1 What kind of interventions can the generator approximate?

In our experiments, we have shown that the way current T2I generators approximate interventions is sufficient to achieve good performance on standard benchmarks. The way T2I generators learn to approximate such manipulations is by leveraging large amounts of weakly-supervised data. Stable Diffusion trains on text-image pairs scraped from the web with minimal post-processing (weak supervision): this is significantly less expensive than manually providing class and domain labels (with the added effort of controlling the environmental conditions). A natural question is then

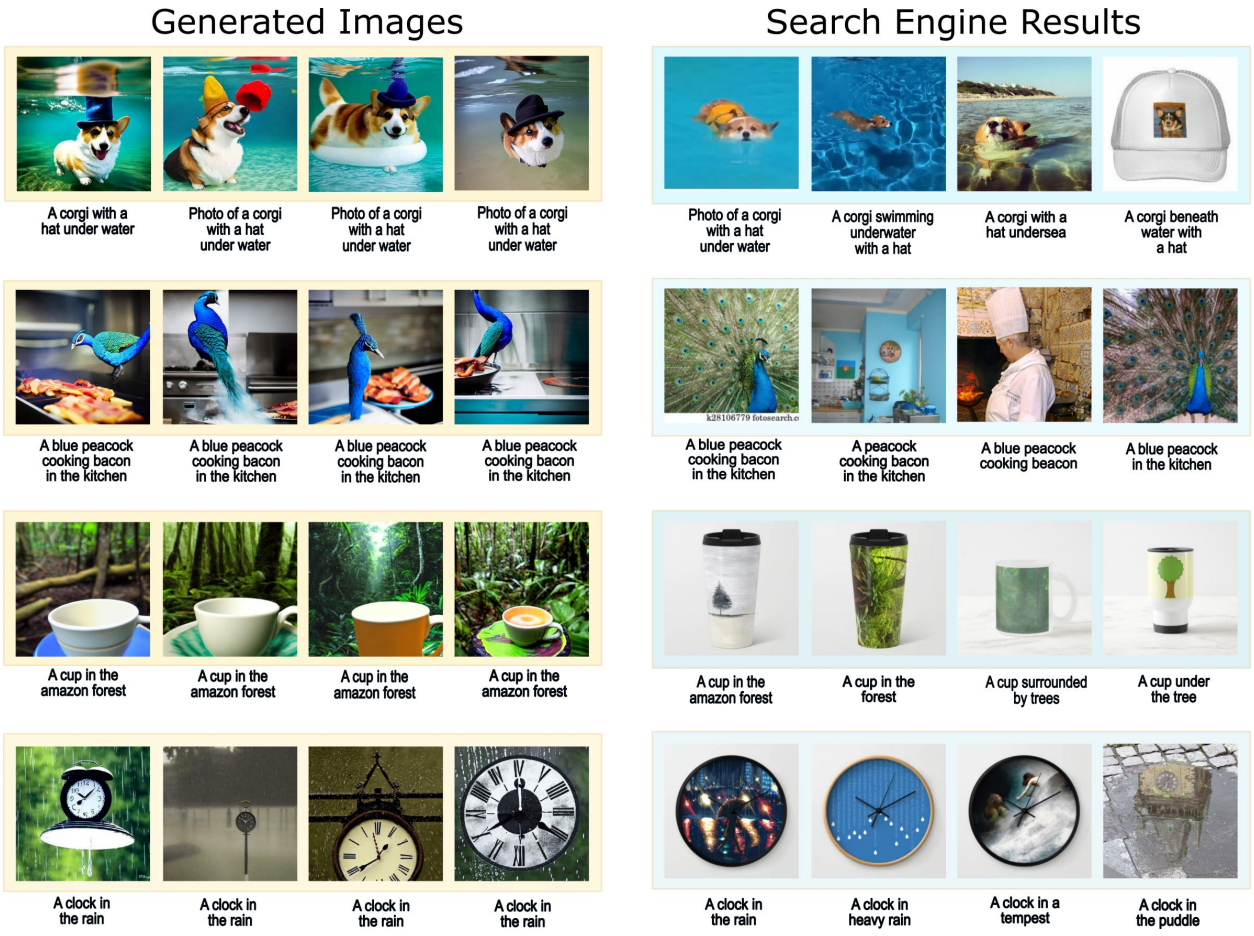


Figure C.9.2: Comparison between Search Engine retrieval result and Stable Diffusion manipulation results. Images on the left are generated with Stable Diffusion; images on the right are retrieved from LAION-5B by querying the search engine with the prompt indicated below

whether generators can approximate forms of interventions that are not represented in the training set. This would require them to combine learned concepts in novel ways. We answer this question through a simple experiment: we compare the results of generating images and retrieving images from the training set through a search engine[11] (see Figure C.9.2). Although the individual entities specified in the prompts are in the training set, we were unable to retrieve any images depicting the specific combination of entities and relations between them that was specified in the prompt. Since the dataset we are querying is huge (> 200TB, which can be impossible to store in lack of extremely expensive hardware), it is infeasible to give a certain answer about whether a sample representing the query is present or not in it. Additionally, the system leverages CLIP embeddings

[11] Search Engine can be accessed through `https://rom1504.github.io/clip-retrieval`

Figure C.9.3: Stable Diffusion manipulation for in-distribution samples with prompt indicated below. For each group of images of four, the first image on the left is the original image, and the rest three are manipulated images

Figure C.9.4: Stable Diffusion manipulation out-distribution samples with prompt indicated below. For each group of images of four, the first image is generated with prompt indicated from scratch, and the rest three are manipulated base on that.

to search for images similar to the query, so small differences in queries sometimes return highly variable results. For this reason, we try a variety of queries in an attempt to return images similar to the one that the generator produced to increase our confidence about the absence of a given image. While the first two examples (*"A corgi with a hat under water" and "A blue peacock cooking bacon in the kitchen"*) might be unlikely to occur in daily life, they might still occur in the context of captioning creative artworks (e.g., captioning of frames of animation movies or collage) and be useful to alleviate the reliance on spurious features (e.g., by perturbing the background or location in which an object is found). The last two examples (*"A cup in the amazon forest" and "A clock in the rain"*) exemplify much more common observations from the real world, that we could not retrieve from the training set. We also show SD can meaningfully manipulate synthetic images that cannot be found in the training set (see Figure C.9.4).

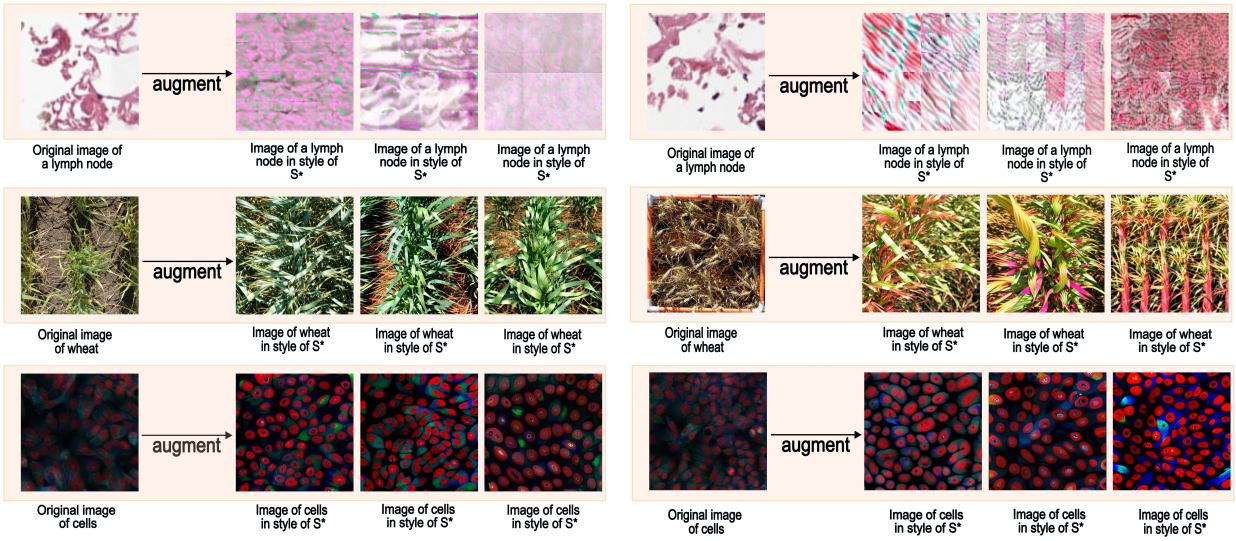


Figure C.9.5: Text Inversion manipulation results for dramatically out-of-distribution data to Stable Diffusion training domain, as a domain adaptation approach. For each case, four sample images are randomly selected from the target test domain, and a style token S_* is learnt with text inversion and used as a style prompt to augment the original training domain image. Images are manipulated with the Text Inversion prompt from the left first original image in each group of four images. The samples from top to bottom are 1) Histological image from Camelyon-17 [Bandi et al., 2018] 2) Cell image from RxRx1 [Taylor et al., 2019] 3) Wheat image from GlobalWheat [David et al., 2020, 2021].

C.9.2 Qualitative examples of the augmented images

In Figure C.9.3 we show additional examples of editing produced by Stable Diffusion. As it can be observed, Stable Diffusion may unintentionally manipulate features associated to the class label, without changing it. For instance, the augmented variants House and the Dog pictures in Figure C.9.3 significantly change their structure (e.g., structure of the house or breed of the dog), while preserving some similarities. Notice, this behavior is actually required when translating from domains with insufficient class-relevant information (e.g., when translating from a pencil sketch to a photo or painting, generators must infer color information).

C.9.3 The more the (synthetic) data, the better?

While in our framework the diversity of interventional samples is controlled by prompting strategy, a natural question is whether generating more samples can be beneficial. Therefore, for the PACS experiment, we ablate the amount of images we generate for each target domain. As shown in Figure C.9.6, increasing the amount of generated images up to 6 per-domain produces a 1.52%

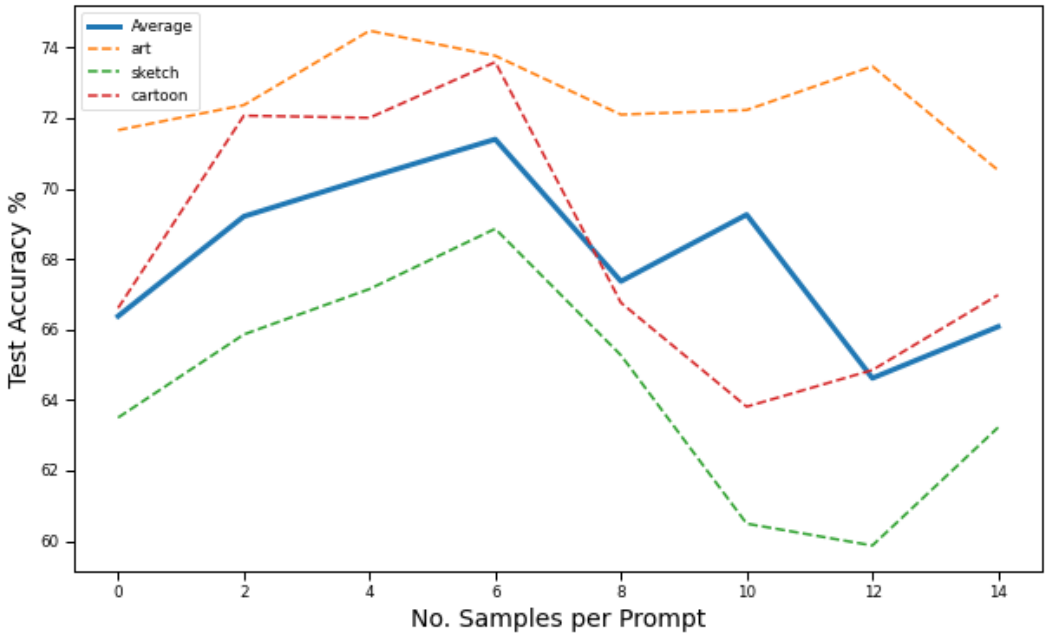


Figure C.9.6: Number of samples generated for each prompt against test accuracy. The test accuracy is based on SDEdit(M) with ResNet-18 trained on Photo source domain.

increase in the performance. Adding more data seems to degrade the performance. Note that, to ensure a fair comparison and disentangle the effect of having more data, we fix the number of samples seen across all iterations of the training procedure to be the same across all data points in the figure (i.e., the same batch size and training epoch, but more synthetic data sampled from a larger pre-generated candidate set). We leave to future work understanding whether this is due to the shift induced by the inevitable artifacts or low-quality images that might be produced when increasing the amount of generated samples or by the potentially low variety in the generated results.

C.9.4 Qualitative examples of failures

In Figure C.9.5 we present three failure cases of Stable Diffusion. In the first row, we observe Stable Diffusion fails at manipulating histological input images from the Camelyon-17 [Bandi et al., 2018] and the RxRx1 [Taylor et al., 2019] datasets. Camelyon-17 images contain tumoral and non-tumoral tissue captured in different environments. Since the changes between domains are hard to describe through language, we use Text inversion in order to learn how to transform from the source to the target domains. As it can be seen, Stable Diffusion fails to produce realistic samples

in this setting, probably because the input images and text are well out-of-domain. A less severe failure occurs on RxRx1 (second row), which represents HUVEC cells. In this case, the generated images still result in a distortion of the input that makes them unrealistic. For the GlobalWheat [David et al., 2020, 2021] dataset, it is apparent that while Stable Diffusion can generate plants but it does not reproduce the specific species depicted in the original input and sometimes produces completely unrealistic instantiations of plants. This failure is particularly bad considering its training set contains several images of wheat crops; however, in those images, the crops are not captured from the angle in which they are captured in GlobalWheat (thus inducing a distribution shift). These failures suggest future research should be directed towards improving the ability of T2I generators to manipulate only the environmental variables for out-of-domain data, under the assumption a few text and image pairs from these unknown domains can be leveraged.

C.9.5 The Domain Shift between Target Domain and Synthetic Target Domain

Sometimes the target domain description cannot fully represent the domain features as prompts to the generative model. For example, we observe the "Sketch" domain of the PACS dataset and the synthetic "Sketch" Domain is visually different as shown in Figure C.9.7. This is mainly due to the bias in specific dataset collection processes and also the bias in the training data of the generative model, which introduces the discrepancy in understanding of some natural language concepts.

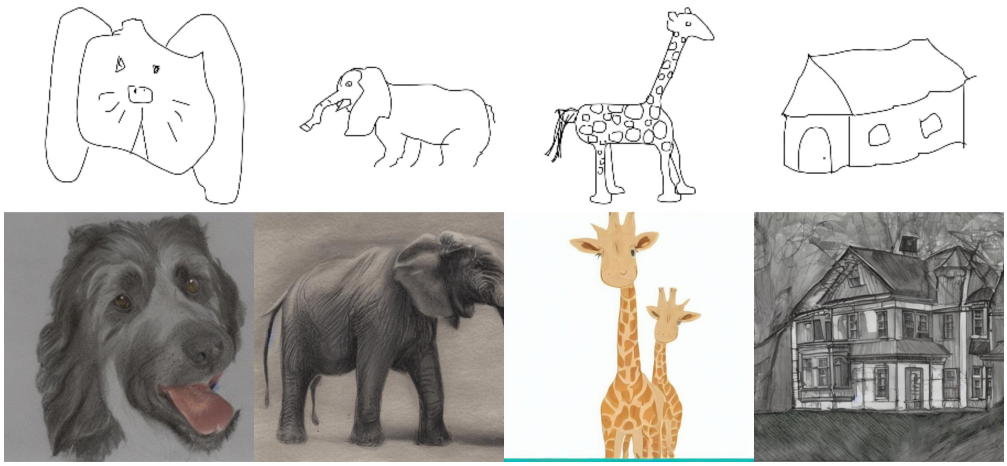


Figure C.9.7: Comparison between *"Sketch"* domain in PACS and Stable Diffusion Synthetic Data. **Top:** Sample sketch images from PACS dataset. **Bottom:** Sample synthetic data generated with SDEdit.

To further investigate the distributional mismatch and its relationship to classifier generalization

, we have performed additional quantitative analysis to show . We utilized the Fréchet Inception Distance (FID) score [Heusel et al., 2017], calculated with a pre-trained InceptionV3 model, to quantify the distribution shift between training and target out-of-distribution (OOD) test samples. As shown in the table below, we measured the FID between the training samples of each method and the target PACS test set, as indicated in the column names. As shown in Table C.9.1 We observe that the methods with the lowest FID score (Text2Image and Retrieval) yield classifiers with the highest accuracies, and that training on the original data (i.e., ERM) yields the lowest accuracy. These indicate a general negative correlation between distribution mismatch and generalization (as measured by the average test accuracy under SDG). However, we note that the FID scores are also very close among all generative methods, which makes FID a less sensitive metric to reflect the generalization of downstream classifiers. We hypothesize that this variation is due to the limited capacity of FID to reflect a more fine-grained distribution shift, indicating an important future research direction.

Table C.9.1: FID scores between PACS training data by augmentation method and target SDG test set, compared against average SDG test accuracy when training ResNet18

Method	Art	Photo	Sketch	Cartoon	Avg Distribution Shift	Avg Accuracy
No Augmentation (ERM)	264.2	311.7	358.0	221.7	288.9	58.74
SDEdit(M)	250.1	296.9	354.3	210.1	277.8	72.69
ControlNet(M)	249.8	294.9	354.4	210.9	277.5	72.32
Text2Image(M)	251.3	291.1	353.6	211.3	276.8	82.26
Retrieval(M)	249.0	294.6	354.0	210.2	276.9	80.83

C.9.6 Duplication Check

Table C.9.2: Proportion of image pairs in augmented training set and PACS test set with feature similarity higher than 0.9

	Art	Photo	Sketch	Cartoon
SDEdit(M)	0.10%	0.53%	1.13%	0.68%
Text2Image(M)	0.03%	0.55%	1.10%	0.56%
ControlNet(M)	0.04%	0.53%	1.22%	0.63%
Retrieval(M)	0.07%	0.68%	1.19%	0.64%

To ensure that the synthetic images used for augmentation do not cause data leakage, we have included additional visual duplication checks. Specifically, we leverage a pre-trained ResNet50

model to extract image features and calculate the cosine similarity between the training and test samples. We set a similarity threshold of 0.9, considering sample pairs above this threshold as potential duplicates. We report the proportion of such instances as shown in Table C.9.2 and visually inspect the most similar pairs, finding no evidence of duplication.

C.10 Image-Generation Prompts

We list the actual prompts used in all settings. The language enhancement prompts can either be generated by users following hint and language model specified in Section 4.3.2, or see our repo under *prompt* directory.

C.10.1 PACS

PACS: prompt is set in format "[TEMPLATE] of [CLASS LABEL]". The templates are as follows:

1. Minimal: 'art painting':['an art painting of'],'sketch':['a sketch of'],'cartoon':['a cartoon of'],'photo':['a photo of']

2. Hand-crafted: 'art painting': ['an oil painting of', 'a painting of', 'a fresco of', 'a colourful painting of', 'an abstract painting of', 'a naturalistic painting of', 'a stylised painting of', 'a watercolor painting of', 'an impressionist painting of', 'a cubist painting of', 'an expressionist painting of','an artistic painting of'], 'sketch':['an ink pen sketch of', 'a charcoal sketch of', 'a black and white sketch', 'a pencil sketch of', 'a rough sketch of', 'a kid sketch of', 'a notebook sketch of','a simple quick sketch of'], 'photo': ['a photo of', 'a picture of', 'a polaroid photo of', 'a black and white photo of', 'a colourful photo of', 'a realistic photo of'], 'cartoon': ['an anime drawing of', 'a cartoon of', 'a colorful cartoon of', 'a black and white cartoon of', 'a simple cartoon of', 'a disney cartoon of', 'a kid cartoon style of'] 3. Language Enhancement Moderate/Conservative: Generate with hint and language model specified in Section 4.3.2

C.10.2 OfficeHome

1. Minimal: 'Art':['an art image of'],'Clipart':['a clipart image of'],'Product':['an product image of '],'Real World':['a real world image of']

2. Handcrafted: 'Art':['a sketch of', 'a painting of', 'an artistic image of'],'Clipart':['a clipart image of'],'Product':['an image without background of '],'Real World':['a realistic photo of']

3. Language Enhancement Moderate/Conservative: Generate with hint and language model specified

in Section 4.3.2

C.10.3 NICO++

1. Minimal: 'autumn':['autumn'],'dim':['dim'],
'grass':['grass'],'outdoor':['outdoor'],'rock':['rock'],
'water':['water']}
2. Hand-crafted: 'autumn': ['in autumn', 'autumn', 'autumn with fallen leaves'], 'dim':['during sunset','in the evening','twilight'], 'grass': ['on grass','on grass meadow', 'with grass'], 'outdoor': ['in outdoor environment','outdoor', 'in wild environment'], 'rock':['on the rock','rock','with rock'], 'water':['in water','under water','water'] 3. Language Enhancement Moderate/Conservative: Generate with hint and language model specified in Section 4.3.2

C.10.4 DomainNet

1.Minimal: 'real': ['a photo of'], 'clipart': ['a clipart of'], 'sketch': ['a sketch of'], 'infograph': ['a infograph of'], 'quickdraw': ['a quickdraw of'], 'painting': ['a painting of']
2.Hand-crafted = 'real': ['a photo of', 'realistic photo of'], 'clipart': ['a clipart of', 'a prodcut image of'], 'sketch': ['a sketch of'], 'infograph': ['a infograph of'], 'quickdraw': ['a quickdraw of'], 'painting': ['a painting of']
3. Language Enhancement Moderate/Conservative: Generate with hint and language model specified in Section 4.3.2

C.10.5 ImageNet-9

1. Hand-crafted:background:[" in a parking lot", " on a sidewalk", " on a tree root", " on the branch of a tree", " in an aquarium", " in front of a reef", " on the grass", " on a sofa", " in the sky", " in front of a cloud", " in a forest", " on a rock", " in front of a red-brick wall", " in a living room", " in a school class", " in a garden", " on the street", " in a river", " in a wetland", " held by a person", " on the top of a mountain", " in a nest", " in the desert", " on a meadow", " on the beach", " in the ocean", " in a plastic container", " in a box", " at a restaurant", " on a house roof", " in front of a chair", " on the floor", " in the lake", " in the woods", " in a snowy landscape", " in a rain puddle", " on a table", " in front of a window", " in a store", " in a blurred backround"]

C.10.6 CelebA-sub

1. Hand-crafted experiment:
"blonde":["male"],"non-blonde":["female"]

C.10.7 Texture

We apply human-in-the-loop to iteratively improve the quality of prompt and augmentation in Texture dataset. We start with a set of heuristic prompt as original version. Then based on the image generated, we add more representative prompts to further diversity the texture features. As shown in Table C.2.2, by iteratively improving prompts, we achieve a final **8.28%** improvement more than **5.44%** of the initial improvement with respect to ERM.

1. Hand-crafted Final Version:
texture:['pointillism','rubin statue', 'rusty statue','ceramic','vaporwave','stained glass','wood statue','metal statue','bronze statue','iron statue','marble statue','stone statue','mosaic','furry','corel draw','simple sketch','stroke drawing', 'black ink painting','silhouette painting','black pen sketch','quickdraw sketch','grainy','surreal art','oil painting','fresco', 'naturalistic painting', 'stylised painting', 'watercolor painting', 'impressionist painting', 'cubist painting', 'expressionist painting','artistic painting']

2. Hand-crafted Original Version:
texture:['corel draw','simple sketch','stroke drawing', 'black ink painting','silhouette painting','black pen sketch','quickdraw sketch','grainy','surreal art','oil painting','fresco', 'naturalistic painting', 'stylised painting', 'watercolor painting', 'impressionist painting', 'cubist painting', 'expressionist painting','artistic painting']

D
Appendix of Chapter 5

Contents

D.1 Proofs

D.1.1 Noisy SGD

In this section, we present Algorithm 2, an adapted version of the Noisy SGD algorithm from [Bassily et al., 2014b] for d-dimensional linear halfspaces $\mathcal{H}^d$, that is used as a sub-procedure in Algorithm 1. Algorithm 2 first applies a base procedure $\mathcal{A}_{Base}$ on $\mathcal{H}^d$ for k times to generate a set of k results while preserving (ϵ, δ)-DP, and then applies the exponential mechanism $\mathcal{M}_E$ to output one final result from the set.

In Lemma 1, we state the privacy guarantee and the high probability upper bound on the excess error of the adpated version of Noisy SGD (Algorithm 2). This is a corollary of Theorem 2.4 in [Bassily et al., 2014a], which provides an upper bound on the expected excess risk of $\mathcal{A}_{Base}$. The proof of Lemma 1 follows directly from Markov inequality and the post-processing property of DP (Lemma 3), as described in Appendix D of [Bassily et al., 2014a].

Lemma 1 (Theoretical guarantees of Noisy SGD [Bassily et al., 2014b])**.** *Let the loss function ℓ be $\mathcal{L}$-Lipschitz and $\mathcal{H}^d$ be the d-dimensional linear halfspace with diameter* 1*. Then $\mathcal{A}_{Noisy-SGD}$ is (ϵ, δ)-DP, and with probability $1-\beta$, its output $\hat{w}$ satisfies the following upper bound on the excess risk,*

$$\sum_{(x,y)\in S} \ell(\hat{w}, (x,y)) - \sum_{(x,y)\in S} \ell(w^\star, (x,y)) = \frac{\mathcal{L}\sqrt{d}}{\epsilon} \cdot polylog(n, \frac{1}{\beta}, \frac{1}{\delta}),$$

for a labelled dataset S of size n. Here, $w^\star$ is the empirical risk minimizer

$$w^\star = argmin_{w\in\mathcal{C}} \sum_{(x,y)\in S} \ell(w, (x,y))$$

.

D.1.2 Theoretical results under no distribution shift and proofs

In this section, we provide a proof for Theorem 1, which demonstrates that PILLAR is (ϵ, δ)-DP with respect to the private dataset and can achieve accuracy with only a modest amount of private data. To establish Theorem 1, we begin by proving Lemma 2. This lemma shows that PILLAR attains a convergence guarantee in excess loss for all Lipschitz continuous loss functions in learning the linear halfspace $\mathcal{H}^d$.

Lemma 2. *Let $k \leq d \in \mathbb{N}, \gamma_0 \in (0,1)$, and $\xi_0 \in (0,1)$. Consider the family of distributions $\mathcal{D}_{\gamma_0,\xi_0}$ which consists of all (γ, ξ_k)-large margin low rank distributions over $\mathcal{X}_d \times \mathcal{Y}$, where $\gamma \geq \gamma_0$*

Algorithm 2 $\mathcal{A}_{Noisy-SGD}(S^L, \ell, (\epsilon, \delta), \beta)$

procedure $\mathcal{A}_{Noisy-SGD}(S^L, \ell, (\epsilon, \delta), \beta)$
  Input: a labelled dataset S^L, a loss function ℓ, privacy parameters ϵ, δ, and the failure probability β.
  Set $k = \lceil \log 1/\beta \rceil$.
  for $i = 1$ to k **do**
    $\hat{w}^{(i)} \leftarrow \mathcal{A}_{Base}(S^L, \ell, (\epsilon/k, \delta/k))$
  end for
  Let $O \leftarrow \{\hat{w}^{(1)}, \ldots, \hat{w}^{(k)}\}$.
  $\hat{w} \leftarrow \mathcal{M}_E(S^L, -\ell, O, \epsilon)$.
  Output: $\hat{w}$
end procedure
procedure $\mathcal{A}_{Base}((S^L, \ell, (\epsilon, \delta)))$
  Input: a labelled dataset S^L, a loss function ℓ, privacy parameters ϵ, δ.
  Let $\mathcal{L}$ be the Lipschitz coefficient of the loss function ℓ and n^L be the size of S^L.
  Set noise variance $\sigma^2 \leftarrow \frac{32L^2\left(n^L\right)^2 \log(n^L/\delta)\log(1/\delta)}{\epsilon}$.
  Randomize $\hat{w}^0 \in \mathcal{H}^d$.
  Set the learning rate function $\eta(t) = \frac{1}{\sqrt{t(n^L)^2\mathcal{L}^2 + m\sigma^2}}$.
  for $t = 1$ to $(n^L)^2 - 1$ **do**
    Uniformly choose $(x, y) \in S^L$.
    Update $\hat{w}^{t+1} = \Pi_{\mathcal{W}}\left(\hat{w}_t - \eta(t)[n^L \nabla \ell(\hat{w}^t; (x, y)) + \xi]\right)$ where $\xi \sim N(0, \mathbb{I}_d\sigma^2)$.
  end for
  Output: $\hat{w} = \hat{w}^{(n^L)^2}$
end procedure
procedure $\mathcal{M}_E(S^L, \ell, O, \epsilon)$
  Input: a dataset S^L, a loss function ℓ, an set of parameters O, and a privacy parameter ϵ.
  Set the global sensitivity as $\Delta_U = \max_{S,S'} \max_{w \in O} |\ell(S, w) - \ell(S', w)|$, for any S, S' of size $|S^L|$ differing at exactly one entry.
  Output: $w \in O$ with probability proportional to $\exp\left(\frac{\epsilon \ell(S^L, w)}{2\Delta_U}\right)$.
end procedure

and $\xi_k \leq \xi_0$. For any $\alpha \in (0,1), \beta \in (0, 1/4)$, $\epsilon \in (0, 1/\sqrt{k})$, and $\delta \in (0,1)$, PILLAR $\mathcal{A}_{\epsilon,\delta}(k,\ell)$, described by Algorithm 1 with an L-Lipschitz loss function ℓ in step 5, is (ϵ,δ)-DP and outputs an estimator $\hat{w}$ satisfying

$$\mathbb{P}_{(S^U,S^L)\sim D,\hat{w}\sim\mathcal{A}_{\epsilon,\delta}}\left[\mathbb{E}_{(x,y)\sim D}\left[\ell(\hat{w};(x,y))\right] - \min_{w\in B_2^d}\mathbb{E}_{(x,y)\sim D}\left[\ell(w;(x,y))\right] \leq \alpha\right] \geq 1-\beta,$$

given a public unlabelled and private labelled sample S^U, S^L from distribution D of size

$$n^U = O\left(\frac{\log 2/\beta}{(1-\xi_0)^2\Delta_k^2}\right), n^L = \widetilde{O}\left(\left(\frac{1}{\alpha^2}+\frac{1}{\alpha\epsilon}\right)L\sqrt{k}\right).$$

Proof. **Privacy guarantee** Algorithm $\mathcal{A}_{\epsilon,\delta}(k,\zeta)$ computes the transformation matrix $\hat{A}_k$ on the public unlabelled dataset. This step is independent of the labelled data S^L and has no impact on the privacy with respect to S^L. $\mathcal{A}_{\text{Noisy-SGD}}$ ensures the operations on the labelled dataset S^L to output v_k is (ϵ,δ)-DP with respect to S^L (Lemma 1). The final output $\hat{w} = \hat{A}_k v_k$ is attained by post-processing of v_k and preserves the privacy with respect to S^L by the post-processing property of differential privacy (Lemma 3).

Lemma 3 (Post-processing [Dwork et al., 2006])**.** *For every (ϵ,δ)-DP algorithm $\mathcal{A}: S \to \mathcal{Y}$ and every (possibly random) function $f: \mathcal{Y} \to \mathcal{Y}'$, $f \circ \mathcal{A}$ is (ϵ,δ)-DP.*

Accuracy guarantee By definition, all distributions $D_{\gamma,\xi_k} \in \mathcal{D}_{\gamma_0,\xi_0}$ are (γ,ξ_k)-large margin low rank for some $\gamma \geq \gamma_0, \xi_k \leq \xi_0$. Let the empirical covariance matrix of D_{γ,ξ_k} calculated with the unlabelled dataset S^U be $\widehat{\Sigma} = \frac{1}{n^U}\sum_{x\in S^U}(x-\bar{x})(x-\bar{x})^\top$ and $\hat{A}_k \in \mathbb{R}^{d\times k}$ be the projection matrix whose i^{th} column is the i^{th} eigenvector of $\widehat{\Sigma}$. Let Σ be the population covariance matrix and similarly, let A_k the matrix of top k eigenvectors of Σ.

For any distribution $D_{\gamma,\xi_k} \in \mathcal{D}_{\gamma_0,\xi_0}$, let $D_{X,(\gamma,\xi_k)}$ be the marginal distribution of X and $w^\star$ be the large margin linear classifier that is guaranteed to exist by Definition 3. The margin after projection by $\hat{A}_k$ is lower bounded by $\frac{y\left\langle \hat{A}_k^\top z, \hat{A}_k^\top w^\star\right\rangle}{\|\hat{A}_k^T z\|_2 \|\hat{A}_k^\top w^\star\|_2}$ for any $z \in \operatorname{supp} D_{X,(\gamma,\xi_k)}$.

We will first derive a high-probability lower bound for this term to show that, after projection, with high probability, the projected distribution still has a large margin. Then, we will invoke existing algorithms in the literature with the right parameters, to privately learn a large margin classifier in this low-dimensional space.

Let z be any vector in $\operatorname{supp} D_{X,(\gamma,\xi_k)}$. We can write $z = a_z w^\star + b^\perp$ for some a_z where $b^\perp$ is in the nullspace of $w^\star$. Then, it is easy to see that using the large-margin property in Definition 3, we get

$$y a_z = \frac{\langle w^\star, z\rangle}{\|w^\star\|_2 \|z\|_2} \geq \gamma \geq \gamma_0. \tag{D.1}$$

Then, we lower bound $\frac{y\left\langle \hat{A}_k^\top z, \hat{A}_k^\top w^\star \right\rangle}{\left\|\hat{A}_k^\top z\right\|_2 \left\|\hat{A}_k^\top w^\star\right\|_2}$ as

$$\frac{y\left\langle \hat{A}_k^\top z, \hat{A}_k^\top w^\star \right\rangle}{\left\|\hat{A}_k^\top z\right\|_2 \left\|\hat{A}_k^\top w^\star\right\|_2} \overset{(a)}{=} \frac{y a_z \left\|\hat{A}_k^\top w^\star\right\|_2^2}{\left\|\hat{A}_k^\top z\right\|_2 \left\|\hat{A}_k^\top w^\star\right\|_2} \overset{(b)}{\geq} \gamma_0 \left\|\hat{A}_k^\top w^\star\right\|_2 \tag{D.2}$$

where step (a) is due to $\langle w^\star, b^\perp \rangle = 0$ and step (b) follows from $\left\|\hat{A}_k^\top z\right\|_2 \leq \left\|\hat{A}_k\right\|_{\text{op}} \|z\|_2 \leq 1$ and Equation (D.1).

To lower bound $\left\|\hat{A}_k^\top w^\star\right\|_2$, note that

$$\begin{aligned}
\left\|\hat{A}_k^\top w^\star\right\|_2 = \left\|\hat{A}_k \hat{A}_k^\top w^\star\right\|_2 &\geq \left\|A_k A_k^\top w^\star\right\|_2 - \left\|\hat{A}_k \hat{A}_k^\top w^\star - A_k A_k^\top w^\star\right\|_2 \quad \text{by Triangle Inequality} \\
&\geq \left\|A_k A_k^\top w^\star\right\|_2 - \left\|\hat{A}_k \hat{A}_k^\top - A_k A_k^\top\right\|_F \left\|w^\star\right\|_2 \quad \text{by Cauchy Schwarz Inequality} \\
&\geq 1 - \xi_k - \left\|\hat{A}_k \hat{A}_k^\top - A_k A_k^\top\right\|_F .
\end{aligned} \tag{D.3}$$

where the last step follows from the low rank assumption in Definition 3 and observing that $\left\|w^\star\right\|_2 = 1$.

To upper bound $\left\|\hat{A}_k \hat{A}_k^\top - A_k A_k^\top\right\|_F$, we use Lemma 4.

Lemma 4 (Theorem 4 in [Zwald and Blanchard, 2005])**.** *Let D be a distribution over $\{x \in \mathbb{R}^d | \ \|x\|^2 \leq 1\}$ with covariance matrix Σ and zero mean $\mathbb{E}_{x\sim D}[x] = 0$. For a sample S of size n from D, let $\widehat{\Sigma} = \frac{1}{n}\sum_{x\in S} xx^T$ be the empirical covariance matrix. Let $A_k, \hat{A}_k$ be the matrices whose columns are the first k eigenvectors of Σ and $\widehat{\Sigma}$ respectively and let $\lambda_1(\Sigma) > \lambda_2(\Sigma) > \ldots > \lambda_d(\Sigma)$ be the ordered eigenvalues of Σ. For any $k > 0, \beta \in (0,1)$ such that $\lambda_k(\Sigma) > 0$ and $n \geq \frac{16\left(1+\sqrt{\beta/2}\right)^2}{(\lambda_k(\Sigma)-\lambda_{k+1}(\Sigma))^2}$, we have that with probability at least $1 - e^{-\beta}$,*

$$\left\|A_k A_k^T - \hat{A}_k \hat{A}_k^T\right\|_F \leq \frac{4\left(1 + \sqrt{\frac{\beta}{2}}\right)}{\left(\lambda_k(\Sigma) - \lambda_{k+1}(\Sigma)\right)\sqrt{n}}.$$

It guarantees that with probability $1 - \frac{\beta}{2}$,

$$\left\|A_k A_k^\top - \hat{A}_k \hat{A}_k^\top\right\|_F \leq \frac{4\left(1 + \sqrt{\frac{\log(2/\beta)}{2}}\right)}{\left(\lambda_k(\Sigma) - \lambda_{k+1}(\Sigma)\right)\sqrt{n^U}} \leq \frac{1-\xi_0}{10}. \tag{D.4}$$

where the last inequality follows from choosing the size of unlabelled data $n^U \geq \frac{1600\left(1+\sqrt{\frac{\log(2/\beta)}{2}}\right)^2}{(1-\xi_0)^2(\Delta_{\min}\lambda_k)^2}$.

Substituting Equation (D.4) into Equation (D.3), we get that with probability $1-\frac{\beta}{2}$,

$$\left\|\hat{A}_k^\top w^\star\right\|_2 \geq 1-\xi_k-\frac{1-\xi_0}{10} \geq 1-\xi_0-\frac{1-\xi_0}{10} = 0.9\left(1-\xi_0\right) \tag{D.5}$$

Plugging Equation (D.5) into Equation (D.2), we derive a high-probability lower bound on the distance of any point to the decision boundary in the transformed space. For all $z \in \operatorname{supp} D_{X,(\gamma,\xi_k)}$,

$$\frac{y\left\langle \hat{A}_k^\top z, \hat{A}_k^\top w^\star\right\rangle}{\left\|\hat{A}_k^\top z\right\|_2\left\|\hat{A}_k^\top w^\star\right\|_2} \geq 0.9\gamma_0\left(1-\xi_0\right). \tag{D.6}$$

This implies that the margin in the transformed space is at least $0.9\gamma_0\left(1-\xi_0\right)$.

For a halfspace with parameter $v \in B_2^k$, denote the empirical loss on a dataset S by $\hat{L}(w;S) = \frac{1}{|S|}\sum_{(x,y)\in S}\ell(w,(x,y))$ and the loss on the distribution D by $L(w;D) = \mathbb{P}_{(x,y)\sim D}\left[\ell(w,(x,y))\right]$. Let D_k be the k-dimension transformation of the original distribution D obtained by projecting each $x \in \mathcal{X}$ to $\hat{A}_k^\top x$.

By the convergence bound in Lemma 1 for $\mathcal{A}_{\text{Noisy-SGD}}$, we have with probability $1-\frac{\beta}{2}$, $\mathcal{A}_{\text{Noisy-SGD}}$ outputs a hypothesis $v_k \in B_2^k$ and $\hat{w} = A_k v_k \in B_w^d$ such that

$$\hat{L}(\hat{w};S^L) - \hat{L}(w_{ERM};S^L) \overset{(a)}{=} \hat{L}(v_k;S_k^L) - \hat{L}(v_{ERM};D_k) = O\left(\frac{L\sqrt{k}}{n^L\epsilon}\text{polylog}\left(n^L,\frac{1}{\delta},\frac{1}{\beta}\right)\right) \tag{D.7}$$

where $w_{ERM} = \text{argmin}_{w\in B_2^d}\hat{L}(w;S^L)$ and $v_{ERM} = \text{argmin}_{v\in B_2^k}\hat{L}(v;S_k^L)$.

Let $w^\star$ be the ground truth of the given distribution. The generalization error can be decomposed as

$$\begin{aligned} L(\hat{w}) - L(w^\star) &= \left(L(\hat{w}) - \hat{L}(\hat{w})\right) + \left(\hat{L}(\hat{w}) - \hat{L}(w_{ERM})\right) + \left(\hat{L}(w_{ERM}) - \hat{L}(w^\star)\right) + \left(\hat{L}(w^\star) - \hat{L}(w^\star)\right) \\ &\overset{(a)}{\leq} \underbrace{\left(L(\hat{w}) - \hat{L}(\hat{w})\right)}_{(a)} + \underbrace{\left(\hat{L}(\hat{w}) - \hat{L}(w_{ERM})\right)}_{(b)} + \underbrace{\left(\hat{L}(w^\star) - \hat{L}(w^\star)\right)}_{(c)} \end{aligned} \tag{D.8}$$

where step (a) follows as $\hat{L}(w_{ERM}) - \hat{L}(w^\star) \leq 0$ by the definition of w_{ERM}.

We have shown in Equation (D.7) that the second term (b) is upper bounded by $\frac{\alpha}{2}$ for $n^L = \tilde{O}\left(\frac{L\sqrt{k}}{\alpha\epsilon}\right)$. It remains to bound the generalization error of linear halfspace $\mathcal{H}^d$ for L-Lipschitz loss function, ie. term (a) and term (c). That is, we need to show that the empirical error of a linear halfspace is a good approximation of the error on the distribution. To achieve this, we apply uniform convergence bound using Rademacher complexity [Mohri et al., 2012].

With probability $1-\frac{\beta}{4}$,

$$\sup_{w\in B_2^d}\left(\mathbb{E}_{x,y\sim D}\ell(w;(x,y))-\frac{1}{n^L}\sum_{(x,y)\in S^L}\ell(w;(x,y))\right)\leq 2\mathfrak{R}_{S^L}(\mathcal{H}_\ell)+\sqrt{\frac{3\log\frac{8}{\beta}}{2n^L}}, \tag{D.9}$$

where $\mathcal{H}_\ell=\{h_w(x,y)=\ell(w,(x,y))|w\in B_2^d\}$ is the composition of the loss function with the linear halfspace.

$$\mathfrak{R}_{S^L}(\mathcal{H}_\ell)\leq L\mathfrak{R}_{S^L}(\mathcal{H}^d)=\frac{L}{n^L}. \tag{D.10}$$

Substituting Equation (D.10) into Equation (D.9), we can upper bound both term (a) and (c) by $\frac{\alpha}{4}$ with probability at least $1-\frac{\beta}{2}$ for $n^L\geq\frac{L}{\alpha^2}\text{polylog}\left(\left(\right)\frac{4}{\beta}\right)$, *i.e.*

$$L(\hat{w})-\hat{L}(\hat{w})\leq\frac{\alpha}{4},\quad \hat{L}(w^\star)-L(w^\star)\leq\frac{\alpha}{4}. \tag{D.11}$$

Combining Equation (D.8), Equation (D.7) and Equation (D.11) concludes the proof.

□

In the following, we use Lemma 2 to prove Theorem 1. Recall that we define cross entropy loss and scaled hinge loss in Table 5.3.1.

Theorem 1. *Let $k\leq d\in\mathbb{N},\gamma_0\in(0,1)$, and $\xi_0\in(0,1)$. Consider the family of distributions $\mathcal{D}_{\gamma_0,\xi_0}$ which consists of all (γ,ξ_k)-large margin low rank distributions over $\mathcal{X}_d\times\mathcal{Y}$, where $\gamma\geq\gamma_0$ and $\xi_k\leq\xi_0$. For any $\alpha\in(0,1)$,$\beta\in(0,1/4)$, $\epsilon\in(0,1/\sqrt{k})$, and $\delta\in(0,1)$, PILLAR with scaled hinge loss or cross entropy loss, is an $(\alpha,\beta,\epsilon,\delta)$-semi-private learner for linear halfspaces $\mathcal{H}^d$ on $\mathcal{D}_{\gamma_0,\xi_0}$ with sample complexity*

$$n^U=O\left(\frac{\log 2/\beta}{(1-\xi_0)^2\Delta_k^2}\right),n^L=\widetilde{O}\left(\frac{L_\ell\sqrt{k}}{\alpha\epsilon}\right)$$

where Δ_k denote the gap between the k^{th} and the $k+1^{\text{th}}$ eigenvalue of the population covariance matrix, and L_ℓ is the Lipschitz coefficient of the loss function ℓ[1].

Proof. **Guarantees for PILLAR with (scaled) hinge loss function:** Note that the (scaled) hinge loss function ℓ_ζ^h defined in Table 5.3.1 is $\frac{1}{0.9\gamma_0(1-\xi_0)}$-Lipschitz. Substituting $L_\ell=\frac{1}{0.9\gamma_0(1-\xi_0)}$ into the sample complexity in Lemma 2 upper bounds the excess hinge loss of PILLAR's output $\hat{w}$ with probability at least $1-\beta$, *i.e.*

$$\begin{aligned}&\mathbb{E}_{(x,y)\sim D}\left[\ell(\hat{w};(x,y))\right]-\min_{w\in B_2^d}\mathbb{E}_{(x,y)\sim D}\left[\ell(w;(x,y))\right]\\ \overset{(a)}{=}&\mathbb{E}_{(x,y)\sim D}\left[\ell(v_k;(A_k^\top x,y))\right]-\min_{v\in B_2^k}\mathbb{E}_{(x,y)\sim D}\left[\ell(v;(A_k^\top x,y))\right]\leq\alpha\end{aligned} \tag{D.12}$$

[1]Note that $\widetilde{O}$ neglects the logarithmic terms associated with $\frac{1}{\delta}$ and $\frac{1}{\beta}$.

where step (a) follows from the definition of $\hat{w} = A_k v_k$ by the last step in PILLAR using the same notation as in the proof of Lemma 2.

By Equation (D.5) following the same argument as in the proof of Lemma 2, the k-dimensional space projected by A_k has a positive margin at least $0.9\gamma_0(1-\xi_0)$. Thus, the empirical risk minimizer in the low-dimensional space is zero, *i.e.*

$$\mathbb{E}_{(x,y)\sim D}\left[\ell(\hat{w};(x,y))\right] = \min_{v\in B_2^k} \mathbb{E}_{(x,y)\sim D}\left[\ell(v;(A_k^\top x,y))\right] = 0. \tag{D.13}$$

Then, we can upper bound the empirical 0-1 error by the empirical (scaled) hinge loss in the k-dimensional transformed space, For $n^L = O\left(\frac{\sqrt{k}}{\alpha\epsilon\gamma_0(1-\xi_0-0.1\gamma_0)}\text{polylog}\left(\frac{1}{\delta},\frac{1}{\epsilon},\frac{1}{\beta},\frac{1}{\alpha},\frac{1}{\gamma_0},\frac{1}{\xi_0},k,n^L\right)\right)$, with probability $1-\frac{\beta}{4}$,

$$\begin{aligned}\mathbb{E}_{(x,y)\sim D}\left[\mathbb{1}\{y\langle x,\hat{w}\rangle\}\right] &= \mathbb{E}_{(x,y)\sim D}\left[\mathbb{1}\{y\left\langle A_k^\top x, v_k\right\rangle\}\right] \\ &\le \mathbb{E}_{(x,y)\sim D}\left[\ell(v_k;(A_k^\top x,y))\right]\end{aligned} \tag{D.14}$$

Combining Equation (D.12), Equation (D.13) and Equation (D.14) concludes the proof.

Guarantees for PILLAR with cross entropy loss: As cross entropy loss function ℓ_{CN} defined in Table 5.3.1 is 2-Lipschitz, directly applying Lemma 2 shows that excess cross-entropy loss ℓ_{CN} is upper bounded by $\frac{\alpha}{2}$ with the given public unlabelled and private labelled samples, *i.e.*

$$\mathbb{P}_{(S^U,S^L)\sim D,\hat{w}\sim\mathcal{A}_{\epsilon,\delta}}\left[\mathbb{E}_{(x,y)\sim D}\left[\ell_{CN}(\hat{w};(x,y))\right] - \min_{w\in B_2^d}\mathbb{E}_{(x,y)\sim D}\left[\ell_{CN}(w;(x,y))\right] \le \frac{\alpha}{2}\right] \ge 1-\beta, \tag{D.15}$$

when $n^U = O\left(\frac{\log 2/\beta}{(1-\xi_0)^2\Delta_k^2}\right), n^L = \widetilde{O}\left(\frac{\sqrt{k}}{\alpha\epsilon}\right)$.

We apply Theorem 7 in [Bartlett et al., 2003] with $\psi(\theta)=\theta$ and $\alpha=1$ for cross entropy loss to obtain an upper bound on excess 0-1 loss,

$$\begin{aligned}&\mathbb{E}_{(x,y)\sim D}\left[\ell_{CN}(\hat{w};(x,y))\right] - \min_{w\in B_2^d}\mathbb{E}_{(x,y)\sim D}\left[\ell_{CN}(w;(x,y))\right] \\ \ge&\frac{1}{2}\left(\mathbb{E}_{(x,y)\sim D}\left[\mathbb{1}\{y\,(x,\hat{w})>0\}\right] - \min_{w\in B_2^d}\mathbb{E}_{(x,y)\sim D}\left[\mathbb{1}\{y\,(x,\hat{w})>0\}\right]\right)\end{aligned} \tag{D.16}$$

Substitute Equation (D.16) into Equation (D.15), we obtain the convergence guarantee on 0-1 loss.

$$\begin{aligned}&\mathbb{P}_{(S^U,S^L)\sim D,\hat{w}\sim\mathcal{A}_{\epsilon,\delta}}\left[\mathbb{E}_{(x,y)\sim D}\left[\ell_{CN}(\hat{w};(x,y))\right] - \min_{w\in B_2^d}\mathbb{E}_{(x,y)\sim D}\left[\ell_{CN}(w;(x,y))\right] \le \frac{\alpha}{2}\right] \\ \le&\mathbb{P}_{(S^U,S^L)\sim D,\hat{w}\sim\mathcal{A}_{\epsilon,\delta}}\left[\mathbb{E}_{(x,y)\sim D}\left[\mathbb{1}\{y\,(x,\hat{w})>0\}\right] - \min_{w\in B_2^d}\mathbb{E}_{(x,y)\sim D}\left[\mathbb{1}\{y\,(x,\hat{w})>0\}\right] \le \alpha\right] \ge 1-\beta.\end{aligned} \tag{D.17}$$

This completes the proof. □

D.1.3 Privacy guarantees for PILLAR on the original image dataset

As described in Figure 5.2.1, in practice PILLAR is applied on the set of representations obtained by passing the private dataset of images through a pre-trained feature extractor. Therefore, a straightforward application of Theorem 1 yields an (ϵ, δ)-DP guarantee on the set of representations and not on the dataset in the raw pixel space themselves. Here, we show that PILLAR provides (at least) the same DP guarantees on the dataset in the pixel space as long as the pre-training dataset cannot be manipulated by the privacy adversary. One way to achieve this, as we show is possible in this paper, is by using the same pre-trained model across different tasks. Investigating the extent of privacy harm that can be caused by allowing the adversary to manipulate the pre-training data remains an important future direction.

Corollary 1. *Let $f : \mathbb{R}^p \to \mathbb{R}^d$ be a feature extractor pre-trained using any algorithm. Let S_1, S_2 be any two neighbouring datasets of private images in $\mathbb{R}^p$. Then, for any $Q \subseteq \mathcal{H}^d$ where $\mathcal{H}^d$ is the class of linear halfspaces in d dimensions,*

$$\mathbb{P}_{h\sim\mathcal{A}_{\epsilon,\delta}\circ f(S_1)}\left[h \in Q\right] \leq e^{\epsilon}\mathbb{P}_{h\sim\mathcal{A}_{\epsilon,\delta}\circ f(S_2)}\left[h \in Q\right] + \delta$$

where $\mathcal{A}_{\epsilon,\delta}$ is Algorithm 1 (PILLAR) run with privacy parameters ϵ, δ.

Proof. Note that f is a deterministic many-to-one function from the dataset of images to the dataset of representations [2]. For any two neighbouring datasets S_1, S_2 in the image space, let $S_1^{\mathrm{R}}, S_2^{\mathrm{R}}$ be the corresponding set of representations extracted by f, *i.e.* $S_1^{\mathrm{R}} = \{f(x) : x \in S_1\}$ and $S_2^{\mathrm{R}} = \{f(x) : x \in S_2\}$. Then for any $Q \subseteq \mathcal{H}^d$

$$\begin{aligned}
\mathbb{P}_{h\sim\mathcal{A}_{\epsilon,\delta}\circ f(S_1)}\left[h \in Q\right] &= \mathbb{P}_{h\sim\mathcal{A}_{\epsilon,\delta}\left(S_1^{\mathrm{R}}\right)}\left[h \in Q\right] \\
&\leq e^{\epsilon}\mathbb{P}_{h\sim\mathcal{A}_{\epsilon,\delta}\left(S_2^{\mathrm{R}}\right)}\left[h \in Q\right] + \delta \\
&= e^{\epsilon}\mathbb{P}_{h\sim\mathcal{A}_{\epsilon,\delta}\circ f(S_2)}\left[h \in Q\right] + \delta
\end{aligned}$$

where the first and the last equality follows by using the definition $S_1^{\mathrm{R}}, S_2^{\mathrm{R}}$ and due to the fact that f is a many-to-one function. The second inequality follows from observing that $S_1^{\mathrm{R}}, S_2^{\mathrm{R}}$ can differ on at most one point as f is a deterministic many-to-one function and $\mathcal{A}_{\epsilon,\delta}$ is (ϵ, δ)-DP. □

D.1.4 Theoretical results under distribution shifts and proofs

In this section, we provide the theoretical guarantees of PILLAR under distribution shifts. Before that, we formally define η-TV tolerant semi-private learning.

[2] f can be designed to normalize the extracted features in a d-dimensional unit ball.

Definition 4 (η-TV tolerant $(\alpha,\beta,\epsilon,\delta)$-semi-private learner on a family of distributions $\mathcal{D}$)**.** *An algorithm $\mathcal{A}$ is an η-TV tolerant $(\alpha,\beta,\epsilon,\delta)$-semi-private learner for a hypothesis class $\mathcal{H}$ on a family of distributions $\mathcal{D}$ if for any distribution $D^L \in \mathcal{D}$, given a labelled dataset S^L of size n^L sampled i.i.d. from D^L and an unlabelled dataset S^U of size n^U sampled i.i.d. from any distribution D^U with η-bounded TV distance from D^L_X as well as third moment bounded by η, $\mathcal{A}$ is (ϵ,δ)-DP with respect to S^L and outputs a hypothesis h satisfying*

$$\mathbb{P}\left[\mathbb{P}_{(x,y)\sim D}\left[h(x)\neq y\right]\leq\alpha\right]\geq 1-\beta,$$

where the outer probability is over the randomness of the samples and the intrinsic randomness of the algorithm. In addition, the sample complexity n^L and n^U must be polynomial in $\frac{1}{\alpha}$ and $\frac{1}{\beta}$, and n^L must also be polynomial in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$.

In Theorem 4, we prove a full version of Theorem 2 that demonstrates PILLAR is an η-TV tolerant $(\alpha,\beta,\epsilon,\delta)$-semi-private learner for linear halfspaces $\mathcal{H}^d$. We define the scaled hinge loss that depends on η,ξ_0,γ_0 as

$$\ell(w;(x,y))=\max\left\{0,1-\frac{y\langle w,x\rangle}{\gamma_0\left(0.9(1-\xi_0)-\frac{14\eta}{\Delta_k}\right)}\right\}. \tag{D.18}$$

Theorem 4. *For $k\leq d\in\mathbb{N}$, $\gamma_0\in(0,1)$, $\xi_0\in[0,1)$, let $\mathcal{D}_{\gamma_0,\xi_0}$ be the family of distributions consisting of all (γ,ξ_k)-large margin low rank distributions over $\mathcal{X}_d\times\mathcal{Y}$ with $\gamma\geq\gamma_0$ and $\xi_k\leq\xi_0$ and third moment bounded by η. For any $\alpha\in(0,1)$, $\beta\in(0,1/4)$, $\epsilon\in(0,1/\sqrt{k})$, $\delta\in(0,1)$ and $\eta\in[0,9(1-\xi_0)\Delta_k/140)$, PILLAR with scaled hinge loss ℓ defined in Equation* (D.18)*, is an η-TV tolerant $(\alpha,\beta,\epsilon,\delta)$-semi-private learner of the linear halfspace $\mathcal{H}^d$ on $\mathcal{D}_{\gamma_0,\xi_0}$ with sample complexity*

$$n^U=O\left(\frac{\log\frac{2}{\beta}}{(\gamma_0\Delta_k)^2}\right),n^L=\tilde{O}\left(\frac{\sqrt{k}}{\alpha\epsilon\zeta}\right)$$

where $\Delta_k=\lambda_k\left(\Sigma^L\right)-\lambda_{k+1}\left(\Sigma^L\right)$ and $\zeta=\gamma_0\left(0.9(1-\xi_0)-14\eta/\Delta_k\right)$.

Proof. **Privacy Guarantee** A similar argument as the proof of the privacy guarantee in Theorem 1 shows that Algorithm $\mathcal{A}_{\epsilon,\delta}(k,\zeta)$ preserves (ϵ,δ)-DP on the labelled dataset S^L. We now focus on the accuracy guarantee.

Accuracy Guarantee For any unlabelled distribution D^U with η-bouneded TV distance from the labelled distribution D^L_{γ,ξ_k}, let the empirical covariance matrix of the unlabelled dataset S^U be $\widehat{\Sigma}^U=\frac{1}{n^U}\sum_{x\in S^U}xx^\top$ and $\hat{A}_k\in\mathbb{R}^{d\times k}$ be the projection matrix whose i^{th} column is the i^{th}

eigenvector of $\widehat{\Sigma}^U$. Let Σ^L and Σ^U be the population covariance matrices of the labelled and unlabelled distributions D^L and D^U. Similarly, let A_k^L and A_k^U be the matrices of top k eigenvectors of Σ^L and Σ^U respectively.

By definition, all distributions $D^L_{\gamma,\xi_k} \in \mathcal{D}_{\gamma_0,\xi_0}$ are (γ,ξ_k)-large margin low rank distribution, as defined in Definition 3, for some $\gamma \geq \gamma_0$, $\xi_k \leq \xi_0$. Let $w^\star$ be the large margin linear classifier that is guaranteed to exist by Definition 3. Then, for all $z \in \operatorname{supp} D^L_{X,(\gamma,\xi_0)}$, where $D^L_{X,(\gamma,\xi_0)}$ is the marginal distribution of D_{γ,ξ_k}, its margin is lower bounded by $\frac{y\left\langle \hat{A}_k^\top z, \hat{A}_k^\top w^\star \right\rangle}{\left\|\hat{A}_k^\top z\right\|_2 \left\|\hat{A}_k^\top w^\star\right\|_2}$. Similar to the proof of Lemma 2, we will first lower bound this term to show that, with high probability, the projected dataset still retains a large margin. Then, we will invoke existing algorithms in the literature with scaled hinge loss with the right parameters, to privately learn a large margin classifier in this low dimensional space.

First, let $z = a_z w^\star + b^\perp$ for some a_z where $b^\perp$ is in the nullspace of $w^\star$. Then, it is easy to see that using the large-margin property in Definition 3, we get

$$y a_z = \frac{\langle w^\star, z\rangle}{\|w^\star\|_2 \|z\|_2} \geq \gamma \geq \gamma_0. \tag{D.19}$$

Then, we lower bound $\dfrac{y\left\langle \hat{A}_k^\top z, \hat{A}_k^\top w^\star \right\rangle}{\left\|\hat{A}_k^\top z\right\|_2 \left\|\hat{A}_k^\top w^\star\right\|_2}$ as

$$\frac{y\left\langle \hat{A}_k^\top z, \hat{A}_k^\top w^\star \right\rangle}{\left\|\hat{A}_k^\top z\right\|_2 \left\|\hat{A}_k^\top w^\star\right\|_2} \overset{(a)}{=} \frac{y a_z \left\|\hat{A}_k^\top w^\star\right\|_2}{\left\|\hat{A}_k^\top z\right\|_2} \overset{(b)}{\geq} \gamma_0 \left\|\hat{A}_k^\top w^\star\right\|_2, \tag{D.20}$$

where step (a) is due to $\langle w^\star, b^\perp\rangle = 0$ and step (b) follows from $\left\|\hat{A}_k^\top z\right\|_2 \leq \left\|\hat{A}_k\right\|_{\text{op}} \|z\|_2 \leq 1$ and Equation (D.19). To lower bound $\left\|\hat{A}_k^\top w^\star\right\|_2$, we use the triangle inequality to decompose it as follows

$$\begin{aligned}
\left\|\hat{A}_k^\top w^\star\right\|_2 \geq& \left\|A_k^L \left(A_k^L\right)^\top w^\star\right\|_2 - \left\|\left(A_k^L \left(A_k^L\right)^\top - A_k^U \left(A_k^U\right)^\top\right) w^\star\right\|_2 - \left\|\left(A_k^U (A_k^U)^\top - \hat{A}_k (\hat{A}_k)^\top\right) w^\star\right\|_2 \\
\geq& \left\|A_k^L \left(A_k^L\right)^\top w^\star\right\|_2 - \left\|A_k^L (A_k^L)^\top - A_k^U \left(A_k^U\right)^\top\right\|_{\text{op}} \left\|w^\star\right\|_2 - \left\|A_k^U (A_k^U)^\top - \hat{A}_k (\hat{A}_k)^\top\right\|_F \left\|w^\star\right\|_2 \\
\geq& 1 - \xi_k - \left\|A_k^L (A_k^L)^\top - A_k^U \left(A_k^U\right)^\top\right\|_{\text{op}} - \left\|A_k^U (A_k^U)^\top - \hat{A}_k (\hat{A}_k)^\top\right\|_F
\end{aligned} \tag{D.21}$$

where the second inequality follows from applying Cauchy-Schwartz inequality on the second and third term and the third step follows from using the low rank separability assumption in Definition 3 on the first term and observing that $\left\|w^\star\right\|_2 = 1$.

Now, we need to bound the two terms $\left\|A_k^L (A_k^L)^\top - A_k^U \left(A_k^U\right)^\top\right\|_{\text{op}}$ and $\left\|A_k^U (A_k^U)^\top - \hat{A}_k (\hat{A}_k)^\top\right\|_F$. We bound the first term with Lemma 5.

Lemma 5 (Theorem 3 in [Zwald and Blanchard, 2005])**.** *Let $A \in \mathbb{R}^d$ be a symmetric positive definite matrix with nonzero eigenvalues $\lambda_1 > \lambda_2 > \ldots > \lambda_d$. Let $k > 0$ be an integer such that $\lambda_k > 0$. Let $B \in \mathbb{R}^d$ be another symmetric positive definite matrix such that $\|B\|_F < \frac{1}{4}(\lambda_k - \lambda_{k+1})$ and $A + B$ is still a positive definite matrix. Let $P_k(A), P_k(A+B)$ be the matrices whose columns consists of the first k eigenvectors of $A, A + B$, then*

$$\left\|P_k(A)P_k(A)^T - P_k(A+B)P_k(A+B)^T\right\|_F \leq \frac{2\|B\|_F}{\lambda_k - \lambda_{k+1}}.$$

It guarantees that with probability $1 - \beta/4$,

$$\left\|A_k^L\left(A_k^L\right)^\top - A_k^U\left(A_k^U\right)^\top\right\|_{\text{op}} \leq \frac{2\left\|\Sigma^L - \Sigma^U\right\|_{\text{op}}}{\lambda_k\left(\Sigma^L\right) - \lambda_{k+1}\left(\Sigma^L\right)} = \frac{2\left\|\Sigma^L - \Sigma^U\right\|_{\text{op}}}{\Delta_k}. \tag{D.22}$$

Then, we bound the term $\left\|\Sigma^L - \Sigma^U\right\|_{\text{op}}$ with Lemma 6.

Lemma 6. *Let f and g be the Probability Density Functions (PDFs) of two zero-mean distributions F and G over $\mathcal{X}$ with covariance matrices Σ_f and Σ_g respectively. Assume the spectral norm of the third moments of both F and G are bounded by η. If the total variation between the two distributions is bounded by η,i.e. $TV(f,g) = \max_{A\subset\mathcal{X}}\left|f(A) - g(A)\right| \leq \eta$, then the discrepancy in the covariance matrices is bounded by 7η, i.e. $\left\|\Sigma_f - \Sigma_g\right\|_{\text{op}} \leq 7\eta$.*

By applying Lemma 6 and the assumption of bounded total variation between the labelled and unlabelled distributions to Equation (D.22), we get

$$\left\|A_k^L\left(A_k^L\right)^\top - A_k^U\left(A_k^U\right)^\top\right\|_{\text{op}} \leq \frac{14\eta}{\lambda_k\left(\Sigma^L\right) - \lambda_{k+1}\left(\Sigma^L\right)} = \frac{14\eta}{\Delta_k}, \tag{D.23}$$

where Δ_k is defined as the difference between the k^{th} and $(k+1)^{th}$ eigenvalue of Σ^L.

Similar to the proof for Lemma 2, we upper bound the term $\left\|A_k^U(A_k^U)^\top - \hat{A}_k(\hat{A}_k)^\top\right\|_F$ using Lemma 4, which guarantees that with probability $1 - \beta/4$,

$$\left\|A_k^U\left(A_k^U\right)^\top - \hat{A}_k\hat{A}_k^\top\right\|_F \leq \frac{1-\xi_0}{10}, \tag{D.24}$$

where the inequality follows from choosing the size of unlabelled data $n^U = O\left(\frac{\log\frac{2}{\beta}}{((1-\xi_0)\Delta_k)^2}\right)$.

Substituting Equations (D.23) and (D.24) into Equation (D.21) and then plugging Equation (D.21) into Equation (D.20), we get that with probability at least $1 - \beta/2$, the margin in the projected space is lower bounded as

$$\frac{y\left\langle\hat{A}_k^\top z, \hat{A}_k^\top w^\star\right\rangle}{\left\|\hat{A}_k^\top z\right\|_2\left\|\hat{A}_k^\top w^\star\right\|_2} \geq \gamma_0\left(0.9(1-\xi_0) - \frac{14\eta}{\Delta_k}\right).$$

Thus, the (scaled) hinge loss function ℓ defined in Equation (D.18) is $\frac{1}{\gamma_0(0.9(1-\xi_0)-14\eta/\Delta_k)}$-Lipschitz. For a halfspace with parameter $v \in B_2^k$, denote the empirical hinge loss on a dataset S by $\hat{L}(w;S) = \frac{1}{|S|}\sum_{(x,y)\in S}\ell(w,(x,y))$ and the loss on the distribution D by $L(w;D) = \mathbb{E}_{(x,y)\sim D}[\ell(w,(x,y))]$. Let D_k be the k-dimension transformation of the original distribution D by projecting each $x \in \mathcal{X}$ to $\hat{A}_k^\top x$. By the convergence bound in Lemma 1 for $\mathcal{A}_{\text{Noisy-SGD}}$, we have with probability $1-\frac{\beta}{4}$, $\mathcal{A}_{\text{Noisy-SGD}}$ outputs a hypothesis $v_k \in B_2^k$ such that

$$\hat{L}(v_k;S_k^L) - \hat{L}(v_k^\star;D_k) = \hat{L}(v_k;S_k^L) = \tilde{O}\left(\frac{\sqrt{k}}{n^L\epsilon\gamma_0\left(0.9(1-\xi_0)-14\eta/\Delta_k\right)}\right),$$

where $v_k^\star = \text{argmin}_{v\in B_2^k}\hat{L}(v;S_k^L)$ and $\hat{L}(v_k^\star;S_k^L) = 0$ as the margin in the transformed low-dimensional space is at least $\gamma_0\left(0.9(1-\xi_0)-\frac{14\eta}{\Delta_k}\right) > 0$ for $\eta \le \frac{9(1-\xi_0)\Delta_k}{140}$.
For $n^L = O\left(\frac{\sqrt{k}}{\alpha\beta\gamma_0(0.9(1-\xi_0)-14\eta/\Delta_k)}\text{polylog}\left(\frac{1}{\delta},\frac{1}{\epsilon},\frac{1}{\beta},\frac{1}{\alpha},\frac{1}{\gamma_0},\frac{1}{\xi_0},k,n^L\right)\right)$, we can bound the emiprical 0-1 error with probability $1-\frac{\beta}{4}$,

$$\frac{1}{n^L}\sum_{(x,y)\in S_k^L}\mathbb{I}\{y\langle v_k,x\rangle < 0\} \le \hat{L}(v_k;S_k^L) = \tilde{O}\left(\frac{\sqrt{k}}{n^L\epsilon\gamma_0\left(0.9(1-\xi_0)-14\eta/\Delta_k\right)}\right) \le \frac{\alpha}{4}. \tag{D.25}$$

It remains to bound the generalisation error of linear halfspace $\mathcal{H}^k$. We use Lemma 7 for upper bounding this term.

Lemma 7 (Convergence bound on generalisation error [Anthony and Bartlett, 1999])**.** *Suppose $\mathcal{H}$ is a hypothesis class with instance space $\mathcal{X}$ and output space $\{-1,1\}$. Let D be a distribution over $\mathcal{X}\times\mathcal{Y}$ and S be a dataset of size n sampled i.i.d. from D. For $\eta \in (0,1), \zeta > 0$, we have*

$$\mathbb{P}_{S\sim D^n}\left[\sup_{h\in\mathcal{H}} L(h;D) - (1+\zeta)\hat{L}(h;S) > \eta\right] \le 4\Pi_{\mathcal{H}}(2n)\exp\left(-\frac{\eta\zeta n}{4(\zeta+1)}\right),$$

where L and $\hat{L}$ are the population and the empirical 0-1 error and $\Pi_{\mathcal{H}}$ is the growth function of $\mathcal{H}$.

Setting $\zeta = 1$ and $\eta = \frac{\alpha}{2}$ in Lemma 7 gives us that with probability $1-\frac{\beta}{4}$,

$$\mathbb{P}_{(x,y)\sim D_k}[y\langle v_k,x\rangle < 0] - \frac{2}{n^L}\sum_{(x,y)\in S_k^L}\mathbb{I}\{y\langle v_k,x\rangle < 0\} \le \frac{\alpha}{2}. \tag{D.26}$$

Thus, combining Equations (D.25) and (D.26) we get

$$\mathbb{P}_{(x,y)\sim D}\left[y\left\langle v_k,\hat{A}_k^\top x\right\rangle < 0\right] = \mathbb{P}_{(x,y)\sim D_k}[y\langle v_k,x\rangle < 0] \le \frac{2}{n^L}\sum_{(x,y)\in S_k^L}\mathbb{I}\{y\langle v_k,x\rangle < 0\} + \frac{\alpha}{2} = \alpha,$$

for $n^L \geq \frac{k}{\alpha}\text{polylog}\left(\frac{1}{\beta}, \frac{1}{k}\right)$. This is equivalent as stating that the output of Algorithm 1 $\hat{w} = \hat{A}_k v_k$ satisfies

$$\mathbb{P}_{(x,y)\sim D}\left[y\left\langle \hat{w}, x\right\rangle < 0\right] = \mathbb{P}_{(x,y)\sim D}\left[y\left\langle \hat{A}_k v_k, x\right\rangle < 0\right] = \mathbb{P}_{(x,y)\sim D}\left[y\left\langle v_k, \hat{A}_k^\top x\right\rangle < 0\right] \leq \alpha,$$

which concludes the proof. □

Proof of Lemma 6. We first approximate Moment Generating Functions (MGFs) of g and f by their first and second moments. Then, we express the error bound in this approximation by the error bound for Taylor expansion, for any $t \in \mathbb{R}^d$ with $\|t\|_2 > 0$,

$$\begin{aligned}\left|M_f(t) - 1 + t^T\mathbb{E}_f[X] + \frac{t^T\Sigma_f t}{2}\right| &\overset{(a)}{\leq} \frac{\mathbb{E}_f\left[e^{t^T x}xx^T x\right]\|t\|_2^3}{3!} \\ &\overset{(b)}{\leq} \frac{\mathbb{E}_f\left[xx^T x\right]e^{\|t\|_2}\|t\|_2^3}{3!} \\ &\overset{(c)}{\leq} \eta\|t\|_2^3\end{aligned} \tag{D.27}$$

where step (a) follows by the error bound of Taylor expansion, step (b) is due to $e^{t^T x} \leq e^{\|t\|_2\|x\|_2} \leq e^{\|t\|_2}$ for all $x \in B_d^2$, and step (c) follows from $e^{\|t\|_2} \leq 3!$ for $\|t\|_2 \leq 1$. Similarly,

$$\left|M_g(t) - 1 + t^T\mathbb{E}_g[X] + \frac{t^T\Sigma_g t}{2}\right| \leq \eta\|t\|_2^3. \tag{D.28}$$

Rewrite Equation (D.27) and Equation (D.28) and observing that $\mathbb{E}_g[X] = \mathbb{E}_f[X] = 0$, we can bound the terms $\frac{t^T\Sigma_f t}{2}$ and $\frac{t^T\Sigma_g t}{2}$ by

$$\begin{aligned}1 - M_f(t) - \eta\|t\|_2^3 \leq \frac{t^T\Sigma_f t}{2} \leq 1 - M_f(t) + \eta\|t\|_2^3 \\ 1 - M_g(t) - \eta\|t\|_2^3 \leq \frac{t^T\Sigma_g t}{2} \leq 1 - M_g(t) + \eta\|t\|_2^3.\end{aligned} \tag{D.29}$$

Next, we show that the discrepancy in covariance matrices of distributions G and F are upper bounded by the difference in their MGFs. By Equation (D.29), for all $t \in \mathbb{R}^d$ and $\|t\|_2 \neq 0$,

$$\begin{aligned}\left|\frac{t^T\left(\Sigma_f - \Sigma_g\right)t}{2}\right| &\leq 1 - M_f(t) + \eta\|t\|_2^3 - 1 + M_g(t) + \eta\|t\|_2^3 \\ &= \left|M_g(t) - M_f(t) + 2\eta\|t\|_2^3\right| \\ &\leq \left|M_g(t) - M_f(t)\right| + 2\eta\|t\|_2^3\end{aligned} \tag{D.30}$$

Next, we upper bound the difference between the MGFs of distributions G and F by the TV distance between them.

$$\begin{aligned}\left|M_f(t) - M_g(t)\right| &= \left|\int_{x\in B_2^d} e^{t^T x}\left[f(x) - g(x)\right] dx\right| \\ &\leq \int_{x\in B_2^d} e^{t^T x}\left|f(x) - g(x)\right| dx \\ &\leq \int_{x\in B_2^d} e^{\|t\|_2\|x\|_2}\left|f(x) - g(x)\right| dx \leq \frac{e^{\|t\|_2}\eta}{2}\end{aligned} \tag{D.31}$$

where the last inequality follows as $\|x\|_2 = 1$ for $x \in B_2^d$ and $TV(f,g) \leq \eta$.

Combine Equation (D.30) and Equation (D.31), we have for all $t \in \mathbb{R}^d$ and $\|t\|_2 \neq 0$,

$$\left|t^T\left(\Sigma_f - \Sigma_g\right)t\right| \leq e^{\|t\|_2}\eta_1 + 4\eta\|t\|_2^3 \tag{D.32}$$

Choose t as a vector in the direction of the first eigenvector (*i.e.* the eigenvector corresponding to the largest eigenvalue) of $\Sigma_f - \Sigma_g$. For t in this direction, by the definition of operator norm,

$$\left\|\Sigma_f - \Sigma_g\right\|_{\text{op}} = \frac{\left|t^T\left(\Sigma_f - \Sigma_g\right)t\right|}{\|t\|_2}. \tag{D.33}$$

Plugging Equation (D.33) into Equation (D.32) and choose the norm of t as the minimizer of $e^{\|t\|_2}\eta_1 + 4\eta\|t\|_2^3$, we get

$$\left\|\Sigma_f - \Sigma_g\right\|_{\text{op}} \leq \min_{0\leq\|t\|_2\leq 1} \frac{e^{\|t\|_2}\eta}{\|t\|_2^2} + 4\eta\|t\|_2 \leq \frac{\eta(1+\|t\|_2+\|t\|_2^2)}{\|t\|_2^2} + 4\eta\|t\|_2 = 7\eta$$

This conclude the proof. □

D.1.5 Large margin Gaussian mixture distributions

In this section, we present in Example 1 a class of Large margin Gaussian mixture distributions that satisfies the large-margin low rank assumption. For any $\theta, \sigma^2 = O(1/\sqrt{d})$, it is easy to see that this family of distributions satisfies the large margin low rank properties in Definition 3 for $k = 2$ and $\xi_k = 0$.

Example 1. *A distribution D over $\mathcal{X}\times\mathcal{Y}$ is a (θ,σ^2)-Large margin Gaussian mixture distribution if there exists $w^\star, \mu \in B_2^d$, such that $\langle\mu, w^\star\rangle = 0$, the conditional random variable $X|y$ is distributed according to a normal distribution with mean μy and covariance matrix $\theta w^\star\left(w^\star\right)^\top + \sigma^2 I_d$ and $y \in \{-1,1\}$ is distributed uniformly.*

We present Corollary 2 following Theorem 1, which shows that for large margin Gaussian mixture distributions, PILLAR leads to a drop in the private sample complexity from $O(\sqrt{d})$ to $O(1)$.

Corollary 2 (Theoretical guarantees for large margin Gaussian mixture distribution)**.** *For $\theta, \sigma^2 = \tilde{O}\left(1/\sqrt{d}\right)$, let $\mathcal{D}_{\theta,\sigma^2}$ be the family of all (θ, σ^2)-large margin Gaussian mixture distribution (Example 1). For any $\alpha \in (0,1), \beta \in (0, 1/4), \epsilon \in (0, 1/\sqrt{M})$, and $\delta \in (0,1)$, PILLAR $\mathcal{A}_{\epsilon,\delta}(k,\ell)$ with scaled hinge loss defined in Table 5.3.1 is an $(\alpha, \beta, \epsilon, \delta)$-semi-private learner on D_{θ,σ^2} of linear halfspaces $\mathcal{H}^d$ with sample complexity*

$$\begin{aligned} n^U &= O\left(\frac{M^2 \log \frac{2}{\beta}}{\gamma^2 \theta^2}\right), \\ n^L &= \tilde{O}\left(\frac{M\sqrt{k}}{\alpha\epsilon\gamma(1-0.1\gamma)}\right) \end{aligned} \tag{D.34}$$

where $\gamma = 1 - \left(4\sqrt{d} + 2\sqrt{\log \frac{2n^L}{\delta}}\right)\left(\sigma^2 + \theta\right)$, $M = 1 + \left(4\sqrt{d} + 2\sqrt{\log \frac{2n^L}{\delta}}\right)\left(\sigma^2 + \theta\right)$.

Here, in line with the notation of Definition 3, γ intuitively represents the margin in the d-dimensional space and M is the upper bound for the radius of the labelled dataset. For $\theta = \sigma^2 = 1/2C\sqrt{d}$ and ignoring the logarithmic terms, we get $M = 1.5$ and $\gamma = 0.5$. Corollary 2 implies the labelled sample complexity $\tilde{O}\left(1/\alpha\epsilon\right)$.

Proof. To prove this result, we first show that all large-margin Gaussian mixture distributions $D_{\theta,\sigma^2} \in \mathcal{D}_{\theta,\sigma^2}$ are (γ_0, ξ)-large margin low rank distribution (Definition 3) after normalization. In particular, we show that the normalized distribution is (γ_0, ξ)-large margin low rank distribution with $\xi = 0$ and margin $\gamma_0 = \gamma/M$, where $\gamma = 1 - \left(4\sqrt{d} + 2\sqrt{\log \frac{2n^L}{\delta}}\right)\left(\sigma^2 + \theta\right)$ and $M = 1 + \left(4\sqrt{d} + 2\sqrt{\log \frac{2n^L}{\delta}}\right)\left(\sigma^2 + \theta\right)$. Then, invoking Theorem 1 gives the desired sample complexity in Equation (D.34).

To normalize the distribution, we consider the marginal distribution D_X of the mixture distribution $D \in \mathcal{D}_{\theta,\sigma^2}$ and compute its mean and the covariance matrix. By Example 1, D is a mixture of two gaussians with identical covariance matrix $\Sigma = \theta w^\star (w^\star)^\top - \sigma^2 I_d$ and means $\mu_1 = -\mu_2$.With a slight misuse of notation, we denote the probability density function of a normal distribution with mean μ and covariance Σ using $\mathcal{N}(x; \mu, \Sigma)$. Then, we can calculate the mean and covariance matrix as

$$\mathbb{E}_X[X] = \mathbb{E}_y \mathbb{E}_{X|y}[X|y] = \frac{1}{2}\mu_1 + \frac{1}{2}\mu_2 = 0 \tag{D.35}$$

and

$$\begin{aligned}
\Sigma_X &= \mathbb{E}_X\left[XX^\top\right] - \left(\mathbb{E}_X\left[X\right]\right)\left(\mathbb{E}_X\left[X\right]\right)^\top \overset{(a)}{=} \mathbb{E}_y\mathbb{E}_{X|y}\left[XX^\top|y\right] \\
&= \frac{1}{2}\int_{B_2^d} xx^\top \mathcal{N}(x;\mu_1,\Sigma)dx + \frac{1}{2}\int_{B_2^d} xx^\top \mathcal{N}(x;\mu_2,\Sigma)dx \\
&\overset{(b)}{=} \frac{1}{2}\left(\Sigma + \mu_1\mu_1^\top\right) + \frac{1}{2}\left(\Sigma + \mu_2\mu_2^\top\right) \\
&\overset{(c)}{=} \theta w^\star \left(w^\star\right)^\top + \mu_1\mu_1^\top + \sigma^2 I_d
\end{aligned}$$

where step (a) follows by Equation (D.35), step (b) follows by the relationship between covaraince matrix and the second moment $\Sigma = \mathbb{E}_X\left[XX^\top\right] - \mu\mu^\top$, and step (c) follows by the definition of large-margin Gaussian mixture distribution (Example 1) of Σ and μ_1, μ_2.

Then, we show that the first two eigenvectors are μ_1 and $w^\star$ with the corresponding eigenvalues $1+\sigma^2$ and $\theta+\sigma^2$ for $\theta = O\left(1/\sqrt{d}\right) \leq 1$. The remaining non-spiked eigenvalues are σ^2.

$$\begin{aligned}
\Sigma_X\mu_1 &= \theta w^\star \left(w^\star\right)^\top \mu_1 + \mu_1\mu_1^\top\mu_1 + \sigma^2\mu_1 \\
&\overset{(a)}{=} (\|\mu_1\|_2^2 + \sigma^2)\mu_1 = (1+\sigma^2)\mu_1 \\
\Sigma_X w^\star &= \theta w^\star \left(w^\star\right)^\top w^\star + \mu_1\mu_1^\top w^\star + \sigma^2 w^\star \\
&\overset{(b)}{=} (\theta + \sigma^2)w^\star,
\end{aligned}$$

where step (a) and (b) both follow from the fact that $\left(w^\star\right)^\top \mu_1 = 0$. For $k = 2$, it follows immediately that $\Delta_k = \theta$ (Equation (D.36)) and $\xi = 0$ (Equation (D.37)),

$$\Delta_k = \lambda_k\left(\Sigma_X\right) - \lambda_{k+1}\left(\Sigma_X\right) = \theta + \sigma^2 - \sigma^2 = \theta. \tag{D.36}$$

$$\begin{aligned}
\frac{\left\|A_k^\top w^\star\right\|_2}{\|w^\star\|_2} &= \frac{1}{\|w^\star\|_2}\begin{bmatrix}\mu_1^\top \\ \left(w^\star\right)^\top\end{bmatrix} w^\star \\
&= \frac{\left|\mu_1^\top w^\star + \left(w^\star\right)^\top w^\star\right|}{\|w^\star\|_2} \\
&\overset{(a)}{=} 1 = 1 - \xi,
\end{aligned} \tag{D.37}$$

where step (a) follows from $\mu_1^\top w = 0$.

Next, we show that the labelled dataset lies in a ball with bounded radius with high probability, which further implies that original data has a large margin.

Denote the part of the dataset from the gaussian component with $y = 1$ by S_1^L and denote the part from the component with $y = -1$ by S_2^L. We apply the well-known concentration bound on the norm of Gaussian random vectors (Lemma 8) to show a high probability upper bound on the radius of the datasets S_1^L and S_2^L.

Lemma 8 ([Vershynin, 2018]). *Let $X \sim N(\mu, \Sigma)$, where $v \in B_d^2$. Then, with probability at least $1-\delta$,*

$$\|X-\mu\|_2 \leq 4\,\|\Sigma\|_{\text{op}}\ \sqrt{d} + 2\,\|\Sigma\|_{\text{op}}\ \sqrt{\log\frac{1}{\delta}}.$$

This gives the following high probability upper bound on any $x \in S_i^L$ for $i = 1, 2$ and some $\frac{\beta}{2n^L} > 0$,

$$\mathbb{P}_{S^L \sim D^{nL}}\left[\|x-\mu_i\|_2 \leq 4\left(\theta+\sigma^2\right)\ \sqrt{d} + 2\left(\theta+\sigma^2\right)\ \sqrt{\log\frac{4n^L}{\beta}}\right] \geq 1 - \frac{\beta}{4n^L}$$

For $i \in \{1, 2\}$, by applying union bound on all $x \in S_i^L$, we can bound maximum distance of a points $x \in S_i^L$ to the center μ_i,

$$\mathbb{P}_{S^L \sim D^{nL}}\left[\max_{x \in S_i^L}\|x-\mu_i\|_2 \leq \left(\theta+\sigma^2\right)\left(4\,\sqrt{d} + 2\sqrt{\log\frac{4n^L}{\beta}}\right)\right] \leq 1 - \frac{\beta}{4}.$$

Note that the distance between the two centers μ_1 and μ_2 is 2. Thus, with probability at least $1-\frac{\beta}{2}$, all points in the labelled dataset S^L lie in a ball centered at 0 having radius

$$M = 1 + \left(4\,\sqrt{d} + 2\sqrt{\log\frac{4n^L}{\beta}}\right)\left(\sigma^2+\theta\right).$$

Also, the margin in the original labelled dataset is at least

$$\gamma = 1 - \left(4\,\sqrt{d} + 2\sqrt{\log\frac{4n^L}{\beta}}\right)\left(\sigma^2+\theta\right).$$

Normalizing the data by M, it is obvious that the normalized distribution satisfies the definition of (γ, ξ)-large margin low rank distribution with parameters $\xi = 0$, $\Delta_k = \theta/M$ and $\gamma_0 = \gamma/M$, where $\gamma = 1 - \left(4\,\sqrt{d} + 2\sqrt{\log\frac{2n^L}{\delta}}\right)\left(\sigma^2+\theta\right)$, $M = 1 + \left(4\,\sqrt{d} + 2\sqrt{\log\frac{2n^L}{\delta}}\right)\left(\sigma^2+\theta\right)$. Invoking Theorem 1 concludes the proof. □

D.1.6 Discussion of assumptions for existing methods

Analysis of the Restricted Lipschitz Continuity (RLC) assumption [Li et al., 2022b] DP-SGD [Li et al., 2022b] achieves dimension independent sample complexity if the following assumption, known as Restricted Lipschitz Continuity (RLC) is satisfied. For some $k \ll d$,

$$\sum_{i=1}^{\lfloor \log(d/k)+1 \rfloor} G_{2^{i-1}k}^2 \leq O(\sqrt{k/d}), \tag{RLC 1}$$

where $G_0, G_1, ..., G_d$ represent the RLC coefficients. For any $i \in [d]$, the loss function ℓ is said to satisfy RLC with coefficient G_i if

$$G_i \geq \min_{\substack{\operatorname{rank} P_i = i \\ P_i \in \Pi}} \left\| (I - P_i) \nabla \ell(w; (x, y)) \right\|_2, \tag{D.38}$$

for all $w, x, y \in \operatorname{domain}(\ell)$, where Π is the set of orthogonal projection matrices. Equivalently, assumption RLC 1 states that for some $k \ll d$,

$$\sum_{i=k+1}^{d} G_i^2 \leq O(\sqrt{k/d}). \tag{RLC}$$

In this section, we demonstrate that if we assume the Restricted Lipschitz Continuity (RLC) condition from [Li et al., 2022b], our low rank separability assumption on $\left\| A_k A_k^\top w^\star \right\|$ holds for large-margin linear halfspaces. However, using the RLC assumption leads to a looser bound compared to our assumption. More specifically, given the RLC assumption and the loss function ℓ defined in Table 5.3.1, we can show $\left\| A_k A_k^\top w^\star \right\| \geq \gamma$.
Given the parameter ζ in Algorithm 1, for $x, y \in \operatorname{supp} D$ and w satisfying $y \langle w, x \rangle \leq \zeta$, we can calculate the i^{th} restricted Lipschitz coefficient

$$\begin{aligned} G_i &\geq \min_{\substack{\operatorname{rank} P_i = 1 \\ P_i \in \Pi}} \left\| (I - P_i) \nabla \ell(w; (x, y)) \right\|_2 \\ &= \min_{\substack{\operatorname{rank} P_i = 1 \\ P_i \in \Pi}} \left\| \frac{y}{\zeta} (I - P_i) x \right\|_2 \\ &= \min_{\substack{\operatorname{rank} P_i = 1 \\ P_i \in \Pi}} \left\| \frac{1}{\zeta} (x - P_i x) \right\|_2. \end{aligned} \tag{D.39}$$

Equivalently, we can rewrite Equation (D.39) as there exists a rank-i orthogonal projection matrix $P_i^{\min}$ such that

$$\left\| x - P_i^{\min} x \right\|_2 \leq \zeta G_i. \tag{D.40}$$

Thus, for x such that $y \langle w, x \rangle \leq \zeta$,

$$\begin{aligned} \left\| x x^\top - (P_i^{\min} x)(P_i^{\min} x)^\top \right\|_{\text{op}} &\overset{(a)}{=} \left\| (x - P_i^{\min} x)(x + P_i^{\min} x)^\top \right\|_2 \\ &\leq \left\| x + P_i^{\min} x \right\|_2 \left\| x - P_i^{\min} x \right\|_2 \\ &\overset{(b)}{\leq} 2 \left\| x - P_i^{\min} x \right\|_2 \\ &\overset{(c)}{\leq} 2 G_i \zeta \end{aligned} \tag{D.41}$$

where step (a) follows from the orthogonality of $P_i^{\min}$, step (b) follows from $\left\|P_i^{\min}x\right\|_2 \leq \|x\|_2 = 1$, and step (c) follows from Equation (D.40).

Then, we can bound the low-rank approximation error for the covariance matrix of the data distribution.

$$\left\|\Sigma_X - P_i^{\min}\Sigma_X(P_i^{\min})^\top\right\|_{\text{op}} \overset{(a)}{\leq} \mathbb{E}_{x\sim D_X}\left(\left\|xx^\top - (P_i^{\min}x)(P_i^{\min}x)^\top\right\|_{\text{op}}\right) \overset{(b)}{\leq} 2G_i\zeta.$$

where $\Sigma_X = \mathbb{E}_{x\sim D_X}\left[xx^\top\right]$, and step (a) follows from the convexity of the Euclidean norm and step (b) follows from Equation (D.41).

This further provides an upper bound on the last $d-k$ eigenvalues of the covariance matrix Σ_X of the data distribution D_X. Let λ_i denote the i^{th} eigenvalue of the covariance matrix Σ_X. Then, we apply Lemma 9 that gives an upper bound on the singular values of a matrix in terms of the rank k approximation error of the matrix.

Lemma 9 ([Gharan, 2017])**.** *For any matrix $M \in \mathbb{R}^{m\times n}$,*

$$\inf_{\text{rank}\,\hat{M}=k}\left\|M-\hat{M}\right\|_{\text{op}} = \sigma_{k+1},$$

where the infimum is over all rank k matrices $\hat{M}$ and σ_{k+1} is the k^{th} singular value of the matrix M.

This gives an upper bound on the i^{th} eigenvalue of the covariance matrix Σ_X in terms of the i^{th} restricted Lipschitz coefficient,

$$\lambda_{i+1} = \sigma_{i+1}^2 = \inf_{\text{rank}\,\Sigma'_X=i}\left\|\Sigma_X - \Sigma'_X\right\|_{\text{op}}^2 \leq \left\|\Sigma_X - P_i^{\min}\Sigma_X(P_i^{\min})^\top\right\|_{\text{op}}^2 \leq 4G_i^2\zeta^2.$$

Thus, for matrix A_k consisting of the first k eigenvectors of Σ_X, we can upper bound the reconstruction error of $A_k^\top x$ with the eigenvalues of the covariance matrix Σ_X,

$$\begin{aligned}\mathbb{E}_{x\sim D_X}\left[\|x\|_2 - \left\|A_k^\top x\right\|_2\right] &= \mathbb{E}_{x\sim D_X}\left[\left\|xx^\top\right\|_{\text{op}} - \left\|(A_kx)(A_kx)^\top\right\|_{\text{op}}\right] \\ &\leq \mathbb{E}_{x\sim D_X}\left[\left\|xx^\top - (A_kx)(A_kx)^\top\right\|\right]_{\text{op}} \leq \sum_{i=k+1}^{d}\lambda_i \leq 4\zeta^2\sum_{i=k+1}^{d}G_i^2.\end{aligned}$$

By Markov's inequality, with probability at least $1-\beta$,

$$\begin{aligned}&\mathbb{P}_{x\sim D_X}\left[\left\|xx^\top\right\|_{\text{op}} - \left\|(A_k^\top x)(A_k^\top x)^\top\right\|_{\text{op}} \geq \frac{4\zeta^2}{\beta}\sum_{i=k+1}^{d}G_i^2\right] \\ &\leq \mathbb{P}_{x\sim D_X}\left[\left\|xx^\top - (A_k^\top x)(A_k^\top x)^\top\right\|_{\text{op}} \leq \frac{4\zeta^2}{\beta}\sum_{i=k+1}^{d}G_i^2\right] \leq \beta.\end{aligned} \tag{D.42}$$

This implies our assumption with probability at least $1-\beta$,

$$\begin{aligned}\left\|A_k A_k^\top w^\star\right\|_2 &\overset{(a)}{=} \|x\|_2 \left\|A_k A_k^\top w^\star\right\|_2 \geq |\left\langle A_k A_k^\top x, w^\star\right\rangle| \\ &\overset{(b)}{\geq} |\left\langle x, w^\star\right\rangle| - |\left\langle x - A_k A_k^\top x, w^\star\right\rangle| \\ &\overset{(c)}{\geq} \gamma - \left\|x - A_k A_k^\top x\right\|_2 \left\|w^\star\right\|_2 \\ &\overset{(d)}{\geq} \gamma - \frac{4\zeta^2}{\beta} \sum_{i=k+1}^{d} G_i^2\end{aligned} \tag{D.43}$$

where step (a) follows from $\|x\|_2 = 1$, step (b) follows by $\left\langle A_k A_k^\top x, w^\star\right\rangle = \langle x, w^\star\rangle - \left\langle x - A_k A_k^\top x, w^\star\right\rangle$ and the triangle inequality, step (c) follows by the large margin assumption $y\langle x, w^\star\rangle = \left|\langle x, w^\star\rangle\right| \geq \gamma$, and step (d) follows by Equation (D.42) with probability at least $1-\beta$.

The RLC assumption requires the last term in Equation (D.43) to vanish at the rate of $O(k/d)$. This implies our low-rank assumption holds with $\xi = 1-\gamma$.

Analysis on the error bound for GEP To achieve a dimension-independent sample complexity bound in GEP [Yu et al., 2021a], the gradient space must satisfy a low-rank assumption, which is even stronger than the rapid decay assumption in RLC coefficients (Equation (RLC 1)). By following a similar argument as the analysis for the RLC assumption [Li et al., 2022b], we can demonstrate that our low-rank assumption is implied by the assumption in GEP.

D.2 Experimental details and additional experiments

D.2.1 Details and hyperparameter ranges for our method

Unless stated otherwise, we use the PRV accountant [Gopi et al., 2021a] in our experiments. Following [De et al., 2022], we use the validation data for cross-validation of the hyperparameters in all of our experiments and set the clipping constant to 1. We search the learning rate in $\{0.01, 0.1, 1\}$, use no weight decay nor momentum as we have seen it to have little or adverse impact. We search the number of steps in $\{500, 1000, 3000, 5000, 6000\}$ and our batch size in $\{128, 512, 1024\}$. We compute the variance of the noise as a function of the number of steps and the target ϵ using `opacus`. We set $\delta = 1e-5$ in all our experiments. We use the open-source `opacus` [Yousefpour et al., 2021] library to run DP-SGD with the PRV Accountant efficiently. We use `scikit-learn` to implement PCA. Checkpoints of ResNet-50 are taken or trained using the `timm` [Wightman, 2019a] and `solo-learn` [da Costa et al., 2022] libraries. Standard ImageNet pre-processing of images is applied, without augmentations.

Privacy	CIFAR10		CIFAR100	
	Ours	**[De et al., 2022]**	**Ours**	**[De et al., 2022]**
$\epsilon = 0.1$	89.4	-	36.1	-
$\epsilon = 0.7$	93.1	-	69.7	-
$\epsilon = 1$	93.5	93.1	71.8	70.3
$\epsilon = 2$	93.9	93.6	74.9	73.9

Table D.2.1: Result for our algorithm is with pre-training on ImageNet32x32. Results for [De et al., 2022] is taken from their paper where available.

D.2.2 Discrepancy in pre-training resolution

Several works have used different resolutions of ImageNet to pre-train their models. In particular, [De et al., 2022] used ImageNet 32x32 to pre-train their model, which is a non-standard dimensionality of ImageNet, but it matches the dimensionality of their private dataset CIFAR-10. In contrast, we use the standard ImageNet (224x224) for pre-training in all our experiments with both CIFAR datasets as well as other datasets. In this section, we show that using the resolution of 32x32 for pre-training, we can indeed outperform [De et al., 2022] but also highlight why this may not be suitable for privacy applications.

Low-resolution (CIFAR specific) pre-training Different private tasks/datasets may have images of differing resolutions. While all images in CIFAR [Krizhevsky, 2009] are 32x32 dimensional, in other datasets, images not only have higher resolution but their resolution varies widely. For example, GTSRB [Houben et al., 2013] has images of size 222x193 as well as 15x15 , PCAM [Veeling et al., 2018] has 96x96 dimensional images, most images in Dermnet [Der, 2019] have resolution larger than 720x400, and in Pneumonia [Kermany et al., 2018] most x-rays have a dimension higher than 2000x2000. Therefore, it may not be possible to fine-tune the feature extractor at a single resolution for such datasets.

Identifying the optimal pre-training resolution for each private dataset is beyond the scope of our work and orthogonal to the contributions of our work (as we extensively show, our method PILLAR operates well under several pre-training strategies in Figure D.2.3 and Figure D.2.5). Furthermore, assuming the pre-training and private dataset resolution to be perfectly aligned is a strong assumption.

Comparison with [De et al., 2022] Nevertheless, we compare our approach with [De et al., 2022] pre-training a ResNet50 on a 32x32 rescaled ImageNet version, and obtain a non-private accuracy

larger than 94% reported for $\epsilon = 8$ in Table 5 in [De et al., 2022] for *Classifier training*. Note that our approaches is computationally significantly cheaper than theirs as we do no use the tricks proposed in their work (including Augmult, EMA, and extremely large batch sizes (> 16K))

Using ImageNet32x32 for pre-training, we perform slightly better than them in private training. Our results are reported in Table D.2.1. We expect that applying their techniques will result in even higher accuracies at the cost of computational efficiency. Interestingly, Table D.2.1 shows that our model's accuracy for $\epsilon = 0.7$ on CIFAR10, is as good as [De et al., 2022] for $\epsilon = 1.0$. This provides evidence that large batch sizes, which is one of the main hurdles in producing deployable private machine learning models, might not be required using our approach.

D.2.3 Experiments with large ϵ (≥ 1)

While in most of the paper, we focus on settings with small ϵ, in certain practical settings, the large epsilon regime may also be important. In Table D.2.2, we repeat our experiments for CIFAR10 and CIFAR100 with $\epsilon \in 1, 2$ and report the accuracy for the best projection dimension. Our results show that for $\epsilon \in \{0.1, 0.7, 1, 2\}$ our method can provide significant gains on the challenging dataset of CIFAR-100; however for CIFAR-10 with $\epsilon = 1, 2$ the improvements are more modest.

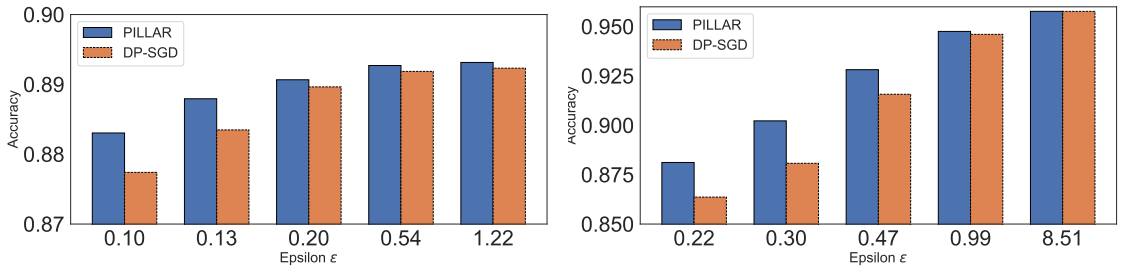
D.2.4 Comparison with PATE

We now discuss discuss the *PATE* family of approaches [Papernot et al., 2017, 2018, Zhu et al., 2020, Mühl and Boenisch, 2022]. These methods partition the training set into disjoint subsets, train an ensemble of teacher models on them, and use them to pseudo-label a public dataset using a privacy-preserving mechanism. For PATE to provide tight privacy guarantees, a large number (150-200 [Papernot et al., 2018]) of subsets is needed, which reduces the test accuracy of each teacher. Large amounts of public data are also required. For CIFAR-10, [Papernot et al., 2018, Zhu et al., 2020] use 29000 examples (58% of training set size), whereas we only use 5000 (10% of training set size) public unlabelled data points (and to retain its accuracy, in Section 5.5.2 we show 500 (1%) samples are sufficient). Of these 29000 examples, [Zhu et al., 2020] reports only half of them is labelled due to the private labelling mechanism, further limiting the student's performance in settings with low amounts of public training data. Despite our best attempts, we could not train PATE-based approaches in our challenging setting to satisfactory levels of accuracy on either CIFAR-10 or CIFAR-100.[3]

[3]For reference, we refer the reader to the accuracies reported for the state-of-the-art implementation in [Boenisch et al., 2023] (Table 12) and [Zhu et al., 2020] (Table 1), which are less than 40% and 75% respectively, whereas we obtain more than 85% for tighter privacy guarantees.

Privacy	Pre-training	CIFAR10		CIFAR100	
		Ours	No Projection	**Ours**	No Projection
$\epsilon = 1$	SL	86.4	85.4	58.8	54.4
	SSL	81.4	80.5	49.0	45.8
$\epsilon = 2$	SL	86.8	86.4	61.8	60.0
	SSL	82.5	81.9	53.03	50.06

Table D.2.2: Experiment with larger ϵ. Pre-training is with ImageNet 224x224.



(a) Image Space

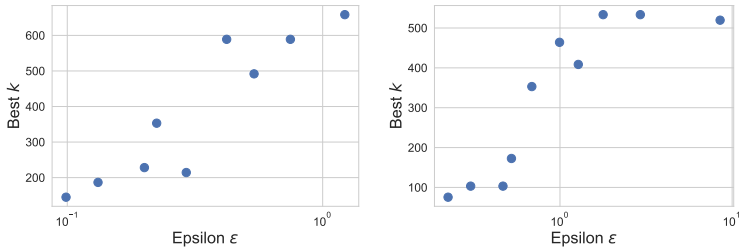
(b) ResNet50 Representations

Figure D.2.1: DP Training of linear classifier on a) Images and b) representations obtained from pre-trained ResNet-50.

D.2.5 Additional Datasets

In this section, we look at results on the MNIST dataset. We consider the standard train-test split of MNIST and train two types of classifiers. The first classifier is the standard linear classifier with cross-entropy loss. The second is a linear classifier with standard cross entropy loss trained on representations of MNIST images obtained from a Resnet-50, pre-trained on ImageNet. Our results, plotted in Figure D.2.1, shows that PILLAR consistently outperforms DP-SGD and the improvement is more prominent for smaller values of ϵ. We also investigated how the best projection dimension k varies as a function of ϵ. The results are shown in Figure D.2.2. As indicated by Theorem 1, the best k increases as ϵ increases.

In addition to results on MNIST dataset, we also conduct experiments using PILLAR on tabular datasets. We select two datasets: Guillermo and Riccardo from the OpenML [Vanschoren et al., 2014] respository. Both of these are binary classification datasets with 4096 dimensions and $16,000$ data points. We train logistic regression models on them using both PILLAR and vanilla DP-SGD. The results presented in Table D.2.3 show that PILLAR consistently outperforms DP-SGD on both of these datasets.



(a) Image Space **(b)** ResNet50 Representations

Figure D.2.2: Best projection dimension k as a function of ϵ on the MNIST dataset.

Privacy	PILLAR	DP-SGD
$\epsilon = 0.1$	75.25	57.8
$\epsilon = 0.3$	76.5	65.4
$\epsilon = 1.0$	78.2	70.2
$\epsilon = 5.0$	79.3	73.6

Riccardo

Privacy	PILLAR	DP-SGD
$\epsilon = 0.1$	60.19	52.3
$\epsilon = 0.3$	61.78	54.6
$\epsilon = 1.0$	63.10	59.2
$\epsilon = 5.0$	64.35	61.62

Guillermo

Table D.2.3: Comparison of PILLAR with DP-SGD on Riccardo and Guillermo datasets from the OpenML repository [Vanschoren et al., 2014].

D.2.6 DP-RAFT Experiments

In this section we present some results that combine DP-RAFT and PILLAR to yield further accuracy improvements. We perform our experiments on CIFAR100, considering learning rate values in $\{0.1, 0.01, 1\}$, training for a number of epochs in $\{5, 10, 50\}$ and for $\epsilon \in \{0.1, 0.7, 1.0\}$. For PILLAR we consider $k \in \{40, 100, 200, 300, 400\}$. In Table D.2.4 we compare the performance of DP-RAFT and the combination of DP-RAFT+PILLAR for ResNet50 and, since the authors of [Panda et al., 2022] consider also additional backbones, we also show the effectiveness of our method on the ConvNeXt-XL backbone. As it can be seen, in all cases using PILLAR in conjunction with DP-RAFT induces a performance improvement.

D.2.7 Experimental details for Section 5.4.2

In this section, we provide details of the other algorithms we compare our approach with in Section 5.4.2. We use the PRV accountant [Gopi et al., 2021b] for all experiments.

JL transformation [Nguyen et al., 2020] [Nguyen et al., 2020] uses JL transformation to reduce the dimensionality of the input. For our baseline, we simulate this method by using Random Matrix

	Public Data	ResNet50 SL	ConvNeXt-XL
Datasets		CIFAR100	CIFAR100
ϵ		0.1 0.7 1.0	0.1 0.7 1.0
DP-RAFT	Unlabelled	28.80 58.35 61.79	68.38 79.12 83.69
DP-RAFT + PILLAR	None	38.75 62.49 64.32	74.69 82.89 85.12

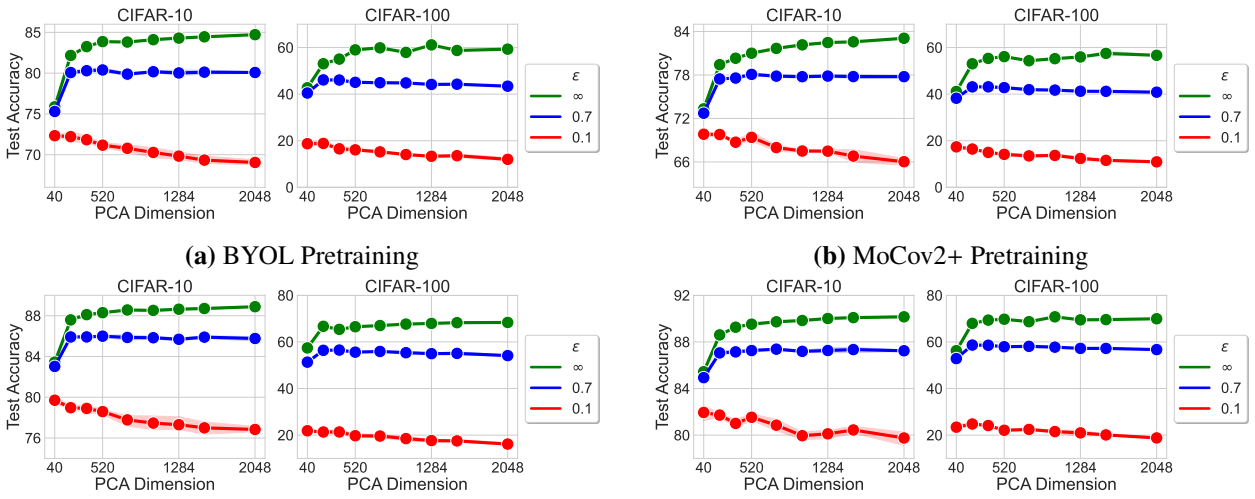
Table D.2.4: Results comparing DP-RAFT and DP-RAFT+PILLAR.

Projection using Gaussian Random Matrices (GRM) instead of PCA to reduce the dimensionality of the inputs. Our experimental results in Table 5.4.1 show that our method outperforms these approaches. Although this approach does not require the availability of public data, this comparison allows us to conclude that reducing the dimensionality of the input is not sufficient to achieve improved performance. Furthermore, even though the JL Lemma [Johnson, 1984] guarantees distances between inputs are preserved up to a certain distortion in the lower-dimensionality space, the dataset size required to guarantee a small distortion is much larger than what is available in practice. We leverage `scikit-learn` to project the data to a target dimension identical to the ones we use for PCA. We similarly search the same hyperparameter space.

GEP [Yu et al., 2021a] We use the code-base[4] released by the authors for implementation of GEP. We conduct hyper-parameter search for the learning rate in $\{0.01, 0.05, 0.1, 1\}$ and the number of steps in $\{500, 1000, 2500, 3000, 5000, 6000, 20000\}$. As recommended by the authors, we set the highest clipping rate to $\{1, 0.1, 0.01\}$ and the lowest clipping rate is obtained by multiplying the highest with 0.20. The anchor dimension ranges in $\{40, 120, 200, 280, 512, 1024, 1580\}$. We try batch sizes in $\{64, 512, 1024\}$. We tried using $\{0.1\%, 0.01\%\}$ of the data as public. Despite this extensive hyperparameter search, we could not manage to make GEP achieve better performance than the DP-SGD baseline (see Table 5.4.1).

AdaDPS [Li et al., 2022a] We use the code-base[5] released by the authors of AdaDPS. We estimate the noise variance as a function of the number of steps and the target ϵ using the code of `opacus` under the RDP accountant (whose implementation is the same as the code released by the authors of AdaDPS). We search the learning rate in $\{0.01, 0.1, 1\}$, the number of steps in $\{500, 1000, 2500, 3000, 5000, 6000, 7500, 10000\}$, the batch size in $\{32, 64, 128, 512, 1024\}$, and

[4] `https://github.com/dayu11/Gradient-Embedding-Perturbation`
[5] `https://github.com/litian96/AdaDPS`



(a) BYOL Pretraining

(b) MoCov2+ Pretraining

(c) SemiSL Pretraining

(d) SemiWeakSL Pretraining

Figure D.2.3: DP Training of linear classifier on different pre-trained features using the PRV accountant for CIFAR-10 and CIFAR-100.

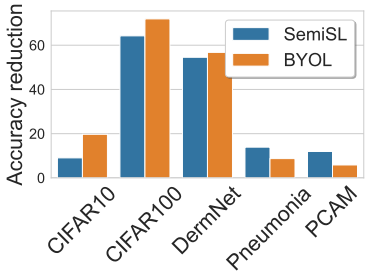


Figure D.2.4: Comparing reduction in test accuracy for different datasets between using SemiSL and BYOL pre-trained networks.

we tried using $\{0.1\%, 0.01\%\}$ of the data as public, and the ϵ_c (the conditioner hyperparameter) in $\{10, 1, 0.1, 1e-3, 1e-5, 1e-7\}$. of the validation data for the public data conditioning. We applied micro-batching in $\{2, 4, 32\}$. Despite this extensive hyperparameter search, we could not manage to make AdaDPS achieve better performance than the DP-SGD baseline.

DP-PCA [Abadi et al., 2016] All settings are the same with respect to PILLAR, except for the additional need of cross-validating the privacy budget consumed by the DP-PCA procedure. We consider $1\%, 25\%, 50\%$. For DP-PCA, we use the `diffprivlib` implementation.

D.2.8 Different pre-training algorithms

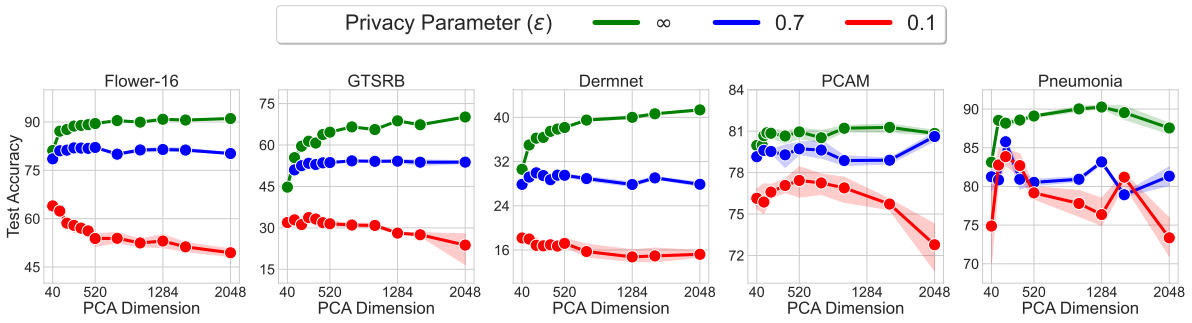
In Figure 5.4.1 and 5.4.2 in the main text, we only reported accuracies for the best performing pre-training algorithm. In this, section we report the performance of our algorithm against the remaining pre-training algorithms that we consider in this paper. In particular, we consider two self-supervised pre-training algorithms: BYOL [Grill et al., 2020] and MoCov2+ [Chen et al., 2020b] and two semi-supervised algorithms [Yalniz et al., 2019]. While one of them is a Semi-Supervised (SemiSL) algorithm, the other only uses weak supervision and we refer to it Semi-Weakly Supervised (SemiWeakSL) algorithm. In Figure D.2.3 we report the results on CIFAR-10 and CIFAR-100. In Figure D.2.5 we report the results for Flower-16 [Flo, 2021], GTSRB [Houben et al., 2013], PCAM [Veeling et al., 2018], Pneumonia [Kermany et al., 2018] and DermNet [Der, 2019].

Similar to Figure 5.5.1 in the main text, we show the accuracy reduction for Semi-Supervised pre-training vs BYOL (Self-Supervised) pre-trianing in Figure D.2.4. Our results shows similar results as [Shi et al., 2023] that labels are more useful for pre-training for tasks where there is a significant label overlap between the pre-training and the final task.

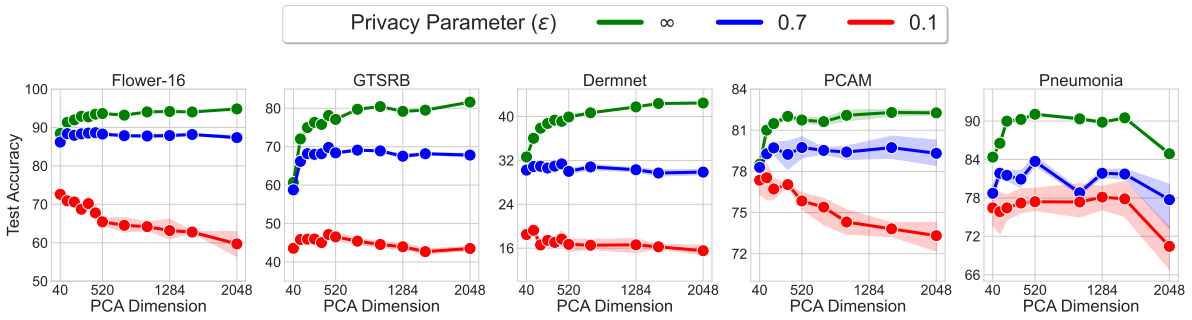
D.3 Computational Cost, Broader Impact and Limitations

Computational cost Except for the supervised training on ImageNet32x32, we leverage pre-trained models. To optimize the training procedure, we checkpoint feature embeddings for each dataset and pre-trained model. Therefore, training requires loading the checkpoint and training a linear layer via SGD (or DP-SGD), accelerating the training procedure by avoiding the forward pass through the feature encoder. We use a single Tesla M40 (11GB) for each run.

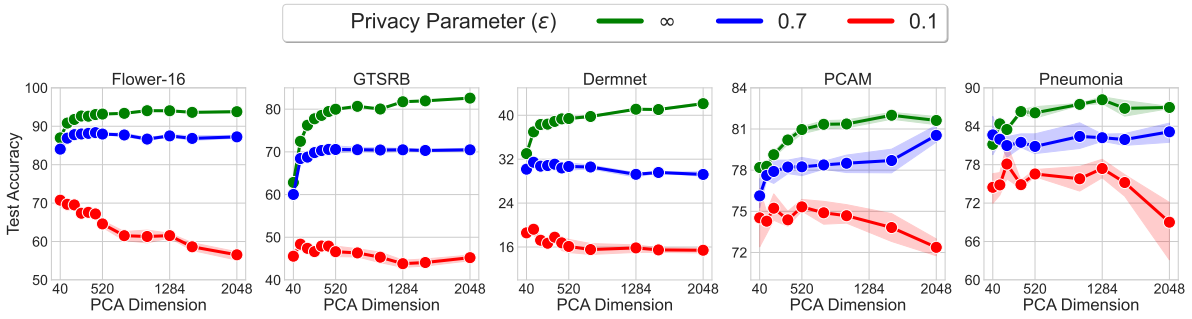
Broader impact and Limitations In this work we show our method can be used in order to increase the utility of models under tight Differential Privacy constraints. Increasing the utility for low ϵ is crucial to foster the adoption of DP methods that provide provable guarantees for the privacy of users. Further, unlike several recent works that have shown improvement in accuracy for deep neural networks, our algorithm can be run on commonly available computational resources like a Tesla M40 11GB GPU as it does not require large batch sizes. We hope this will make DP training of high-performing classifiers more accessible. Finally, we show our algorithm improves not only on commonly used benchmarks like CIFAR10 and CIFAR100 but also in privacy relevant tasks like medical datasets including Pneumonia, PCAM, and DermNet. We hope this will encourage future works to also consider benchmarking their algorithms on more privacy relevant tasks.



(a) MoCov2+ Pretraining



(b) SemiSL Pretraining



(c) SemiWeakSL Pretraining

Figure D.2.5: DP Training of linear classifier on different pre-trained features using the PRV accountant for Flower-16, GTSRB, DermNet, PCAM, and Pneumonia.

As discussed, the assumption that labelled public data is available may not hold true in several applications. Our algorithm does not require the public data to be labelled, however the distribution shift between the public unlabelled data and the private one should not be too large. We have shown that for relatively small distribution shift our method remains effective. Finally, recent works have suggested that differentially private learning may disparately impact certain subgroups more than others [Bagdasaryan et al., 2019, Cummings et al., 2019, Sanyal et al., 2022]. It remains to explore whether semi-private learning can help alleviate these disparity.

E

Appendix of Chapter 6

Contents

E.1 Impact statement

This paper shows it is possible for a malicious user to prompt a model to reveal training data. This phenomenon is studied in a worst-case but plausible condition in which the attacker knows the training image and question, except for the answer. Our study only represents a starting step in the direction of prompting VLMs to elicit the extraction of private data. It may be possible for an attacker to develop more sophisticated attack strategies. Such strategies can be used both in a beneficial way (e.g., for organizations to audit the privacy preserving properties of their systems) or maliciously (e.g., for an attacker to obtain confidential information).

In this study we have used public data, and for further caution we have anonymised all the sensitive samples we reported in our qualitative analysis. Indeed, in some parts of the world the Right To Be Forgotten is in place, and the individuals whose data is reported in the considered public dataset my ask for their data to be cancelled. When performing our quantitative analysis, we report aggregate numbers and described the extractable samples without revealing their exact content for the same reasons. Therefore, we expect no individual or organization to be harmed by reporting our results.

Furthermore, although we propose a countermeasure (EB) that is effective across all the attack scenarios we considered, it is still a heuristic approach and may not prevent extraction in case more sophisticated attack techniques are developed. Furthermore, it may hypothetically introduce a "side-channel" that an adversary might exploit to increase the exposure to membership inference attacks: if the model responds with the default negation, this may be seen as an index the sample was in the training set. This may not be relevant for several applications, where the information to be protected is not the membership of a document to the training set but the specific content of the document, but may be problematic in other applications. An obvious solution would be to apply Differentially Private (DP) training. However, scaling DP to VLMs without causing significant utility degradation is beyond the scope of this work.

E.2 Computational Cost of Training

Donut Fine-tuning Donut at maximum input resolution requires 64 A100 GPUs for a day. Given its relatively compact size (176M parameters), Donut can be trained on high-resolution input images ($2560 \times 1920 \approx 5$M pixels), a crucial aspect for achieving optimal performance. Lowering the resolution can significantly reduce the cost of training, however, as we observe, it increases the tendency of the model to memorize the training data and reduces the generalization capabilities of the models. Therefore it is not recommended.

$\|M\|$/ #PII	No Text	Paraphrasing	Shuffling	R5°	R10°	T20px	T200px	B×2	B×1.3	B×0.8	B×0.5
Donut	0 / 0	0 / 0	0 / 0	1 / 0	0 / 0	0 / 0	0 / 0	6 / 1	6 / 1	2 / 0	5 / 0
Pix2Struct-B	0 / 0	1 / 0	1 / 0	2 / 0	2 / 0	0 / 0	0 / 0	0 / 0	4 / 0	4 / 0	0 / 0
Pix2Struct-L	1 / 0	0 / 0	0 / 0	1 / 0	1 / 0	0 / 0	0 / 0	0 / 0	4 / 0	2 / 0	0 / 0
PaLI	0 / 0	0 / 0	2 / 0	1 / 0	1/ 0	0 / 0	2 / 0	3 / 0	1 / 0	0 / 0	1 / 0

Table E.3.1: Effectiveness of extraction blocking for the various contexts portrayed in Figure 6.4.2. Notice, we do not include in the training sets any of the contexts we consider in this table. This indicates the protection offered by Extraction Blocking extends beyond the types of context provided at training time.

Pix2Struct Fine-tuning Pix2Struct Base, independently of the resolution, requires 32 TPUv2 for about 5 hours. Training Pix2Struct Large, independently of the resolution, requires 64 TPUv2 for about 5 hours. Due to its relatively larger size, the smaller model is fine-tuned at a resolution of about $1.2M$ pixels, while the larger model is fine-tuned at a resolution of about $0.8M$ pixels.

PaLI-3 Fine-tuning PaLI-3 64 TPUv2 for 15 hours. Due to its size (5B parameters), it is typically fine-tuned at a resolution of approximately $1.1M$ pixels (1064×1064).

Computing the memorization scores The amount of compute needs to be multiplied by the number of runs for each measurement: for the simplest attribution method we consider, we only need 2 runs; for the counterfactual extractable memorization and simplicity scores, we need to perform 50 runs. Perform more is both computationally prohibitive and expensive for the storage of the largest models we consider.

E.3 Further results

Effectivenes of EB for prompting strategies not used in the training set In Section 6.5 we have considered several ways to prompt the model. Since EB includes only samples using a worst-case prompting strategy (I^{-a}, Q), it may be natural to wonder whether EB is still effective if an adversary prompts the model in different ways. We observe the technique is actually still extremely effective, see Table E.3.1

E.4 PII categories and their frequencies in the canaries

We manually annotate each answer in the canaries set as either PII or non-PII. We also classify each PII element as one of the following classes: Places, Person, Temporal, Contact (Phone/Fax/Email), NRP (Nationality Religion Politic), URL, and other forms of IDs (e.g. card numbers, serial numbers

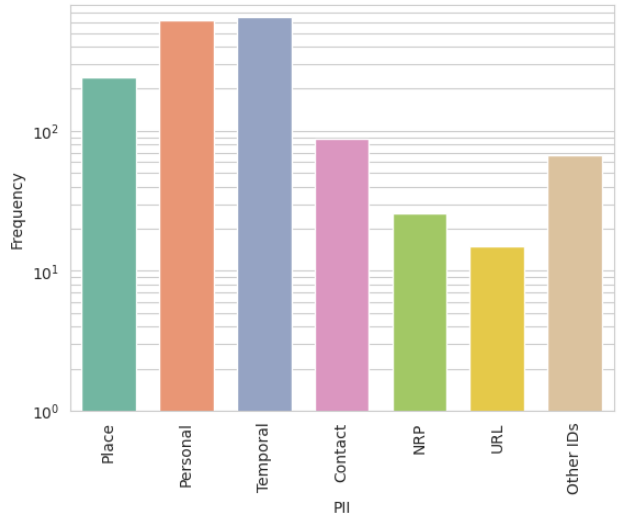


Figure E.4.1: Frequency of different types of Personally Identifying Information (PII) in the canaries set $\mathcal{D}^C$.

of tickets, document or people numerical identifiers etc.). The distribution of PII in the canaries set $\mathcal{D}^C$ is reported in Figure E.4.1.

E.5 Further Related Works

E.5.1 Document-Based Visual Question Answering

Given the greater simplicity of solving the VQA problem by separating the tasks of document reading and document understanding, OCR-reliant systems have been the state of the art for a few years [Tito et al., 2022, Huang et al., 2022]. However, as argued by [Kim et al., 2022], OCR-reliant systems have the disadvantage of requiring an expensive OCR-preprocessing step, making the inference cost higher in case high-quality OCR results are required, with errors of the OCR system propagating to the VQA component. The phenomenon is particularly apparent for languages with complex character sets, requiring an expensive post-OCR correction module [Rijhwani et al., 2020, Schaefer and Neudecker, 2020]. For these reasons, OCR-free systems like [Kim et al., 2022, Lee et al., 2023] have received increasing attention, with state-of-the-art models like PALI-3 [Chen et al., 2023b] closing the performance gap between the OCR-reliant and OCR-free models. In this work we mainly focus on three state-of-the-art OCR-free systems that differ in model size, architecture and pre-training stages. We consider both **Donut** [Kim et al., 2022] and **Pix2Struct** [Lee et al., 2023] among the set of models that are specialised to perform document understanding. We also consider **PALI-3** [Chen et al., 2023b], a foundational vision-language model that can be fine-tuned

in order to solve the task of document understanding, achieving state-of-the-art performance.

E.5.2 Relations to Distributional Shortcut Learning in VQA

It is known that VQA systems can produce correct responses due to their ability to learn and leverage the frequent association of a specific answer to some question (linguistic shortcut) [Jabri et al., 2016, Niu et al., 2021, Goyal et al., 2017, Chen et al., 2020a]. For instance, if the question is *"What is the colour of the grass?"*, if the grass is green in most of the training images for which the question is asked, the model will respond green independently of the actual colour in the considered test image. This type of shortcuts does not need to be exclusively linguistic, and may involve the frequent co-occurrence of elements in the input image (visual shortcut) or their combination with specific words in the question (multimodal shortcut) [Dancette et al., 2021, Si et al., 2022]. In other terms, VQA systems can learn simple rules relying on spurious but predictive features that co-occurr across multiple samples in order to respond accurately even when the input image lacks the considered information or contradicts it.

The concurrent work of [Tito et al., 2023] has shown this phenomenon occurring also in document-based Visual Question Answering. The authors propose a new federated learning dataset containing invoices from several data providers. Since a provider's information (specifically, their name and email address) is *repeated across several invoices* that share visual and linguistic similarities (e.g., identical layout, formatting, logos, fields etc.), a model can infer a provider's name or email address correctly on *previously unseen test* documents from the known provider that do not contain the requested information.

Bibliography

Gpt4-v(ision) system card. https://cdn.openai.com/papers/GPTV_System_Card.pdf. Accessed: 2024-02-18. Cited on page 95.

Dataset for 23 skin lesions. https://www.kaggle.com/datasets/shubhamgoel27/dermnet, 2019. Cited on pages 86, 190, and 196.

Flowers dataset. https://tinyurl.com/2p8vpsp2, 2021. Cited on pages 86 and 196.

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM SIGSAC Conference on Computer and Communications Security*, 2016. Cited on pages vii, 7, 71, 72, 76, 80, 83, 84, and 195.

Giacomo Acciarini, Francesco Pinto, Sascha Metz, Sarah Boufelja, Sylvester Kaczmarek, Klaus Merz, José A Martinez-Heras, Francesca Letizia, Christopher Bridges, and Atılım Güneş Baydin. Spacecraft collision risk assessment with probabilistic programming. *ML 4 Physical Sciences Workshop NeurIPS*, 2020. Cited on page 4.

Giacomo Acciarini, Francesco Pinto, Francesca Letizia, José A Martinez-Heras, Klaus Merz, Christopher Bridges, and Atılım Güneş Baydin. Kessler: A machine learning library for spacecraft collision avoidance. In *8th European Conference on Space Debris*, pages 1–9, 2021. Cited on page 4.

Noga Alon, Raef Bassily, and Shay Moran. Limits of private learning with access to public data. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. Cited on pages 70, 73, and 79.

Ehsan Amid, Arun Ganesh, Rajiv Mathews, Swaroop Ramaswamy, Shuang Song, Thomas Steinke, Vinith M Suriyakumar, Om Thakkar, and Abhradeep Thakurta. Public data-assisted mirror descent for private model training. In *International Conference on Machine Learning (ICML)*, 2022. Cited on page 70.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023. Cited on page 105.

Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999. Cited on page 181.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant Risk Minimization. *arXiv e-prints*, art. arXiv:1907.02893, July 2019. Cited on page 37.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. Cited on page 53.

Hilal Asi, John C. Duchi, Alireza Fallah, Omid Javidbakht, and Kunal Talwar. Private adaptive gradient methods for convex optimization. *CoRR*, abs/2106.13756, 2021. Cited on pages 72 and 76.

Pranjal Awasthi, Natalie Frank, and Mehryar Mohri. On the Rademacher complexity of linear hypothesis sets. *arXiv:2007.11045*, 2020. Cited on page 81.

Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J. Fleet. Synthetic data from diffusion models improves imagenet classification. *ArXiv*, abs/2304.08466, 2023. Cited on pages 52 and 55.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. Cited on page 34.

Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. Cited on page 197.

Yutong Bai, Jieru Mei, Alan Yuille, and Cihang Xie. Are transformers more robust than cnns? 2021. Cited on pages 33 and 36.

Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE Transactions on Medical Imaging*, 2018. Cited on pages x, 161, and 162.

Hritik Bansal and Aditya Grover. Leaving reality to imagination: Robust classification via generated datasets. In *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*, 2023. Cited on pages 52 and 55.

Peter Bartlett, Michael Jordan, and Jon Mcauliffe. Large margin classifiers: Convex loss, low noise, and convergence rates. In *Advances in Neural Information Processing Systems*, 2003. Cited on page 176.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Annual Symposium on Foundations of Computer Science (FOCS)*, 2014a. Cited on pages 70, 72, 75, and 170.

Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. Private empirical risk minimization, revisited. *ICML Workshop on Learning, Security and Privacy*, 2014b. Cited on pages 79 and 170.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Characterizing the sample complexity of private learners. In *Innovations in Theoretical Computer Science (ITCS)*, 2013a. Cited on page 70.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 2013b. Cited on page 70.

Abhijit Bendale and Terrance Boult. Towards open set deep networks. November 2015. Cited on page 5.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009. Cited on page 156.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738. Cited on page 121.

Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the SuLQ framework. In *ACM symposium on Principles of database systems*, 2005. Cited on page 70.

Franziska Boenisch, Christopher Mühl, Adam Dziedzic, Roy Rinberg, and Nicolas Papernot. Have it your way: Individualized privacy assignment for DP-SGD. *arXiv:2303.17046*, 2023. Cited on page 191.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022. Cited on page 62.

John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. Cited on page 62.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284, 2019. Cited on pages 7 and 107.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021. Cited on pages 7, 96, and 97.

Nicholas Carlini, Steve Chien, Milad Nasar, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *IEEE Symposium on Security and Privacy (SP)*, 2022a. Cited on pages 70 and 72.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022b. Cited on pages 6, 103, and 107.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2023a. URL `https://openreview.net/forum?id=TatRHT_1cK`. Cited on pages 95, 96, 97, 99, 100, and 105.

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5253–5270, 2023b. Cited on pages 96 and 97.

Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS'00, page 395–401, Cambridge, MA, USA, 2000. MIT Press. Cited on page 17.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research (JMLR)*, 2011. Cited on pages 70 and 72.

Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10800–10809, 2020a. Cited on pages 96, 108, 109, and 203.

Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023a. Cited on page 95.

Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, Daniel Salz, Xi Xiong, Daniel Vlasic, Filip Pavetic, Keran Rong, Tianli Yu, Daniel Keysers, Xiaohua Zhai, and Radu Soricut. Pali-3 vision language models: Smaller, faster, stronger, 2023b. Cited on pages 95, 98, 99, and 202.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv:2003.04297*, 2020b. Cited on pages 75 and 196.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019. Cited on page 18.

Filipe Condessa, Jelena Kovacevic, and Jose Bioucas-Dias. Performance measures for classification systems with rejection. April 2015. Cited on pages 4 and 44.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. Cited on pages 54 and 57.

Ilke Cugu, Massimiliano Mancini, Yanbei Chen, and Zeynep Akata. Attention consistency on visual corruptions for single-source domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4165–4174, June 2022. Cited on pages 54 and 57.

Rachel Cummings, Varun Gupta, Dhamma Kimpara, and Jamie Morgenstern. On the compatibility of privacy and fairness. In *Conference on User Modeling, Adaptation and Personalization*, 2019. Cited on page 197.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Math. Control Signal Systems 2, 303–314*, 1989. Cited on page 36.

Victor Guilherme Turrisi da Costa, Enrico Fini, Moin Nabi, Nicu Sebe, and Elisa Ricci. solo-learn: A library of self-supervised methods for visual representation learning. *Journal of Machine Learning Research (JMLR)*, 2022. Cited on page 189.

Corentin Dancette, Remi Cadene, Damien Teney, and Matthieu Cord. Beyond question-based biases: Assessing multimodal shortcut learning in visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1574–1583, 2021. Cited on pages 95 and 203.

Felix Dangel, Frederik Kunstner, and Philipp Hennig. Backpack: Packing more into backprop. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=BJlrF24twB`. Cited on pages 115 and 120.

Etienne David, Simon Madec, Pouria Sadeghi-Tehran, Helge Aasen, Bangyou Zheng, Shouyang Liu, Norbert Kirchgessner, Goro Ishikawa, Koichi Nagasawa, Minhajul A Badhon, Curtis Pozniak, Benoit de Solan, Andreas Hund, Scott C. Chapman, Frederic Baret, Ian Stavness, and Wei Guo. Global wheat head detection (gwhd) dataset: a large and diverse dataset of high-resolution rgb-labelled images to develop and benchmark wheat head detection methods. *Plant Phenomics*, 2020, 2020. Cited on pages x, 161, and 163.

Etienne David, Mario Serouart, Daniel Smith, Simon Madec, Kaaviya Velumani, Shouyang Liu, Xu Wang, Francisco Pinto Espinosa, Shahameh Shafiee, Izzat S. A. Tahir, Hisashi Tsujimoto, Shuhei Nasuda, Bangyou Zheng, Norbert Kichgessner, Helge Aasen, Andreas Hund, Pouria Sadhegi-Tehran, Koichi Nagasawa, Goro Ishikawa, Sebastien Dandrifosse, Alexis Carlier, Benoit Mercatoris, Ken Kuroki, Haozhou Wang, Masanori Ishii, Minhajul A. Badhon, Curtis Pozniak, David Shaner LeBauer, Morten Lilimo, Jesse Poland, Scott Chapman, Benoit de Solan, Frederic Baret, Ian Stavness, and Wei Guo. Global wheat head dataset 2021: an update to improve the benchmarking wheat head localization with more diversity, 2021. Cited on pages x, 161, and 163.

Brian Davis, Bryan Morse, Brian Price, Chris Tensmeyer, Curtis Wigington, and Vlad Morariu. End-to-end document recognition and understanding with dessurt. In *European Conference on Computer Vision*, pages 280–296. Springer, 2022. Cited on page 95.

Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv:2204.13650*, 2022. Cited on pages xiv, 70, 75, 82, 84, 86, 189, 190, and 191.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009a. Cited on pages 22, 33, and 35.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009b. Cited on page 55.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. Cited on page 38.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. Cited on pages 54 and 57.

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. October 2020. Cited on pages 33 and 35.

Cynthia Dwork. A firm foundation for private data analysis. *Communications of the ACM*, 2011. Cited on page 82.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. *Theory of Cryptography*, 2006. Cited on pages 6, 70, 72, and 172.

Vitaly Feldman. Does learning require memorization? A short tale about a long tail. *CoRR*, abs/1906.05271, 2019. URL http://arxiv.org/abs/1906.05271. Cited on pages 95, 97, 102, 103, and 107.

Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. In *Conference on Learning Theory (COLT)*, 2014. Cited on page 70.

Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020. Cited on page 97.

Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of Out-of-Distribution detection. June 2021. Cited on page 33.

Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: an end-to-end case study of personalized warfarin dosing. In *Proceedings of the 23rd USENIX Conference on Security Symposium*, SEC'14, page 17–32, USA, 2014. USENIX Association. ISBN 9781931971157. Cited on page 7.

Giorgio Fumera and Fabio Roli. Support vector machines with embedded reject option. In *Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines*, SVM '02, page 68–82, Berlin, Heidelberg, 2002. Springer-Verlag. ISBN 354044016X. Cited on page 44.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022. URL https://arxiv.org/abs/2208.01618. Cited on pages 153 and 154.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1050–1059. JMLR.org, 2016a. Cited on page 120.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016b. Cited on page 5.

Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 619–633, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356930. doi: 10.1145/3243734.3243834. URL https://doi.org/10.1145/3243734.3243834. Cited on page 7.

Jin Gao, Jialing Zhang, Xihui Liu, Trevor Darrell, Evan Shelhamer, and Dequan Wang. Back to the source: Diffusion-driven test-time adaptation. *arXiv preprint arXiv:2207.03442*, 2022. Cited on page 55.

R Geirhos, P Rubisch, C Michaelis, M Bethge, and others. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv*, 2018a. Cited on pages 35 and 39.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018b. Cited on page 56.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019. Cited on pages 54, 56, and 145.

Shayan Oveis Gharan. Low rank approximation. Lecture notes for CSE 521: Design and Analysis of Algorithms I, 2017. Cited on page 188.

Muhammad Ghifary, David Balduzzi, W Bastiaan Kleijn, and Mengjie Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1414–1430, 2016. Cited on page 154.

Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Conference on Neural Information Processing Systems (NeurIPS)*, 2021a. Cited on pages 82 and 189.

Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021b. Cited on page 193.

Sindhu C.M. Gowda, Shalmali Joshi, Haoran Zhang, and Marzyeh Ghassemi. Pulling up by the causal bootstraps: Causal data augmentation for pre-training debiasing. *arXiv preprint arXiv:2108.12510*, 2021. Cited on pages 52, 53, and 54.

Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6325–6334, Los Alamitos, CA, USA, jul 2017. IEEE Computer Society. doi: 10.1109/CVPR.2017.670. URL `https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.670`. Cited on pages 95 and 203.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. Cited on pages vii, 71, 75, and 196.

Xin Gu, Gautam Kamath, and Zhiwei Steven Wu. Choosing public datasets for private machine learning via gradient subspace distance. *arXiv:2303.01256*, 2023. Cited on page 75.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020. Cited on pages 51 and 53.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1321–1330. JMLR.org, 2017. Cited on pages 19, 29, 42, and 119.

Chuan Guo, Florian Bordes, Pascal Vincent, and Kamalika Chaudhuri. Do ssl models have d\'ej\a vu? a case of unintended memorization in self-supervised learning. *arXiv preprint arXiv:2304.13850*, 2023. Cited on pages 97 and 100.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90. Cited on page 22.

Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? In *The Eleventh International Conference on Learning Representations*, 2023. Cited on pages 52, 55, 61, 64, 65, and 151.

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *CVPR Workshops*, pages 58–74, 2019. URL http://openaccess.thecvf.com/content_CVPRW_2019/html/Uncertainty_and_Robustness_in_Deep_Visual_Learning/Hein_Why_ReLU_networks_yield_high-confidence_predictions_far_away_from_the_CVPRW_2019_paper.html. Cited on page 5.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019. Cited on page 22.

Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415, 2016a. URL http://arxiv.org/abs/1606.08415. Cited on page 34.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and Out-of-Distribution examples in neural networks. October 2016b. Cited on page 40.

Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. December 2019a. Cited on pages 15 and 23.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. July 2019b. Cited on pages 23, 35, 40, and 44.

Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. Cited on pages 54 and 57.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021. Cited on pages 23 and 35.

Dan Hendrycks, Andy Zou, Mantas Mazeika, Leonard Tang, Bo Li, Dawn Song, and Jacob Steinhardt. Pixmix: Dreamlike pictures comprehensively improve safety measures. *CVPR*, 2022. Cited on pages 54 and 57.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two Time-Scale update rule converge to a local nash equilibrium. June 2017. Cited on page 164.

Marius Hobbhahn, Agustinus Kristiadi, and Philipp Hennig. Fast predictive uncertainty for classification with bayesian deep networks, 2021. URL https://openreview.net/forum?id=KcImcc3j-qS. Cited on pages 5, 115, and 121.

Minui Hong, Jinwoo Choi, and Gunhee Kim. Stylemix: Separating content and style for enhanced data augmentation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14857–14865, 2021. doi: 10.1109/CVPR46437.2021.01462. Cited on pages 52 and 54.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(89)90020-8. URL https://www.sciencedirect.com/science/article/pii/0893608089900208. Cited on page 36.

Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, 2013. Cited on pages 86, 190, and 196.

Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized ODIN: Detecting out-of-distribution image without learning from out-of-distribution data. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2020. Cited on page 5.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. Snapshot ensembles: Train 1, get m for free, 2017. Cited on page 5.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, 2022. Cited on page 202.

Zeyi Huang, Haohan Wang, Eric P. Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *ECCV*, 2020. Cited on page 57.

Maximilian Ilse, Jakub M Tomczak, and Patrick Forré. Selecting data augmentation for simulating interventions. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4555–4562. PMLR, 18–24 Jul 2021. Cited on pages 51, 52, 53, and 54.

Allan Jabri, Armand Joulin, and Laurens van der Maaten. Revisiting visual question answering baselines. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 727–739, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46484-8. Cited on pages 95 and 203.

Philip T Jackson, Amir Atapour-Abarghouei, Stephen Bonner, Toby P Breckon, and Boguslaw Obara. Style augmentation: Data augmentation via style randomization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 83–92, 2019. Cited on page 54.

Carlos Jensen, Colin Potts, and Christian Jensen. Privacy practices of internet users: Self-reports versus observed behavior. *International Journal of Human-Computer Studies*, 2005. Cited on page 74.

Yiding Jiang*, Behnam Neyshabur*, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJgIPJBFvH. Cited on page 130.

William B. Johnson. Extensions of lipschitz mappings into hilbert space. *Contemporary Mathematics*, 1984. Cited on page 194.

Thomas Joy, Francesco Pinto, Ser Nam Lim, Philip H. S. Torr, and Puneet Kumar Dokania. Sample-dependent adaptive temperature scaling for improved calibration. In *AAAI Conference on Artificial Intelligence*, 2022. URL https://api.semanticscholar.org/CorpusID:250492910. Cited on page 5.

Peter Kairouz, Mónica Ribero, Keith Rush, and Abhradeep Thakurta. Fast dimension independent private adagrad on publicly estimated subspaces. *arXiv:2008.06570*, 2020. Cited on pages 70 and 80.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR, 2022. Cited on page 96.

Shiva Prasad Kasiviswanathan. SGD with low-dimensional gradients with applications to private and distributed learning. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2021. Cited on pages 70 and 80.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 2011. Cited on page 70.

Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *cell*, 2018. Cited on pages 86, 190, and 196.

Daniel Khashabi, Xinxi Lyu, Sewon Min, Lianhui Qin, Kyle Richardson, Sean Welleck, Hannaneh Hajishirzi, Tushar Khot, Ashish Sabharwal, Sameer Singh, and Yejin Choi. Prompt waywardness: The curious case of discretized interpretation of continuous prompts. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3631–3643, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.266. URL https://aclanthology.org/2022.naacl-main.266. Cited on page 154.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *European Conference on Computer Vision (ECCV)*, 2022. Cited on pages viii, 95, 98, 99, and 202.

Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28, pages 2575–2583. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/bc7316929fe1545bf0b98d114ee3ecb8-Paper.pdf. Cited on page 120.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (BiT): General visual representation learning. 2020. Cited on pages 33 and 35.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in ReLU networks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International*

Conference on Machine Learning, volume 119 of *Proceedings of Machine Learning Research*, pages 5436–5446. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/kristiadi20a.html. Cited on pages 5, 23, 117, 120, and 121.

Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009. URL http://www.cs.toronto.edu/~kriz/cifar.html. Cited on pages 81 and 190.

Alexey Kurakin, Steve Chien, Shuang Song, Roxana Geambasu, Andreas Terzis, and Abhradeep Thakurta. Toward training at ImageNet scale with differential privacy. *arXiv:2201.12328*, 2022. Cited on pages 70 and 82.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*, 2016. Cited on page 5.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 6402–6413. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/9ef2ed4b7fd2c810847ffa5fa85bce38-Paper.pdf. Cited on pages 15, 16, and 24.

Thomas C. W. Landgrebe, David M. J. Tax, Pavel Paclík, and Robert P. W. Duin. The interaction between classification and reject performance for distance-based reject-option classifiers. *Pattern Recogn. Lett.*, 27 (8):908–917, jun 2006. ISSN 0167-8655. doi: 10.1016/j.patrec.2005.10.015. URL https://doi.org/10.1016/j.patrec.2005.10.015. Cited on page 44.

Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. 2015. Cited on page 23.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *IEEE Symposium on Security and Privacy (SP)*, 2019. Cited on page 18.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2Struct: Screenshot parsing as pretraining for visual language understanding. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 18893–18912. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/lee23g.html. Cited on pages 95, 98, 99, and 202.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Deeper, broader and artier domain generalization. In *The International Conference on Computer Vision (ICCV 2017)*, pages 5543–5551, 2017. ISBN 978-1-5386-1033-6. Cited on page 56.

Tian Li, Manzil Zaheer, Sashank Reddi, and Virginia Smith. Private adaptive optimization with side information. In *International Conference on Machine Learning (ICML)*, 2022a. Cited on pages vii, 70, 71, 84, and 194.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353. Cited on page 154.

Xuechen Li, Daogao Liu, Tatsunori Hashimoto, Huseyin A Inan, Janardhan Kulkarni, YinTat Lee, and Abhradeep Guha Thakurta. When does differentially private learning not suffer in high dimensions? In *Conference on Neural Information Processing Systems (NeurIPS)*, 2022b. Cited on pages vii, 71, 79, 83, 186, 187, and 189.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *International Conference on Learning Representations (ICLR)*, 2022c. Cited on page 70.

Yingwei Li, Qihang Yu, Mingxing Tan, Jieru Mei, Peng Tang, Wei Shen, Alan Yuille, and Cihang Xie. Shape-texture debiased neural network training. *arXiv preprint arXiv:2010.05981*, 2020. Cited on pages 52 and 54.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online, November 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.findings-emnlp.165. Cited on pages 151 and 152.

Jeremiah Zhe Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In *NeurIPS*, June 2020a. Cited on pages 15, 16, 23, 27, 116, and 120.

Jeremiah Zhe Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *NeurIPS*, 2020b. Cited on page 5.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021a. Cited on page 154.

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection, 2020c. Cited on pages 15 and 118.

Xiyang Liu, Weihao Kong, Prateek Jain, and Sewoong Oh. Dp-pca: Statistically optimal and differentially private pca. In *Advances in Neural Information Processing Systems*, 2022a. Cited on page 80.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. March 2021b. Cited on pages 33 and 35.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. 2022b. Cited on pages 32, 33, and 35.

Shangyun Lu, 1 Bradley Nott, and 1 Aaron Olson. Harder or different? a closer look at distribution shift in dataset reproduction. http://www.gatsby.ucl.ac.uk/~balaji/udl2020/accepted-papers/UDL2020-paper-101.pdf. Accessed: 2021-11-10. Cited on page 23.

Zhiyun Lu, Eugene Ie, and Fei Sha. Uncertainty estimation with infinitesimal jackknife, its distribution and mean-field approximation, 2020. Cited on pages 115 and 121.

Michal Lukasik, Vaishnavh Nagarajan, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. What do larger image classifiers memorise?, 2023. Cited on pages 95, 97, 103, and 107.

Fangrui Lv, Jian Liang, Shuang Li, Bin Zang, Chi Harold Liu, Ziteng Wang, and Di Liu. Causality inspired representation learning for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8046–8056, 2022. Cited on page 54.

Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble distribution distillation. April 2019. Cited on pages 44 and 45.

Minesh Mathew, Dimosthenis Karatzas, and C.V. Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2200–2209, January 2021. Cited on pages viii, 95, 98, and 108.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. Infographicvqa. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2582–2591, 2022. doi: 10.1109/WACV51458.2022.00264. Cited on page 98.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Annual Symposium on Foundations of Computer Science (FOCS)*, 2007. Cited on page 79.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. Cited on page 55.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5316–5330, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.365. URL https://aclanthology.org/2022.acl-long.365. Cited on page 154.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. June 2021. Cited on pages 5, 33, and 43.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, February 2018. Cited on page 23.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012. Cited on page 174.

Milad Moradi and Matthias Samwald. Evaluating the robustness of neural language models to input perturbations. *arXiv preprint arXiv:2108.12237*, 2021. Cited on page 57.

Eduardo Mosqueira-Rey, Elena Hernández-Pereira, David Alonso-Ríos, José Bobes-Bascarán, and Ángel Fernández-Leal. Human-in-the-loop machine learning: A state of the art. *Artificial Intelligence Review*, pages 1–50, 2022. Cited on page 156.

Christopher Mühl and Franziska Boenisch. Personalized PATE: Differential privacy for machine learning with individual privacy guarantees. In *Proceedings on Privacy Enhancing Technologies (PoPETS)*, 2022. Cited on page 191.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip HS Torr, and Puneet K Dokania. Calibrating deep neural networks using focal loss. In *NeurIPS*, 2020. Cited on pages 29 and 42.

Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. *Proc. Conf. AAAI Artif. Intell.*, 2015:2901–2907, January 2015. Cited on page 42.

Priyanka Nanayakkara, Mary Anne Smart, Rachel Cummings, Gabriel Kaptchuk, and Elissa M Redmiles. What are the chances? explaining the epsilon parameter in differential privacy. In *USENIX Security Symposium (USENIX Security)*, 2023. Cited on page 82.

Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papemoti, and Nicholas Carlin. Adversary instantiation: Lower bounds for differentially private machine learning. In *IEEE Symposium on Security and Privacy (SP)*, 2021. Cited on page 82.

Milad Nasr, Saeed Mahloujifar, Xinyu Tang, Prateek Mittal, and Amir Houmansadr. Effectively using public data in privacy preserving machine learning. In *Proceedings of the 40th International Conference on Machine Learning*, pages 25718–25732, 2023. Cited on pages 70 and 79.

Yuval Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Ng. Reading digits in natural images with unsupervised feature learning. 2011. Cited on page 23.

Behnam Neyshabur, Srinadh Bhojanapalli, David Mcallester, and Nati Srebro. Exploring generalization in deep learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/10ce03a1ed01077e3e289f3e53c72813-Paper.pdf`. Cited on page 131.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A PAC-bayesian approach to spectrally-normalized margin bounds for neural networks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Skz_WfbCZ`. Cited on page 131.

Huy Le Nguyen, Jonathan R. Ullman, and Lydia Zakynthinou. Efficient private algorithms for learning large-margin halfspaces. In *Algorithmic Learning Theory (ALT)*, 2020. Cited on pages vii, 7, 71, 80, 83, and 193.

Quang Nguyen, Truong Vu, Anh Tran, and Khoi Nguyen. Dataset diffusion: Diffusion-based synthetic data generation for pixel-level semantic segmentation. In *Advances in Neural Information Processing Systems*, volume 36, pages 76872–76892, 2023. Cited on page 66.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. *CoRR*, abs/2112.10741, 2021. URL https://arxiv.org/abs/2112.10741. Cited on page 54.

Y. Niu, K. Tang, H. Zhang, Z. Lu, X. Hua, and J. Wen. Counterfactual vqa: A cause-effect look at language bias. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12695–12705, Los Alamitos, CA, USA, jun 2021. IEEE Computer Society. doi: 10.1109/CVPR46437.2021.01251. URL https://doi.ieeecomputersociety.org/10.1109/CVPR46437.2021.01251. Cited on pages 95 and 203.

Christine Osborne. Statistical calibration: A review. *International Statistical Review / Revue Internationale de Statistique*, 59(3):309–336, 1991. ISSN 03067734, 17515823. URL http://www.jstor.org/stable/1403690. Cited on pages 4, 5, and 29.

Cheng Ouyang, Chen Chen, Surui Li, Zeju Li, Chen Qin, Wenjia Bai, and Daniel Rueckert. Causality-inspired single-source domain generalization for medical image segmentation. *IEEE Transactions on Medical Imaging*, 42(4):1095–1106, 2022. Cited on page 52.

Ashwinee Panda, Xinyu Tang, Vikash Sehwag, Saeed Mahloujifar, and Prateek Mittal. Dp-raft: A differentially private recipe for accelerated fine-tuning. *arXiv: 2212.04486*, 2022. Cited on pages 7, 76, and 193.

Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations (ICLR)*, 2017. Cited on pages 70, 84, and 191.

Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Ulfar Erlingsson. Scalable private learning with PATE. In *International Conference on Learning Representations (ICLR)*, 2018. Cited on pages 70, 84, and 191.

Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. 2022. Cited on pages 5, 33, 35, 36, 37, and 40.

Tim Pearce, Mohamed Zaki, and Andy Neely. Bayesian neural network ensembles. November 2018. Cited on page 5.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019. Cited on page 56.

Francesco Pinto, Giacomo Acciarini, Sascha Metz, Sarah Boufelja, Sylvester Kaczmarek, Klaus Merz, José A Martinez-Heras, Francesca Letizia, Christopher Bridges, and Atılım Güneş Baydin. Towards automated satellite conjunction management with bayesian deep learning. *AI 4 Earth Sciences Workshop NeurIPS*, 2020. Cited on page 4.

Francesco Pinto, Philip Torr, and Puneet K. Dokania. Are vision transformers always more robust than convolutional neural networks? In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021a. URL https://openreview.net/forum?id=CSXa8LJMttt. Cited on page 9.

Francesco Pinto, Harry Yang, Ser-Nam Lim, Philip Torr, and Puneet K. Dokania. Mix-maxent: Improving accuracy and uncertainty estimates of deterministic neural networks. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021b. URL https://openreview.net/forum?id=hlVgM8XcssV. Cited on page 8.

Francesco Pinto, Philip H. S. Torr, and Puneet K. Dokania. An impartial take to the cnn vs transformer robustness contest. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 466–480, Cham, 2022a. Springer Nature Switzerland. ISBN 978-3-031-19778-9. Cited on pages 5 and 9.

Francesco Pinto, Philip HS Torr, and Puneet K Dokania. An impartial take to the cnn vs transformer robustness contest. In *European Conference on Computer Vision*, pages 466–480. Springer, 2022b. Cited on page 53.

Francesco Pinto, Harry Yang, Ser-Nam Lim, Philip Torr, and Puneet K. Dokania. Using mixup as a regularizer can surprisingly improve accuracy & out-of-distribution robustness. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022c. URL https://openreview.net/forum?id=5j6fWcPcc0. Cited on pages 5, 8, and 54.

Francesco Pinto, Yaxi Hu, Fanny Yang, and Amartya Sanyal. PILLAR: How to make semi-private learning more effective. In *2nd IEEE Conference on Secure and Trustworthy Machine Learning*, 2024a. URL https://openreview.net/forum?id=Ps1IHhzx4Z. Cited on pages 7 and 10.

Francesco Pinto, Rauschmayr Nathalie, Tramèr Florian, Torr Philip H.S., and Tombari Federico. Extracting training data from document-based visual question answering. In *ICML*, 2024b. Cited on pages 7 and 11.

Richard Plant, Valerio Giuffrida, and Dimitra Gkatzia. You are what you write: Preserving privacy in the era of large language models. *arXiv preprint arXiv:2204.09391*, 2022. Cited on page 97.

Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12556–12565, 2020. Cited on page 56.

Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009. ISBN 0262170051. Cited on pages 4, 15, 37, 51, and 53.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. Cited on page 64.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, 2019. Cited on page 98.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020. Cited on pages 58 and 151.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. Cited on page 54.

Gonzalo Ramos, Christopher Meek, Patrice Simard, Jina Suh, and Soroush Ghorashi. Interactive machine teaching: a human-centered approach to building machine-learned models. *Human–Computer Interaction*, 35(5-6):413–451, 2020. Cited on page 156.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do CIFAR-10 classifiers generalize to CIFAR-10? June 2018a. Cited on page 23.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do CIFAR-10 classifiers generalize to CIFAR-10? *arxiv:1806.00451*, 2018b. Cited on page 89.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *CoRR*, abs/1902.10811, 2019. URL `http://arxiv.org/abs/1902.10811`. Cited on pages 23 and 35.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*, 2020. Cited on page 57.

Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses, 2021. Cited on page 35.

Shruti Rijhwani, Antonios Anastasopoulos, and Graham Neubig. OCR Post Correction for Endangered Language Texts. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5931–5942, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.478. URL `https://aclanthology.org/2020.emnlp-main.478`. Cited on page 202.

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Skdvd2xAZ`. Cited on pages 23 and 120.

Rebecca Roelofs, Nicholas Cain, Jonathon Shlens, and Michael Curtis Mozer. Mitigating bias in calibration error estimation. September 2020. Cited on page 42.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. Cited on pages 52, 54, and 56.

Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Map-guided curriculum domain adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. doi: 10.1109/TPAMI.2020.3045882. Cited on pages 51 and 53.

A Sanyal, P H S Torr, and P K Dokania. Stable rank normalization for improved generalization in neural networks and GANs. In *ICLR*, 2020. Cited on pages 23 and 116.

Amartya Sanyal, Yaxi Hu, and Fanny Yang. How unfair is private learning? In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2022. Cited on page 197.

Mert Bulent Sariyildiz, Karteek Alahari, Diane Larlus, and Yannis Kalantidis. Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. Cited on page 55.

Robin Schaefer and Clemens Neudecker. A two-step approach for automatic OCR post-correction. In *Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 52–57, Online, December 2020. International Committee on Computational Linguistics. URL https://www.aclweb.org/anthology/2020.latechclfl-1.6. Cited on page 202.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022a. Cited on page 56.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022b. Cited on page 56.

Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 3179–3189. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/a981f2b708044d6fb4a71a1463242520-Paper.pdf. Cited on pages 117 and 126.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9573–9585. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/6cfe0e6127fa25df2a0ef2ae1067d915-Paper.pdf. Cited on pages 33 and 38.

Yuge Shi, Imant Daunhawer, Julia E. Vogt, Philip H.S. Torr, and Amartya Sanyal. How robust are pre-trained models to distribution shift? In *International Conference on Learning Representations (ICLR)*, 2023. Cited on pages 88 and 196.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (SP)*, 2017. Cited on pages 6, 7, and 70.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019. Cited on page 55.

Qingyi Si, Fandong Meng, Mingyu Zheng, Zheng Lin, Yuanxin Liu, Peng Fu, Yanan Cao, Weiping Wang, and Jie Zhou. Language prior is not the only shortcut: A benchmark for shortcut learning in vqa. In *Conference on Empirical Methods in Natural Language Processing*, 2022. URL https://api.semanticscholar.org/CorpusID:252780087. Cited on page 203.

Ray Smith. An overview of the tesseract ocr engine. In *ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition*, pages 629–633, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-2822-8. URL https://storage.googleapis.com/pub-tools-public-publication-data/pdf/33418.pdf. Cited on page 99.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6048–6058, 2023a. Cited on pages 97 and 108.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. *arXiv preprint arXiv:2305.20086*, 2023b. Cited on pages 96 and 97.

Shuang Song, Om Thakkar, and Abhradeep Thakurta. Characterizing private clipped gradient descent on convex generalized linear problems. *arxiv:2006.06783*, 2020. Cited on page 76.

Shuang Song, Thomas Steinke, Om Thakkar, and Abhradeep Thakurta. Evading the curse of dimensionality in unconstrained private glms. 2021. Cited on pages 79 and 80.

J. Taylor, B. Earnshaw, B. Mabey, M. Victors, and J. Yosinski. Rxrx1: An image set for cellular morphological variation across many experimental batches. In *International Conference on Learning Representations (ICLR)*, 2019. Cited on pages x, 161, and 162.

Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 13888–13899. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/36ad8b5f42db492827016448975cc22d-Paper.pdf`. Cited on page 15.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290, 2022. Cited on pages 96 and 105.

Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. Hierarchical multimodal transformers for multi-page docvqa. *arXiv preprint arXiv:2212.05935*, 2022. Cited on page 202.

Rubèn Tito, Khanh Nguyen, Marlon Tobaben, Raouf Kerkouche, Mohamed Ali Souibgui, Kangsoo Jung, Lei Kang, Ernest Valveny, Antti Honkela, Mario Fritz, and Dimosthenis Karatzas. Privacy-aware document visual question answering. *arXiv preprint arXiv:2312.10108*, 2023. Cited on pages 95, 97, and 203.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. December 2020. Cited on pages 33 and 35.

Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. *arXiv preprint arXiv:2302.07944*, 2023. Cited on pages 52 and 55.

Florian Tramer and Dan Boneh. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations (ICLR)*, 2021. Cited on pages 7, 70, 75, 82, and 86.

Florian Tramèr, Gautam Kamath, and Nicholas Carlini. Considerations for differentially private learning with large-scale public pretraining. *arXiv:2212.06470*, 2022. Cited on pages 71 and 86.

Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal Gal. Uncertainty estimation using a single deep deterministic neural network. In *ICML*, 2020. Cited on pages 5, 16, and 23.

Joaquin Vanschoren, Jan N Van Rijn, Bernd Bischl, and Luis Torgo. Openml: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 2014. Cited on pages xiv, 192, and 193.

V. Vapnik. Principles of risk minimization for learning theory. In *Proceedings of the 4th International Conference on Neural Information Processing Systems*, NIPS'91, page 831–838, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc. ISBN 1558602224. Cited on page 16.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł Ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. Cited on page 33.

Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant CNNs for digital pathology. *arXiv:1806.03962*, 2018. Cited on pages 86, 190, and 196.

Joshua Vendrow, Saachi Jain, Logan Engstrom, and Aleksander Madry. Dataset interfaces: Diagnosing model failures using controllable counterfactual generation. *arXiv preprint arXiv:2302.07865*, 2023. Cited on pages 55, 64, and 65.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017. Cited on pages 56 and 155.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018. Cited on page 186.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518, 2019. Cited on pages 23 and 35.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022a. Cited on pages 51 and 53.

Ruoyu Wang, Mingyang Yi, Zhitang Chen, and Shengyu Zhu. Out-of-distribution generalization with causal invariant transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 375–385, 2022b. Cited on pages 51, 53, and 54.

Ruoyu Wang, Mingyang Yi, Zhitang Chen, and Shengyu Zhu. Out-of-distribution generalization with causal invariant transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 375–385, 2022c. Cited on page 52.

Xuezhi Wang, Haohan Wang, and Diyi Yang. Measure and improve robustness in nlp models: A survey. *arXiv preprint arXiv:2112.08313*, 2021a. Cited on page 57.

Zihao Wang and Victor Veitch. A unified causal view of domain invariant representation learning. In *ICML 2022: Workshop on Spurious Correlations, Invariance and Stability*, 2022. URL https://openreview.net/forum?id=-l9cpeEYwJJ. Cited on pages 52 and 53.

Zijian Wang, Yadan Luo, Ruihong Qiu, Zi Huang, and Mahsa Baktashmotlagh. Learning to diversify for single domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 834–843, 2021b. Cited on page 57.

Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Sklf1yrYDr. Cited on page 5.

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 *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=g11CZSghXyY. Cited on pages 15 and 20.

Florian Wenzel, Kevin Roth, Bastiaan S Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the bayes posterior in deep neural networks really? February 2020. Cited on page 5.

Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019a. Cited on page 189.

Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019b. Cited on pages 35 and 116.

Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen. Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1206–1217, 2023. Cited on page 66.

Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *ArXiv preprint arXiv:2006.09994*, 2020a. Cited on pages 35 and 39.

Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *ArXiv preprint arXiv:2006.09994*, 2020b. Cited on pages 56 and 57.

Kai Yuanqing Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=gl3D-xY7wLq. Cited on pages 54, 56, and 145.

I. Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arxiv:1905.00546*, 2019. Cited on pages 75 and 196.

Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *ACM SIGSAC Conference on Computer and Communications Security*, 2022. Cited on pages 6 and 70.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *IEEE computer security foundations symposium (CSF)*, 2018. Cited on page 82.

Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. Opacus: User-friendly differential privacy library in PyTorch. *arXiv:2109.12298*, 2021. Cited on page 189.

Da Yu, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. In *International Conference on Learning Representations (ICLR)*, 2021a. Cited on pages vii, 70, 71, 80, 84, 189, and 194.

Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Large scale private learning via low-rank reparametrization. In *International Conference on Machine Learning (ICML)*, 2021b. Cited on page 70.

Runpeng Yu, Songhua Liu, Xingyi Yang, and Xinchao Wang. Distribution shift inversion for out-of-distribution prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3592–3602, June 2023. Cited on page 55.

Jianhao Yuan, Francesco Pinto, Adam Davies, and Philip Torr. Not just pretty pictures: Toward interventional data augmentation using text-to-image generators. In *ICML*, 2024. URL `https://api.semanticscholar.org/CorpusID:264406016`. Cited on pages 5 and 9.

Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis E H Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-Token ViT: Training vision transformers from scratch on ImageNet. Cited on page 33.

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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. Cited on page 57.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.87. URL `https://dx.doi.org/10.5244/C.30.87`. Cited on pages 19 and 22.

Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Counterfactual memorization in neural language models. *CoRR*, abs/2112.12938, 2021a. URL `https://arxiv.org/abs/2112.12938`. Cited on pages 102 and 103.

Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Counterfactual memorization in neural language models. *arXiv preprint arXiv:2112.12938*, 2021b. Cited on page 97.

Chongzhi Zhang, Mingyuan Zhang, Shanghang Zhang, Daisheng Jin, Qiang Zhou, Zhongang Cai, Haiyu Zhao, Shuai Yi, Xianglong Liu, and Ziwei Liu. Delving deep into the generalization of vision transformers under distribution shifts. June 2021c. Cited on page 5.

Chongzhi Zhang, Mingyuan Zhang, Shanghang Zhang, Daisheng Jin, Qiang Zhou, Zhongang Cai, Haiyu Zhao, Shuai Yi, Xianglong Liu, and Ziwei Liu. Delving deep into the generalization of vision transformers under distribution shifts. June 2021d. Cited on pages 33 and 36.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. Cited on page 57.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=r1Ddp1-Rb`. Cited on pages 14, 15, and 23.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. Cited on page 62.

Xingxuan Zhang, Yue He, Renzhe Xu, Han Yu, Zheyan Shen, and Peng Cui. Nico++: Towards better benchmarking for domain generalization, 2022. Cited on page 56.

Long Zhao, Ting Liu, Xi Peng, and Dimitris Metaxas. Maximum-entropy adversarial data augmentation for improved generalization and robustness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. Cited on page 57.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. Cited on page 154.

Yingxue Zhou, Steven Wu, and Arindam Banerjee. Bypassing the ambient dimension: Private SGD with gradient subspace identification. In *International Conference on Learning Representations (ICLR)*, 2021. Cited on page 80.

Yuqing Zhu, Xiang Yu, Manmohan Chandraker, and Yu-Xiang Wang. Private-knn: Practical differential privacy for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. Cited on page 191.

Laurent Zwald and Gilles Blanchard. On the convergence of eigenspaces in kernel principal component analysis. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2005. Cited on pages 173 and 180.

[heading=bibintoc]