Université de Montréal

Improving predictive behavior under distributional shift

par

Faruk Ahmed

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée en vue de l'obtention du grade de
Philosophiæ Doctor (Ph.D.)
en Informatique (l'intelligence artificielle)

August 8, 2023

Université de Montréal

Faculté des arts et des sciences

Cette thèse intitulée

Improving predictive behavior under distributional shift

présentée par

Faruk Ahmed

a été évaluée par un jury composé des personnes suivantes :

Dhanya Sridhar
(président-rapporteur)

Aaron Courville
(directeur de recherche)

Guillaume Lajoie
(membre du jury)

Hakan Bilen
(examinateur externe)

William Arbour
(représentant du doyen de la FESP)

Résumé

L'hypothèse fondamentale guidant la pratique de l'apprentissage automatique est qu'en phase de test, les données sont *indépendantes et identiquement distribuées* à la distribution d'apprentissage. En pratique, les ensembles d'entraînement sont souvent assez petits pour favoriser le recours à des biais trompeurs. De plus, lorsqu'il est déployé dans le monde réel, un modèle est susceptible de rencontrer des données nouvelles ou anormales. Lorsque cela se produit, nous aimerions que nos modèles communiquent une confiance prédictive réduite. De telles situations, résultant de différentes formes de changement de distribution, sont incluses dans ce que l'on appelle actuellement les situations *hors distribution* (OOD). Dans cette thèse par article, nous discutons des aspects de performance OOD relativement à des changement de distribution sémantique et non sémantique – ceux-ci correspondent à des instances de détection OOD et à des problèmes de généralisation OOD.

Dans le premier article, nous évaluons de manière critique le problème de la détection OOD, en se concentrant sur l'analyse comparative et l'évaluation. Tout en soutenant que la détection OOD est trop vague pour être significative, nous suggérons plutôt de détecter les anomalies sémantiques. Nous montrons que les classificateurs entraînés sur des objectifs auxiliaires auto-supervisés peuvent améliorer la sémanticité dans les représentations de caractéristiques, comme l'indiquent notre meilleure détection des anomalies sémantiques ainsi que notre meilleure généralisation.

Dans le deuxième article, nous développons davantage notre discussion sur le double objectif de robustesse au changement de distribution non sémantique et de sensibilité au changement sémantique. Adoptant une perspective de compositionnalité, nous décomposons le changement non sémantique en composants systématiques et non systématiques, la généralisation en distribution et la détection d'anomalies sémantiques formant les tâches correspondant à des compositions complémentaires. Nous montrons au moyen d'évaluations empiriques sur des tâches synthétiques qu'il est possible d'améliorer simultanément les performances sur tous ces aspects de robustesse et d'incertitude. Nous proposons également une méthode simple qui améliore les approches existantes sur nos tâches synthétiques.

Dans le troisième et dernier article, nous considérons un scénario de boîte noire en ligne dans lequel non seulement la distribution des données d'entrée conditionnées sur les étiquettes change de l'entraînement au test, mais aussi la distribution marginale des étiquettes. Nous montrons que sous de telles contraintes pratiques, de simples estimations probabilistes en ligne du changement d'étiquette peuvent quand même être une piste prometteuse.

Nous terminons par une brève discussion sur les pistes possibles.

Mots-clés: Changement de distribution, détection d'anomalies

Abstract

The fundamental assumption guiding practice in machine learning has been that test-time data is *independent and identically distributed* to the training distribution. In practical use, training sets are often small enough to encourage reliance upon misleading biases. Additionally, when deployed in the real-world, a model is likely to encounter novel or anomalous data. When this happens, we would like our models to communicate reduced predictive confidence. Such situations, arising as a result of different forms of distributional shift, comprise what are currently termed *out-of-distribution* (OOD) settings. In this thesis-by-article, we discuss aspects of OOD performance with regards to semantic and non-semantic distributional shift — these correspond to instances of OOD detection and OOD generalization problems.

In the first article, we critically appraise the problem of OOD detection, with regard to benchmarking and evaluation. Arguing that OOD detection is too broad to be meaningful, we suggest detecting semantic anomalies instead. We show that classifiers trained with auxiliary self-supervised objectives can improve semanticity in feature representations, as indicated by improved semantic anomaly detection as well as improved generalization.

In the second article, we further develop our discussion of the twin goals of robustness to non-semantic distributional shift and sensitivity to semantic shift. Adopting a perspective of compositionality, we decompose non-semantic shift into systematic and non-systematic components, along with in-distribution generalization and semantic anomaly detection forming the complementary tasks. We show by means of empirical evaluations on synthetic setups that it is possible to improve performance at all these aspects of robustness and uncertainty simultaneously. We also propose a simple method that improves upon existing approaches on our synthetic benchmarks.

In the third and final article, we consider an online, black-box scenario in which both the distribution of input data conditioned on labels changes from training to testing, as well as the marginal distribution of labels. We show that under such practical constraints,

simple online probabilistic estimates of label-shift can nevertheless be a promising approach. We close with a brief discussion of possible avenues forward.

Keywords: Distributional shift, anomaly detection.

Contents

List of tables

List of figures

List of abbreviations

wrt	**w**ith **r**espect **t**o
IID	**i**ndependent and **i**dentically **d**istributed
OOD	**o**ut-**o**f-**d**istribution
MLP	**m**ulti-**l**ayer **p**erceptron
CNN	**c**onvolutional **n**eural **n**etwork
RNN	**r**ecurrent **n**eural **n**etwork
LSTM	**l**ong **s**hort-**t**erm **m**emory
BN	**b**atch **n**ormalization
ReLU	**re**ctified **l**inear **u**nit
SGD	**s**tochastic **g**radient **d**escent
MSE	**m**ean **s**quared **e**rror
AUROC	**a**rea **u**nder the **r**eceiver-**o**perating-**c**haracteristics curve
AUPR	**a**rea **u**nder the **p**recision-**r**ecall curve
NLL	**n**egative **l**og-**l**ikelihood
ERM	**e**mpirical **r**isk **m**inimization
IRM	**i**nvariant **r**isk **m**inimization
DRO	**d**istributionally **r**obust **o**ptimization
MMD	**m**aximum **m**ean **d**iscrepancy
KL	**K**ullback-**L**eibler
AI	**a**rtificial **i**ntelligence
ML	**m**achine **l**earning

List of symbols

Symbol	*Meaning*	*Notes*
$[N]$	First N-set	The set of first N natural numbers, $\{1 \cdots N\}$
$\boldsymbol{A}^T$	Matrix transpose	Transpose of the matrix $\boldsymbol{A}$
$\mathcal{I}(\cdot)$	Iverson bracket	Equals 1 when the predicate $\cdot$ is true, 0 otherwise
$\odot$	Hadamard operator	$\boldsymbol{x}_1 \odot \boldsymbol{x}_2$ denotes element-wise multiplication of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$
$\left\|\left\| \cdot \right\|\right\|$	Euclidean norm	$\|\|\boldsymbol{x}\|\| = \sqrt{\sum_i x_i^2}$
$\boldsymbol{x}$	Input variable	In this thesis, most often an image
y	Output variable	In this thesis, most often an object category
$\boldsymbol{\theta}$	Parameters of a model	Typically a vector; individual elements are θ

Acknowledgements

I would like to thank my advisor, Aaron Courville, for providing me with all the freedom to explore any squiggles of thought (within reason), for being an inspirational intellectual influence on my thinking, and for being unfailingly supportive at all times.

Aside from Aaron, I am grateful to have had the opportunity to work with some excellent minds during my years: Chin-Wei Huang, Samuel Lavoie, Kundan Kumar, Alexandre Lacoste, Phong Nguyen, Harm van Seijen, and Yoshua Bengio. Ishaan Gulrajani, Tim Cooijmans, David Krueger, Olexa Bilaniuk, and Rachel Rolland have significantly influenced the ideas described in this thesis, and I am indebted to you for all the things you have taught me.

I am fortunate to have been part of Mila, a wonderful institute made of wonderful people. Mohammad Pezeshki and Reyhane Askari have educated me about Persian cuisine, and helped me get through thick and thin. César Laurent, Çağlar Gülçehre, Kyle Kastner, Lluís Castrejón, and the rest of the gang have been upstanding partners in crime. Thank you all for your support and presence.

Towards the end of my studies, I spent a very edifying year as a visiting student at the Mayo Clinic. I am thankful to John Kalantari, Kia Khezeli, and Nicholas Chia for providing me with the opportunity to get some hands-on experience with very real problems. Being advised by a team of top-notch researchers and practitioners in medicine – Celine Vachon, Stacey Winham, Dr. Tufia Haddad, Dr. Sadia Khanani – was an illuminating experience about the realities behind the problems that AI is expected to help us solve.

Finally, I want to thank my family, for patiently maintaining their unjustified faith in me.

Chapter 1

Introduction

In a famous article (Turing, 1950), Alan Turing asks "can machines think?". An attempt at answering this question calls for axiomatic definitions of what it means for something to be a machine, and what it means to think. Since humans have varying opinions about the constitution of thought (and if we are machines), Turing suggested substituting the philosophical question with an empirical one: can an engineered, non-human entity confuse a human interrogator trying to identify which of two participants – a real human and this entity – is human, when conversing purely via text? This test, called the *imitation game* by Turing, and since then, the Turing test in his honor, is somewhat less ambiguous than the original question of thoughtfulness in machines, being an empirically resolvable problem statement. This test has remained one of the cornerstones in the philosophy of artificial intelligence (AI), with its critiques and endorsements, as is usual in philosophical discussions (Oppy and Dowe, 2021). The problem of AI, in this view, might then be simplified to be that of simulating human-like cognition purely in an observed-behaviour sense without necessarily following the same, precise mechanisms, or "experiencing" a relatable nature of subjectivity. Searle calls this *weak AI* (Searle, 1980). While one might choose to argue that not much is practically gained by attempting to replicate the complete human experience, the causal process by which we generate data for training AI, and measure subsequent performance in deployment are still intimately tied to the human perception and cognition of reality. This implies that even for practical tasks we want solved, we might benefit from attempting to endow our AI models with similar mechanisms, whether directly from biological inspiration, or indirectly – for example, by encouraging consistent behaviour across multiple contexts and tasks. It would also be pragmatic to evaluate model behavior on benchmarks that closely resemble the real problems that concern us, and check for alignment with human cognitive output on such

tasks; although measuring the predictive behavior of AI systems on controlled, synthetic test beds can provide useful insight for practical development.

Machine learning (ML), a term coined by Arthur Samuel, is considered a subfield of AI, and began to receive wider attention in the late 50s and early 60s. Samuel's prediction was that "programming computers to learn from experience should eventually eliminate the need for much of [existing] detailed programming effort" (Samuel, 1959). Machine learning was understood to be the principle of learning from data. For example, for the game of *checkers*, a learning machine could learn, from existing gameplay histories, the coefficients of a polynomial that outputs checkers moves, instead of following rules explicitly coded using expert-knowledge. An operational definition of the term, in the spirit of Turing, was developed by Tom Mitchell (Mitchell, 1997), and is the most oft-quoted when it comes to a definition of machine learning:

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."

One must track this improvement on new examples and not only on instances already experienced by the program, since a machine can simply memorize its past experiences. The underlying statistical assumption is that the *training set* consists of instances that are independently and identically distributed (IID) according to the natural distribution of data. The performance, if measured on a separate sampling of IID instances, tells us if the program has meaningfully learned the task, i.e. it can *generalize*.

A common family of tasks comprises of *supervised learning* problems. Expressed notationally, our goal is to map inputs x to outputs y, given a set of training examples $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}, (x^{(i)}, y^{(i)}) \sim p_D(x, y)$, where p_D is the true data distribution. The labels are typically provided by human annotators. We then learn to perform a transformation $x \mapsto f(x)$, such that we minimize the *empirical risk* for a specified loss function $\ell(.,.)$, given as

$$\mathcal{R}_f(\mathcal{D}) = \frac{1}{m} \sum_{i=1}^{m} \ell(y^{(i)}, f(x^{(i)})). \tag{1.0.1}$$

For the learned function to generalize beyond the training set, we would like the *true risk*, $\mathbb{E}_{p_D}[\ell(y, f(x))]$, to be correspondingly minimized. Since we do not have access to the true risk, the way we estimate generalization in practice is by holding out a *test set* of examples, $\{(x^{(i)}, y^{(i)})\}_{i=m_{train}+1}^{m_{train}+m_{test}}, (x^{(i)}, y^{(i)}) \sim p_D(x, y)$, from our collected dataset and evaluating the

trained model on this test set. This maintains the assumption of IID sampling when testing the model's skill at performing its task on new instances of true experiences.

Let us consider a more concrete example: say we have access to a set of training examples of pictures of various objects x, that have been labelled y by human labelers, and we wish to learn how to recognize the object present in an image. Such an object recognition task is a very common and practical task in applications of computer vision. Since the task of interest is that of detecting the identity of the object in the foreground of the image x, we can think of x as being a composition of the foreground object x_f and the background component x_b. Further thought reveals that there are components from within the object that are relatively less related to its identity. For example, most of us would agree that the color of a chair is irrelevant for recognizing chairs[1]. A human who has seen a few chairs would typically not struggle to recognize one if it had a novel color. Yet, if we only ever provided examples of chairs that are red to a learning machine, without specifying any learning criteria save that of minimizing categorization-errors, it is perfectly reasonable for such a learner to assume that redness might be a defining characteristic of chairness. This undesirable inference is particularly exacerbated if, in our training set, chairs were the only object that was consistently red. Imagine a binary classification task where all chairs are red due to selection bias, and the other object of interest, say bananas, never appears in red (for the sake of this thought experiment, let us assume the background is always a bland white). Now "*Is it red?*" can be used exclusively as a predictive rule, so much so that a learner does not even need to learn anything about bananas. Equally worryingly, when such a program is presented with a red object which is neither a chair nor a banana, the output can be a confident "chair!". Evidently, such a predictive rule is likely to be ineffective at large. If we present a learning machine with an infinite data stream, such biases ought to be accounted for, since we can expect accidental correlations to be minimized with richer sampling. However, acquiring large datasets, labeling them, and then training for a correspondingly longer period of time are all costly endeavours, and in some real-world situations, such as with problems in healthcare, a small and potentially-biased training set is all we have at hand.

In the presence of such possibilities, it feels necessary to, at the very least, estimate generalization beyond a sampling of a particular data collection process before deploying an AI model in the wild. In development, a model might work reassuringly well, displaying excellent test set performance. However, it might only have picked up on a bias, which continues to transfer across to the similarly collected test set, but does not exist in general. Would we trust such a learner out of the lab and in our lives, driving our self-driving cars or diagnosing our diseases?

[1]In this thesis, we set aside deeper philosophical perspectives about what really makes for a chair.

Compositional generalization. Cognitive scientists and philosophers have theorized that humans are able to generalize well to novel contexts, and construct novel structures imaginatively because our internal representations are *compositional* in nature and based on *primitive* concepts (Fodor and Pylyshyn, 1988; Marcus, 2001). These primitives can be productively composed to generate new sentences, for example (and more controversially, new thought (Fodor, 1975)). This "algebraic" compositionality is posited to be the explanation for why humans excel at understanding a novel sentence by virtue of understanding another. In the field of deep learning, this has primarily been discussed in the context of languages (Lake and Baroni, 2018) following along the classic descriptions of systematicity[2] in language (Chomsky, 1957) but we can presume similar ideology in other contexts, such as with visual data, for example. We can consider an image x to be composed of an object component h_o, and all other components that are unrelated to the identity h_u, such as the background or object-color[3]. We can express this notationally, for a "composition function" $\mathcal{C}$

$$x = \mathcal{C}(h_o, h_u), \text{ such that } h_o \perp h_u \text{ given } y. \tag{1.0.2}$$

Now we can consider the conditional marginals $p(h_o \mid y)$ and $p(h_u \mid y)$ and ask where the sampling should be from at test time. Sampling outside of $p(h_o \mid y)$ can let us measure the awareness of novelty, and sampling from outside $p(h_u \mid y)$ lets us estimate compositional generalization at test time. All of this prudence about testing generalization applies especially strongly for cases where few inductive biases have been applied, and where the major driving force is that of a simple loss term, such as a misclassification penalty. With additional constraints, we can implicitly bias learners to look for explanations with certain characteristics, or explicitly discourage fits to confounding correlations – if we can identify specific biases, we can discourage reliance on them. We can ask ourselves why some mappings or explanations make more sense to us, and then try to encourage a proclivity towards such mappings in our model-design phase.

In this thesis, we shall touch on such problems and perspectives through three articles.

- In the first article, we discuss one aspect of trusthworthy and robust AI models – they ought to recognize novelty, and communicate reduced confidence when encountering unfamiliar things. In the earlier example, we might come across an object which is neither a chair nor a banana. We provide a critical appraisal of benchmarks in the literature for such problems, recommending more realistic alternatives. Building upon our arguments and intuitions, we propose a method to improve a classifier's sense of

[2] a concept perhaps easier to intuit with the word "recombinability"

[3] assuming we would deem color as irrelevant for a particular object, and the background as being unrelated to identity.

uncertainty when facing novel objects, showing that it leads to better performance on our benchmarks.

- In the second article, we adopt the compositional view and show that such awareness of novelty is but one aspect of reliability – apart from sensitivity to unfamiliar objects, we would also like to be robust to unfamiliar contexts. In the chair/banana example, one might encounter a red or a purple banana, which are less common varieties of banana than the popular Cavendish variant. In order to perform controlled experiments, we create a set of synthetic benchmarks, reflecting tasks requiring different forms of compositional generalization. We also propose a method for improving predictive behavior at such compositional generalization problems.
- Finally, in the third article, we consider real-life problems encountered when deploying models online across multiple locations under resource constraints, and potential approaches to tackling them. Specifically, we consider adapting our predictions from a black-box AI model to deployment contexts associated with particular target-frequencies. For example, if our chair/banana classifier is deployed in a location with vastly more bananas than chairs, we might be able to resolve ambiguous situations taking this prior into account. We consider such label-shift problems combined with a shift in the label-conditioned input distribution (e.g. chairs can look a bit different across different regions of the world). We evaluate existing methods for a mix of synthetic and realistic problems, suggesting some heuristics to potentially improve performance in these deployment settings.

In the following chapter, we begin with an overview of relevant background information.

Chapter 2

Background

In this chapter we provide background on *deep learning* (LeCun et al., 2015), a term used to describe modern machine learning models that employ hierarchical processing of data, with the adjective *deep* referring to depth in the layers of the hierarchy.

2.1. Deep learning

Deep *neural networks* (McCulloch and Pitts, 1943), are currently the most effective instantiation of statistical models for mapping complex, high-dimensional input data to relatively lower-dimensional outputs. For example, one might want to identify the objects present in an image (Krizhevsky et al., 2012), identify the human action being performed in a video clip (Ji et al., 2012), recognize speech from an audio clip (Bahdanau et al., 2016), or categorize high-level meaning in a textual document (Wang et al., 2016).

Deep neural networks are typically trained *end-to-end* to learn a desired mapping, such as in the tasks described above. In some cases, for example when there is too little data in the domain of interest, we can initialize some components of complex models by *pre-training* them on larger datasets in a related or different domain. For example, for problems in medical imaging, one typically finds improvements when initializing the networks with pre-training on large-scale object recognition datasets, which have little to do with medical images (Raghu et al., 2019). Another way to improve performance can be by additional pre-training with *self-supervised* objectives (Azizi et al., 2022; Lai et al., 2023), where one can leverage unlabelled data (whether in the same, related, or even somewhat different domains)

by artificially creating prediction tasks. The most common form of this approach today involves recognizing artificially-perturbed inputs as being identical in content.

Intuitively, one reason pre-training methods based on different datasets work so well is because the higher-level similarity of the tasks[1] implies that one can get closer to the task of interest by learning on more data to perform a related task. To become a radiologist, one must first learn to see. Additionally, training with self-supervised learning objectives can achieve the goal of incorporating *invariances* into the feature extractors. For example, if we train a model to identify contrast-altered images as being similar in content, we can instill contrast-agnosticism in the feature extraction through this task. Thus, if we know what invariances are likely to be useful for a downstream task (such as color-invariance), we can get our networks started on learning such invariances on a larger, unlabelled dataset, or even a less-related dataset. In Section 2.2, we shall discuss several self-supervised methods.

End-to-end training refers to the method by which an optimization process operates on all model-parameters at the same time, all the way from input to output. A counter-example would be training an image-caption generator by separately training the image-feature extractor and the caption-from-features generation module. *Stochastic gradient descent* (Robbins and Monro, 1951) (SGD) is typically used to learn model parameters over the *training set*, and hyperparameters, such as the learning rate or number of layers in the model, are tweaked by evaluating held-out performance on a *validation set*. A hitherto unseen set called the *test set* is used for final evaluation. This procedure is fairly classical, however, if one intends to deploy one's model out in the world at large, having only trained on a set that is not likely to have been statistically representative of fuller reality, models can behave in unexpected ways due to the *distributional shift* (Amodei et al., 2016) in deployment. Even if one develops methods that are capable of endowing robustness to aspects of distributional shift, one must still tune hyper-parameters of their method using a validation set. In such a setting, a validation set must also exhibit distributional shift in order to accommodate meaningful hyper-parameter tuning or model-selection. However, at test-time, one may encounter a significantly different data distribution that bears little resemblance to either training or validation data. It is unclear at the moment how to best go about handling such situations. We discuss some of the current thinking for handling such *out-of-distribution* instances in Section 2.3.

In the next few subsections, we briefly describe some basic neural network architectures, and the methods used to train them.

[1]for example, two different object recognition tasks can both be considered fundamentally visual discrimination.

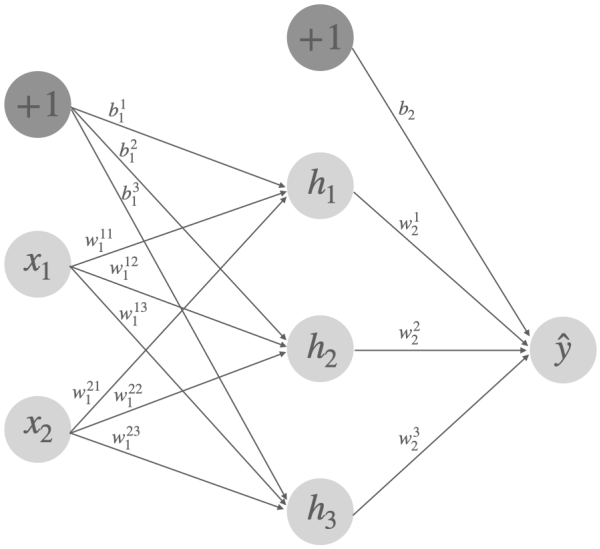


FIG. 1. A feedforward neural network with one hidden layer mapping a 2-dimensional input to a scalar output. The +1 nodes correspond to the bias terms in the affine transformations. The transformation being computed is $\hat{y} = W_2^\top \boldsymbol{h} + b_2$, $\boldsymbol{h} = \sigma(W_1^\top \boldsymbol{x} + \boldsymbol{b_1})$, where σ is a non-linear function.

2.1.1. Multi-layer perceptrons

Let us assume our dataset consists of inputs x which we wish to map to the target y, where $\boldsymbol{x} \in \mathcal{R}^d$, and $y \in [C]$ for classification problems and $y \in \mathcal{R}$ for regression problems (with scalar output dimension). If we model the mapping with a parameterized, hierarchical transform, we would write

$$y = f_L(f_{L-1}(\cdots f_1(\boldsymbol{x}) \cdots)), \tag{2.1.1}$$

where L is the number of *layers* in our model, and f_l is the transformation in the l-th layer. Since a composition of affine functions continues to be affine, such composition is particularly useful when the functions are non-linear. A simple choice for implementing such a transform would be

$$f_l(\boldsymbol{x}) = \sigma(W_l^\top f_{l-1}(\boldsymbol{x}) + \boldsymbol{b}_l), \tag{2.1.2}$$

where W_l is the *weight matrix* and $\boldsymbol{b}_l$ is the bias term associated with the l-th layer, and affinely transforms the input to the layer. The inputs to intermediate layers are referred to as the *activation vectors* from the preceding layers. σ is the non-linearity, also called the *activation function*, that is applied on top of the affine transform, and can take several forms. This overall model is referred to as the *multi-layer perceptron* (MLP), since it can be viewed as a hierarchical extension of the *perceptron* (McCulloch and Pitts, 1943), a linear binary classifier using the rule $y = \mathcal{I}(\boldsymbol{w}^\top \boldsymbol{x} + \boldsymbol{b} > 0)$, trained iteratively by correcting $\boldsymbol{w}$ on mis-classifications by vector addition with $\boldsymbol{x}$.

Some of the choices for the activation function are

$$\text{Heaviside}(x) = \mathcal{I}(x > 0), \tag{2.1.3}$$

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)}, \tag{2.1.4}$$

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}, \tag{2.1.5}$$

$$\text{ReLU}(x) = x \text{ if } x \geq 0, \text{ and } 0 \text{ otherwise}, \tag{2.1.6}$$

$$\text{LeakyReLU}(x) = x \text{ if } x \geq 0, \text{ and } \alpha x \text{ otherwise } (\alpha \ll 1). \tag{2.1.7}$$

While the Heaviside step function is most reminiscent of the perceptron, the non-differentiability of the function renders gradient-based training futile. Hence, the other activation functions, more amenable to gradient-based training, tend to be used – historically, SIGMOID and TANH were preferred (and TANH is still used in *recurrent networks*), but the general non-linearity of choice in feedforward networks is the *rectified linear unit*, or RELU (Glorot et al., 2011). Recently, there have been softer approximates to the ReLU, such as the *Gaussian error linear units* (GELU) (Hendrycks and Gimpel, 2016) which tend to be useful in some modern architectures, such as *transformer*-based language models (Devlin et al., 2018). Given that modern-day feedforward networks tend to avoid step functions, perhaps the name "multi-layer perceptron" is something of a misnomer.

The final layer transforms the penultimate-layer activations into a format applicable for the task being solved. If we are performing binary classification, i.e. we might be modelling a conditional Bernoulli distribution $P(y = 1 \mid \boldsymbol{x})$, we typically apply the sigmoid transform to the scalar output of the last layer. When we are performing multi-class classification, we need to transform the set of *logits* (of dimensionality equalling the number of target categories) output by the final layer into a vector of probabilities corresponding to a conditional Categorical distribution. This is typically performed by the analogue of the sigmoid function for multiple categories, the SOFTMAX transform,

$$P(y = k \mid \boldsymbol{x}) = \frac{\exp a_k}{\sum_{k'} \exp a_{k'}}, \tag{2.1.8}$$

where a_k is the k-th logit output by the model for the input $\boldsymbol{x}$. When the task is regression to an unconstrained scalar target, we can directly output our prediction. If there are constraints on our output space, we can often bake them in: for example, if our outputs are always positive, we can use a RELU or a SOFTPLUS (Dugas et al., 2000) activation at the end.

While MLP modelling seems sensible when there is no shared structure to a fixed-size input, which encourages us to assign different weights to every dimension, we can do better when

structure in the input allows us to share weights – this can improve efficiency, as well as help induce prior knowledge into our model. In the next two sections, we discuss *convolutional neural networks*, which share model parameters across spatial regions in image data, and *recurrent neural networks*, which share parameters temporally for sequence data.

2.1.2. Convolutional neural networks

Let us consider the task of visual object recognition, i.e. we are presented with a natural image, and we would like to identify the object present in it. The two main reasons why we might not want to apply the MLP model of the previous section for such a task are that (1) even slightly differently-sized images would render our model inapplicable, and (2) such input modalities typically exhibit *translation invariance*, which means our understanding of the contents in the image often does not vary at all if the contents are all spatially shifted. The latter point in particular suggests that we might want to develop a model which searches for the same features in all locations of an image, and final results are achieved by summing the results of such searches across all spatial locations. This is what is most commonly done in *convolutional neural networks*, with the intermediate layers consisting of *filters* shared across spatial locations, and features finally being *pooled* globally to provide a feature vector (see Fig. 2). This final vector is then transformed to provide logits. Convolutional operations may also be said to characterize *translation equivariance*, meaning that as the input shifts spatially, the corresponding feature map after performing a convolution operation also shifts the same amount (modulo any *striding* of the filters).

The convolutional aspect in such models comes from the discrete convolution being performed: a *kernel* (a small square matrix) slides over the input to a layer (or the image $\boldsymbol{x}$ in case of the first layer of processing), producing a similarly-sized set of values, which when summed over the number of input channels and transformed with an activation function yields the activation map (see Fig. 2). The final layer of (flattened or pooled) features are typically combined with one or more MLP layers to provide logits, which are processed similarly as for MLPs in the previous section.

2.1.3. Recurrent neural networks

RNNs are models that map sequences to either non-sequential outputs or entire sequences as well, for example, categorizing sentiment in a text-sequence, or translating a text-sequence

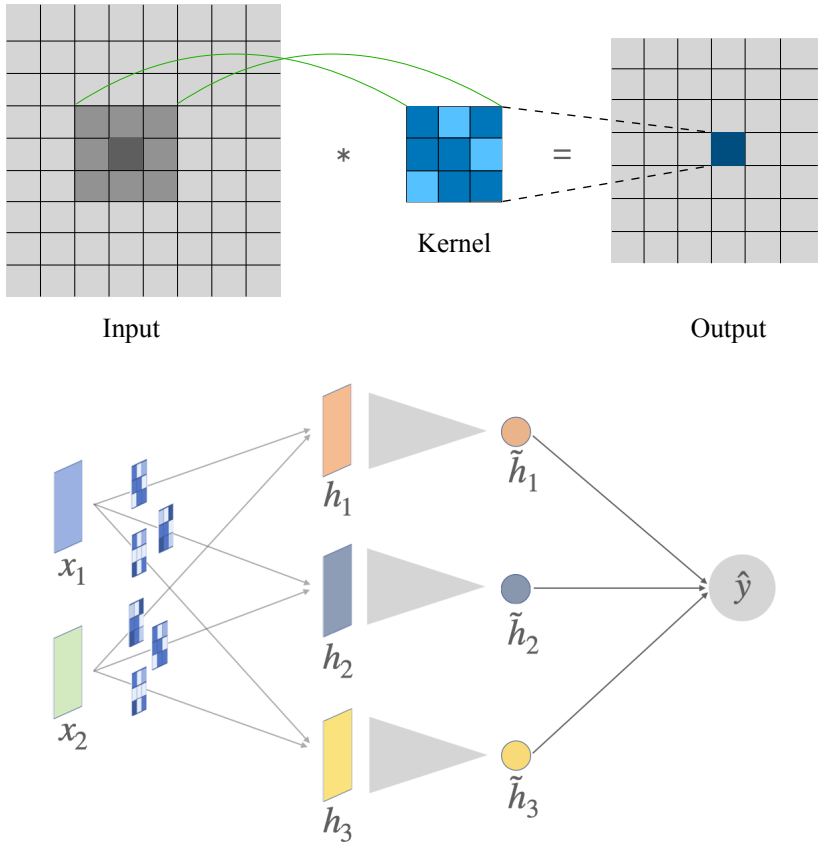


FIG. 2. *(top)* Discrete convolutions are performed by sliding a *kernel* over the input, at each step multiplying overlapping values and summing the products, yielding the corresponding output value at the location where the kernel is centered. Note that this leads to a loss in spatial dimensions, since the valid position for the kernel-center starts at $[\lfloor (K/2) \rfloor, \lfloor (K/2) \rfloor]$ for a $K \times K$-sized kernel. To maintain the same spatial dimensions, we pad the edges of the input so that the output dimensions match the input dimensions. *(bottom)* The basic underlying structure in a convolutional neural network with global average pooling. The input is 2-channel, the spatial analogue of the 2-dimensional input in Fig. 1. Instead of scalar weights multiplying with the inputs, filters are convolved with the spatial inputs, and summing the results of the convolutions at every incoming intermediate node (h_1, h_2, h_3) and applying a non-linearity $\sigma(.)$ element-wise yields a set of feature maps with spatial dimensions. Every feature map can be averaged across their spatial dimensions, producing a vector of average feature activations. This vector, $\tilde{\boldsymbol{h}}$, can now be transformed to the output as before. For a typical image classification problem, one would have several convolutional layers in sequence, before average pooling and a mapping to an output dimension equalling the number of categories. (Bias terms have been omitted to reduce clutter.)

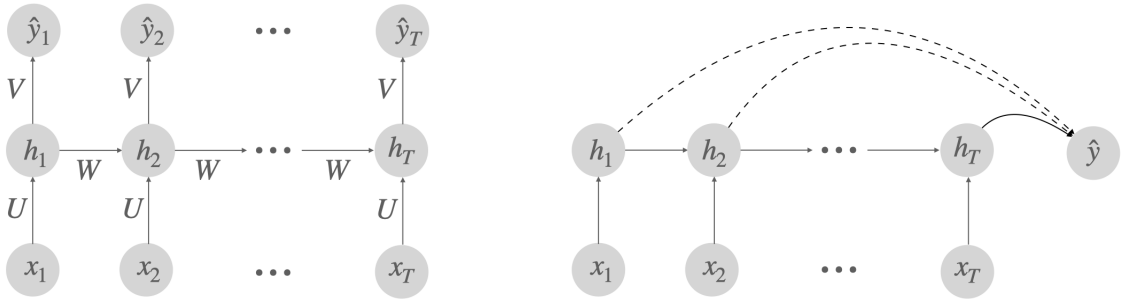


FIG. 3. (*left*) A basic sequence-to-sequence RNN maps an input sequence $\{x_1, \cdots, x_T\}$ to an output sequence $\{\hat{y}_1, \cdots, \hat{y}_T\}$ in a stateful way. (*right*) For a typical sequence classification task, the objective is to map a sequence to a category. One may either transform the final hidden state, h_T to the output, or aggregate over all hidden states over the entire sequence.

from one language to another. This mapping is *stateful*, since there is a temporally evolving hidden state within the network which also determines the output along with the input.

For the case where an input sequence $\{\boldsymbol{x}_t\}$ is mapped to an output sequence $\{y_t\}$, the typical operations for a basic RNN run as follows:

$$\boldsymbol{h}_t = \tanh(W^\top \boldsymbol{h}_{t-1} + U^\top \boldsymbol{x}_t + \boldsymbol{b}), \tag{2.1.9}$$

$$y_t = \text{softmax}(V^\top \boldsymbol{h}_t + \boldsymbol{c}), \tag{2.1.10}$$

where W, U, V are the weight matrices for the hidden-to-hidden, input-to-hidden, and hidden-to-output connections respectively. Since the same weight matrices are used across all steps t in the sequence, RNNs are the temporal analogue of CNNs in terms of parameter-sharing. If the task is to map a sequence to a non-sequential output, one can simply transform the final hidden activations, or aggregate over all hidden states in the sequence. See Fig. 3 for illustrations. Other connectivity patterns exist, such as when generating a sequence from a vector input, or when the predicted outputs at each step are fed back for the prediction at the next step.

Such vanilla RNNs tend to under-perform at tasks requiring memory over longer time-frames. A variant called the *long short-term memory* network, or LSTM (Hochreiter and Schmidhuber, 1997), provides significant improvements, by maintaining a cell-state which can hold information perfectly for as long as necessary, and (softly) erase held information when it is not required anymore. This is performed by using a gating mechanism per dimension of the cell-state which either overwrites in new information in specific locations, or preserves older information.

2.1.4. Recent advances

In this section, we discuss three modern, high-performing variants of the above "classical" model families.

ResNets. ResNet (He et al., 2016) was a CNN model developed by a team at Microsoft Research, winning the 2015 ImageNet classification challenge. The key idea was to add *short-cut* connections across blocks of layers, enabling the learning of *residual* functions. More precisely, while we have so far been computing activations of the l-th layer using $\boldsymbol{h}_l = f_l(\boldsymbol{h}_{l-1})$, ResNets instead compute $\boldsymbol{h}_l = \sigma(\boldsymbol{h}_{l-1} + f_l(\boldsymbol{h}_{l-1}))$, where σ is the activation function, and f_l now only needs to learn a residual signal over the input $\boldsymbol{h}_{l-1}$. f_l can take the form of an additional 2-3 "sub-layers" of processing. This trick enables training far deeper models than was previously doable, because the short-cut connections can avoid issues of gradients *vanishing* over deeper-layer training (we discuss vanishing gradients in Section 2.1.5).

While this flavor of ResNets make training deeper models easier, in a subsequent variant, He et al. (2016) replaced the computation across a residual block to be $\boldsymbol{h}_l = \boldsymbol{h}_{l-1} + \sigma(f_l(\boldsymbol{h}_{l-1}))$. This version, called a *preactivation* ResNet, makes for far longer short-cut connections – theoretically, all residues can be set to zero, in effect copying the input to the last layer no matter how deep the network, modulo downsamplings to match reduced spatial dimensions. Using this trick enables effective training of much deeper models. An alternative model, which increases layer-widths instead of network depth often performs better, and the wide ResNet (Zagoruyko and Komodakis, 2016) is often a default choice for small-scale datasets such as CIFAR-10 and STL-10.

DenseNets. Instead of summing inputs to the outputs of residual blocks, an alternative connectivity pattern for preserving information is to simply concatenate all previous activations for input to a specific layer, which in turn passes on its own activations to all subsequent layers. Specifically, the activation at the l-th layer is computed as $\boldsymbol{h}_l = f_l([\boldsymbol{x}, \boldsymbol{h}_1, \cdots, \boldsymbol{h}_{l-1}]$. This model is called DenseNet (Huang et al., 2017) due to the dense connectivity pattern.

Intuitively, such concatenation should enable easier information flow through layers than summing inputs with residual information. Additionally, one can achieve similar results using fewer parameters than competing models, possibly because useful earlier feature maps do not need to be explicitly re-learned/retained in subsequent layers.

Transformers. A recent family of sequence-to-sequence models called *transformers* (Vaswani et al., 2017) does away with the statefulness in RNN models, by instead learning features that selectively attend to other features across the sequence dimension (termed *self-attention*). This turns the mapping into a feedforward structure with flexible sequence-dimensionality, enabling processing of arbitrary sequence lengths in parallel without the need for a hidden state. This model family has found resounding success at a variety of tasks, such as machine translation (Vaswani et al., 2017), music generation (Huang et al., 2018), protein generation (Madani et al., 2020), and more.

Interestingly, such sequence models have also been shown to be highly performant at data modalities not usually thought of as sequences. For example, the *vision transformer* (ViT) (Dosovitskiy et al., 2020) carves out 16×16 patches from an image and treats the flattened patches as elements of a sequence (rasterized order across the image). When such a transformer model is trained on this sequence with relatively smaller sized datasets such as IMAGENET, the performance tends to be poorer than existing CNN-based models, a failure attributed to the lack of equally powerful inductive biases. However, when trained on much larger datasets, such as Google's internal JFT-300M, the resulting model performs competitively with, or outperforms, CNN models at transfer learning to smaller datasets like CIFAR-100 or IMAGENET.

2.1.5. Training

Training in deep models is performed using *backpropagation*[2], a method derived from the chain-rule in calculus that allows us to compute the gradients of the loss function *wrt* parameters in hierarchical models. These gradients can then be used by a gradient-based optimization method to (iteratively) update the parameters of the model, thereby minimizing the loss.

As an example, consider the model $\hat{y}(\boldsymbol{x}) = f_{\boldsymbol{\theta}_3}(f_{\boldsymbol{\theta}_2}(f_{\boldsymbol{\theta}_1}(\boldsymbol{x})))$, where the composing functions f_i are parameterized with parameters $\boldsymbol{\theta}_i$[3]. Given a loss function $\ell(y, \hat{y})$, we would like to compute $\frac{\partial \ell}{\partial \boldsymbol{\theta}_i}$ for all $\boldsymbol{\theta}_i$ in the model. The method of backpropagation operates in two stages.

[2] http://people.idsia.ch/~juergen/who-invented-backpropagation.html.

[3] we use bold symbols in this chapter to emphasize when variables are non-scalar, but elsewhere in this thesis we skip bolding when context is sufficient.

Forward pass: Compute all activations, pushing forward the input $\boldsymbol{x}$ through the network (consider $\boldsymbol{x} = \boldsymbol{h}_0$),

$$\boldsymbol{h}_k = f_{\boldsymbol{\theta}_k}(\boldsymbol{h}_{k-1}). \tag{2.1.11}$$

Backward pass: Moving backward from the output layer, using $\boldsymbol{u}_{L+1} = 1$, compute

$$\boldsymbol{g}_l = \boldsymbol{u}_{l+1}^\top \frac{\partial f_{\boldsymbol{\theta}_l}(\boldsymbol{h}_l)}{\partial \boldsymbol{\theta}_l}, \tag{2.1.12}$$

$$\boldsymbol{u}_l^\top = \boldsymbol{u}_{l+1}^\top \frac{\partial f_{\boldsymbol{\theta}_l}(\boldsymbol{h}_l)}{\partial \boldsymbol{h}_l}. \tag{2.1.13}$$

The $\boldsymbol{g}_l$ terms are the gradients for the parameters $\boldsymbol{\theta}_l$. The forward pass is necessary because the expression for the term $\frac{\partial f_{\boldsymbol{\theta}_l}(\boldsymbol{h}_l)}{\partial \boldsymbol{\theta}_l}$ (and in nearly all cases, the accumulated terms in $\boldsymbol{u}_{l+1}$ as well) require the activation values corresponding to the input $\boldsymbol{x}$. Thus, computing the gradients usually requires storing activations as intermediates beforehand.

These gradients can be used to update parameters using *gradient descent* (Hadamard, 1908),

$$\boldsymbol{\theta}_l^{\text{new}} := \boldsymbol{\theta}_l^{\text{old}} - \alpha \frac{1}{N} \sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}_l} \ell(y, \hat{y}(\boldsymbol{x}))\Big|_{\boldsymbol{\theta}_l^{\text{old}}}, \tag{2.1.14}$$

where α controls the length of the update step and is therefore called the *learning rate*. As the formula suggests, given N training data-points, the update step is the average gradient computed over all the points, with the model evaluated at the current state of the parameters. Such an update step can be expensive to compute for large datasets, since we would need to evaluate the entire dataset before every update. A more efficient approximation would be to perform the update for every training point, or for *mini-batches* of training points. This procedure is called *stochastic gradient descent* (SGD) (Robbins and Monro, 1951), and is an approximation to the true gradient. Fortuitously, the noise induced by the approximation often tends to be quite helpful when training neural networks. Per one intuition, the corresponding loss surfaces can have non-robust optima, and noise can help sidestep these *sharp minima* that are likely to correspond to poor generalization. Additionally, the learning procedure can simply make quicker progress due to the efficiency of computing gradients on smaller sample-sizes. Most modern implementations of SGD provide improvements by including momentum terms that accumulate past gradients and/or adaptively scale gradients to avoid instabilities (Tieleman and Hinton, 2012; Kingma and Ba, 2014).

Vanishing and exploding gradients. In Eq. 2.1.12, we saw that we need to use the accumulated product of the terms $\boldsymbol{J}_l = \frac{\partial \boldsymbol{h}_{l+1}}{\partial \boldsymbol{h}_l}$ to compute the gradient for parameters. For sake of understanding, if we assume $\boldsymbol{J}_l$ remains constant across layers, calling it $\boldsymbol{J}$, we can

see that the gradient $\boldsymbol{g}_l$ is proportional to $\boldsymbol{J}^{L-l}$. This suggests that if the maximum absolute value of the eigenvalues of $\boldsymbol{J}$ significantly exceeds 1, gradient magnitudes can explode when training very deep networks. Similarly, they may vanish if the maximum absolute eigenvalue is significantly smaller than 1. Exploding gradients can be managed by simply clipping large gradient magnitudes (while retaining the direction). Shrinking gradients are harder to handle, but certain advances have proven successful. For example, the use of non-saturating activation functions such as the ReLU, or architectures with short-cut connections over longer ranges (as in ResNets and DenseNets discussed earlier) have significantly ameliorated the issue of vanishing gradients in modern feedforward networks. Careful initialization strategies, parameterization choices, and architectural tweaks have been similarly useful for sequence models.

2.2. Representation learning with self-supervision

An inspirational line of work for models of cognition and neural networks originated in the PDP research group led by James McLelland and David Rumelhart in the 80s, promoting *parallel distributed processing* in two volumes (Rumelhart et al., 1986). This view argues for distributed representations computed in parallel (for example, a hidden layer in an MLP) over localized, symbolic structure (for example, knowledge graphs with predefined entities and relations). The intuitive advantage of a PDP representation scheme is that exponentially many more "concepts" can be represented with the same number of units, since *symbolic* representation only allows for individual concepts per unit. Deep learning is at its heart largely PDP-based representation learning, with many levels of hierarchy.

Learning such representations is usually done in the context of a downstream task – we would like to learn efficient representations that result in improved performance at the task. There could be other desiderata: for example, if the task requires easy manipulation of underlying factors, then encouraging the representation units to be independent of each other would allow for easy slider-based adjustment. Representation learning can be performed in an unsupervised manner as well. Learning to perform a task that does not require labels, such as reconstruction of data, can often result in representations that are useful for classification, as shown in Vincent et al. (2008), for example.

Self-supervised methods may be considered a subset of *unsupervised learning* (no labels), where labels are auto-generated from the data using the context of a pretext task. Training can now be done using latest developments in supervised learning methods. These representations,

initially developed by a model for meeting the labelling goals of the pretext task, can subsequently be evaluated for a downstream task such as object recognition. This evaluation is most commonly done by learning a linear classifier on top of the learned representation. Research in this direction has seen a rising wave of attention in the past years, with compelling successes in problems such as language (Devlin et al., 2018) and object recognition in images (Doersch et al., 2015; Pathak et al., 2016; Noroozi and Favaro, 2016; Zhang et al., 2017; van den Oord et al., 2018; Gidaris et al., 2018; Caron et al., 2018). The gap between test set performances using fully supervised methods vs linear classification on self-supervised representations has been drawing ever closer (Tomasev et al., 2022).

Most of self-supervised learning research in the context of image classification started gaining prominence 2015 onwards, and for the next two years or so, most methods mostly involved predictions on patches or autoencoding. For example, Doersch et al. (2015) predicts the relative location of a patch in an image with respect to another; Pathak et al. (2016) trains autoencoders to inpaint images; Noroozi and Favaro (2016) solve jigsaw puzzles by cutting up an image into patches, shuffling them and learning how to put them back together; Zhang et al. (2017) autoencodes across channels – given channels of the input, predict the other channels. Gidaris et al. (2018) showed that tasking a neural network to predict the angle by which an image has been rotated results in learning features significantly useful for object classification, detection, and segmentation. In the next few sections, we will briefly discuss *contrastive methods*, which boosted performances significantly over these earlier approaches. We will also briefly cover newer non-contrastive methods, which are currently on par with contrastive methods, while removing some of the difficulties inherent in contrastive approaches.

2.2.1. Contrastive predictive coding (CPC)

A patch-prediction method, CPC (van den Oord et al., 2018; Hénaff et al., 2019) combined several deep learning techniques to demonstrate very compelling performance on a wide range of tasks: speech recognition, image classification, object detection, classification tasks with natural language data, and reinforcement learning. We will focus on the application to image classification in our discussion, since that is our use-case in the first article.

The goal is to extract patch-encodings from an image in a way that optimizes mutual information between related patches, for a specified notion of relatedness. This encourages an implicit understanding of how things "fit together", in the spirit of Doersch et al. (2015); Noroozi and Favaro (2016), leading to representations that have been shown to be very useful

for downstream tasks. More concretely, a neural network encoder g_{enc} produces encodings of regularly spaced-out, overlapping patches in an image,

$$z_{ij} = g_{\text{enc}}(x_{ij}), \tag{2.2.1}$$

where i and j are row and column indices of the patch (for example, a 256×256 image might be broken into a 7×7 grid of overlapping patches). A masked convolutional network, the *context network* g_{context}, provides context vectors per location [4], which are linearly transformed to predict encodings further down. The context network, being fully convolutional without subsampling, outputs a 7×7 grid,

$$c_{ij} = g_{\text{context}}(z_{ij}). \tag{2.2.2}$$

A matrix W_k is used to linearly transform the context vectors into predictions for patches k rows down,

$$\hat{z}_{i+k,j} = W_k^\top c_{ij}. \tag{2.2.3}$$

InfoNCE. Let us assume that for a context c, we are given a positive sample from $p(x \mid c)$, and $N-1$ negative samples from $p(x)$. In this case, the positive sample would be an encoding of a patch at the right location in the image, and the negative samples are encodings of patches from elsewhere, both within the same image as well as different images.

Given the set of samples $X = \{x_j\}_{j=1}^N$ such that the i-th one is positive, while the remaining $(N-1)$ are negative, the probability of classifying the positive sample correctly is given by

$$p(d = i \mid X, c) = \frac{p(d = i, X \mid c)}{\sum_j p(d = j, X \mid c)} \tag{2.2.4}$$

$$= \frac{\prod_j p(d = i, x_j \mid c)}{\sum_j \prod_l p(d = j, x_l \mid c)} \tag{2.2.5}$$

$$= \frac{p(x_i \mid c) \prod_{j \neq i} p(x_j)}{\sum_j p(x_j \mid c) \prod_{l \neq j} p(x_l)} \tag{2.2.6}$$

$$= \frac{\frac{p(x_i|c)}{p(x_i)}}{\sum_{j=1}^N \frac{p(x_j|c)}{p(x_j)}} \tag{2.2.7}$$

Line 2.2.6 follows from x being independent of c when it is a negative sample, and line 2.2.7 follows from dividing the numerator and denominator by $\prod_i p(x_i)$. Now if we want a classifier to correctly identify the positive sample in $X = \{x_i\}$ given a context c, we would optimize the

[4] The *context* is all the information deemed necessary to identify the related patch correctly, which here is the collection of all patches up to the patch of interest, traversed in raster order.

categorical cross entropy of the classifier, assuming its output is $f(x, c)$, $\forall x \in X$, expressed as

$$\mathcal{L}_N = -\mathbb{E}\left[\log \frac{f(x, c)}{\sum_{x_j \in X} f(x_j, c)}\right]. \tag{2.2.8}$$

Since the optimal value of the loss is achieved when

$$\frac{f(x_i, c)}{\sum_{x_j \in X} f(x_j, c)} = p(d = i \mid X, c), \tag{2.2.9}$$

learning to classify the positive sample amounts to learning the density ratio with $f(x, c) \propto p(x \mid c)/p(x)$ (comparing Equation 2.2.9 and Equation 2.2.7). Note that this density ratio appears in the computation of *mutual information* between x and c,

$$I(x; c) = \int_x \int_c p(x, c) \log \frac{p(x \mid c)}{p(x)} dx dc. \tag{2.2.10}$$

f in Equation 2.2.8 is defined as the exponential of an inner product between the predicted code $\hat{z}$, as defined in Equation 2.2.3, and the true code z,

$$f(x, c) = \exp(\hat{z}^\top z). \tag{2.2.11}$$

Armed with all this, we can write out the *InfoNCE loss*, so-called since it is inspired by noise contrastive estimation (Gutmann and Hyvärinen, 2010) and the Infomax principle (Linsker, 1988).

$$\mathcal{L}_{\text{InfoNCE}} = -\sum_{ijk} \log \frac{\exp(\hat{z}_{i+k,j}^\top z_{i+k,j})}{\exp(\hat{z}_{i+k,j}^\top z_{i+k,j}) + \sum_l \exp(\hat{z}_{i+k,j}^\top z_l)}, \tag{2.2.12}$$

where i, j are spatial location indices, k is the number of rows we are looking ahead for predicting codes, and l indexes negative samples. The negative patches are taken from within other locations in the image, and from other images. One can recognize from the above formulation that the loss is straightforward to implement with a classifier which has a softmax layer at the output with the similarity terms as logits, and an objective that only requires correct identification of the positive sample.

2.2.2. Improved contrastive representation learning

Two follow-up methods for contrastive representation learning introduced useful mechanisms to improve performance further.

SimCLR. Chen et al. (2020) introduced two key ideas: one, they showed that introducing a learnable non-linear transform termed a *projection head* on the representation being learned is useful for the contrastive loss; and two, normalizing the projections to unit norm, followed

by scaling the inner product by a temperature also improves downstream classification. SimCLR is not a patch-based method, rather it aims to map two different augmentations of the same image to the same projection, and different images to different projections. The choice of augmentations plays an important role[5], since it prevents short-cuts for fulfilling the contrastive objective, instilling useful invariances in the feature extractor.

MoCo. He et al. (2020) introduced a dictionary-based perspective on self-supervised contrastive learning. If we view the similarity and dissimilarity between encodings as a dictionary lookup, we can imagine a memory bank of representations from which we aim to recall the value for a matching key. In a more practical instantiation, the method MoCo, short for *momentum contrast*, maintains a queue of examples that are transformed on-the-fly by a copy of the encoder using momentum-updated weights. A later improvement, MoCo-v2 (Chen et al., 2020), borrows ideas from SimCLR to further improve performance, in particular the ideas of using projection heads and stronger data augmentation schemes.

2.2.3. Non-contrastive self-supervised methods

While it certainly makes sense that contrastive learning as instantiated by the above methods might be expected to extract representations from images that are to do with the object-concepts in them, surprisingly, it turns out that we might not require a contrastive aspect at all – merely pulling together representations for similar data suffices. Naturally, this would not work naively, since a trivial solution would be to map everything to the same place. However, with certain architectural and training choices, it has been shown to be an effective representation learning strategy.

Bootstrap Your Own Latents (BYOL). Grill et al. (2020) show that it is possible to learn semantic representations by training an *online* network to yield features that are close to the features produced by a *target* network for the same input with different augmentations. The weights of the target network are a momentum-updated copy of the online network's parameters, which is reminiscent of MoCo. Crucially, a *stop-gradient* operation prevents any updating in the target network, which helps prevent trivial, constant solutions. The target network's momentum is annealed so that as training proceeds, the target network's weights are updated with smaller changes.

[5]the augmentations typically applied are random cropping+resizing, random color distortions, and random Gaussian blurring.

SimSiam. Chen and He (2021) go even further, showing that there is no need to perform a moving average in the target network; it can simply be a mirror of the online network (with the stop-gradient operation preventing collapsed solutions). SimSiam was shown to perform competitively or better than the above methods.

2.3. Out-of-distribution (OOD) settings

As we briefly discussed in the Introduction, it is often the case that models are trained on a dataset that does not capture some of the variations that are subsequently encountered at test-time. The models might also have learned to rely on a feature that only happened to correlate with the target due to bias in the training set. Such problems are broadly referred to in the contemporary literature as *OOD generalization* problems. Yet another aspect one must be aware of, in real life deployments, is the possibility of encountering unfamiliar objects. One can choose to filter out any inputs that are deemed to come from a distribution dissimilar to the training distribution, flagging them down for human intervention. This problem is broadly termed *OOD detection*. In the next two sections, we briefly describe these two problems.

2.3.1. Out-of-distribution detection

Machine learning models have been known to under-perform when they encounter test-time distributions that differ from the training distribution, raising concerns about *AI safety* (Amodei et al., 2016) when such models are deployed in the real world. With this motivation, Hendrycks and Gimpel (2017) proposed the task of OOD detection, since we could then potentially flag down any examples that could lead to potentially unsafe behavior.

There are several related precedents to the problem of OOD detection, developed with specific applications in mind. For example, *open-set recognition* (Scheirer et al., 2012; Bendale and Boult, 2016; Liu et al., 2018; Dhamija et al., 2018) considers classifiers that have been trained with K categories, but encounter categories outside of this set at test-time, and must not mis-identify them as a seen category. The terms *anomaly detection* (Chandola et al., 2009) or *novelty detection* (Pimentel et al., 2014) are often used interchangeably, and are typically applied to settings where the in-distribution data consists of one type of concept, although they have been more generally applied as well. For example, whereas open-set recognition tends to include multiple object categories in the *in-distribution*, an anomaly detection problem would more typically involve only one object category or type in the normal set. This

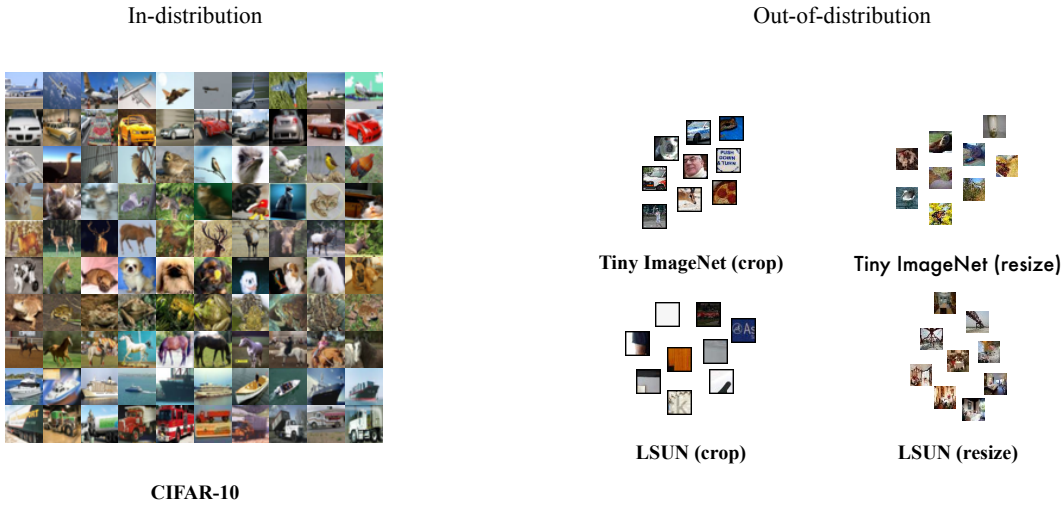


FIG. 4. Examples of typical benchmarks for OOD detection in the literature with CIFAR-10 as the in-distribution set.

is because the motivations in the literature for anomaly detection are more aligned towards that of recognizing deviations from a specific normal type, such as identifying manufacturing defects in samples of the same type of product on a factory production line (Bergmann et al., 2019), or intrusion detection in electronic security systems (Lazarevic et al., 2003). Such problems may be approached with techniques developed for *one-class classification* (Moya and Hush, 1996; Schölkopf et al., 2001). A related problem framework is that of *outlier detection* (Rousseeuw and Hubert, 2011) which presumes a given dataset (collected for some arbitrary downstream task) to contain deviant observations, termed *outliers*, that are to be identified and discarded before analysis or training on the data. The goal is different from anomaly detection in that one wishes to remove observations assumed to arise due to measurement errors or rare extremes in natural variability in order to promote robust estimation of parameters or statistics.

The benchmarks initially proposed for OOD detection in Hendrycks and Gimpel (2017) involve detecting examples from different datasets, i.e. if a model is trained on a particular dataset, say, CIFAR-10, then samples from a different dataset, for example the Scene Understanding Dataset, ought to be detected as OOD for the model (see Fig. 4 for examples). In this thesis, we adopt the perspective that in practical situations – be it evaluating reliability of classifiers, performing anomaly detection as a task, or detecting distributional shift – the deviations that concern us are context-driven. Within a specified context, such as object recognition, one is naturally interested in developing AI models that are *sensitive* to specific deviations and *robust* to others. This is especially true when dealing with high-dimensional data possessing low-dimensional meaning, as for object recognition in images. When building

trustworthy classifiers that can operate usefully under distributional shift, we would like our models to indicate reduced predictive confidence when encountering *semantic anomalies* but be robust to non-semantic distributional shift, where the semanticity is necessarily context-dependent. This suggests that benchmarking and evaluation must take into account the task of interest and the nature of the test dataset. Otherwise we might implicitly encourage the inadvertent development of models that perform worse at generalization to non-semantic distributional shift, i.e. are less robust. As an example, consider the popular method of *outlier exposure* (Hendrycks et al., 2019), which has been shown to improve OOD detection by training a classifier to assign a uniform predictive distribution to a different dataset from the training set. However, if the exposure-data differs mildly from the in-distribution training data in terms of, say, image brightness, then it is quite possible that we would end up with a classifier which is undesirably sensitive to brightness. We expand upon such perspectives in our first and second articles.

2.3.2. Out-of-distribution generalization

Around a decade ago, in 2011, Torralba and Efros (2011) suggested that there was "something rotten in the state" of the Vision datasets and benchmarking at the time, as manifested in the absence of any reporting of cross-dataset generalization results. Their reasoning was: if vision datasets and benchmarks are representatives of the real world, then two datasets are not really different *domains* in a broader sense. They are both equal representatives of the visual world, and a truly effective Vision algorithm trained on one dataset ought to generalize on another one which is related by task. Yet, they found that there were strong drops in performance when methods were evaluated in a cross-dataset fashion. The usual suspects behind such results are *selection bias* – datasets tend to reflect their collection modalities; *capture bias* – certain objects may be captured in a particular way in a specific dataset; *label bias* – where the labelling of objects are a function of the annotation process; and *negative set bias*, where the negative examples for a category in the training set heavily influence how a decision boundary is drawn by a discriminative learning algorithm.

In the years that followed, with the deep learning revolution occurring shortly thereafter, not much has changed with regards to evaluating the robustness of algorithms in a cross-dataset fashion. Leading benchmarks for fundamental tasks such as object recognition have largely continued to remain "worlds unto themselves". However, rising attention to the issue of generalization-failures on unfamiliar but similar test sets resulted in the development of

two families of approaches – *domain adaptation* and *domain generalization*. The two differ primarily in terms of assumptions on data availability at training-time.

Domain adaptation. In the problem of (unsupervised) domain adaptation, one assumes access to a labeled training set belonging to the *source* domain, and an upfront access to an unlabeled test set belonging to the *target* domain (Pan and Yang, 2009). The objective is to learn model parameters using the labels on the source domain, but in a manner that is likely to generalize to the target domain, by using the unlabeled samples only.

A popular bound in Ben-David et al. (2006) expresses the target domain risk, $\mathcal{R}_f(D_T)$, in terms of the empirical source domain risk, $\mathcal{R}_f(\hat{D}_S)$, and an estimate of discrepancy between the source and target domains. The discrepancy between two domains, measured by the so-called $\mathcal{H}$-divergence (Kifer et al., 2004) is given as

$$d_{\mathcal{H}}(D_S, D_T) = 2 \sup_{f \in \mathcal{H}} \left| \mathbb{P}_{x \sim D_S}\Big[f(x) = 1\Big] - \mathbb{P}_{x \sim D_T}\Big[f(x) = 1\Big] \right|, \tag{2.3.1}$$

implying that the difference between two domains can be estimated by the capacity of the hypothesis class $\mathcal{H}$ to tell apart source domain samples from target domain ones. Using this, the bound in Ben-David et al. (2006) is derived as follows.

$$\mathcal{R}_f(D_T) \leq \mathcal{R}_f(\hat{D}_S) + d_{\mathcal{H}}(\hat{D}_S, \hat{D}_T) + \beta + \lambda. \tag{2.3.2}$$

The term β subsumes other terms to do with sample-size, capacity of the hypothesis class $\mathcal{H}$, and probability of sampling the empirical data. λ is a term lower-bounded by the minimum sum of source and target domain risks.

This suggests one might improve domain adaptation by constructing feature spaces that make it harder for a classifier to tell the source domain from the target domain. This idea was developed in Ganin et al. (2016) for deep neural networks, showing that one can improve generalization to a target domain by backpropagating the error from a domain-discriminator but with a flipped-sign gradient. This effectively trains the feature extractor to learn invariant features, promoting generalization. Another alternative is to minimize the *maximum mean discrepancy* (MMD) (Gretton et al., 2012) between the features extracted from the source and target domains, as done in Tzeng et al. (2014). Other simpler alternatives have been developed, such as CORAL (Sun et al., 2016) which only matches second-order statistics across source and target domains; DEEP CORAL (Sun and Saenko, 2016) subsequently showed that this method was also effective when applied to the features extracted by a deep neural network. Simpler distribution-matching strategies can sometimes be more effective in practice, particularly in small-sample regimes, where one does not have sufficient information

to attempt a more precise match, and low-order moment-matching can be more stable and robust than adversarial approaches.

Matching marginal distributions between source and targets domains can lead to poor performance when the label distributions are significantly different. Most works implicitly assume that the label distribution $\mathbb{P}(y)$ stays similar across source and target domains, and that only the conditional distributions $\mathbb{P}(x \mid y)$ change. The problem of *target shift*, where the label distribution changes without any conditional shift in unsupervised domain adaptation has been approached by estimating label proportions across the source and target domains (Zhang et al., 2013; Li et al., 2019; Garg et al., 2020). The case where both the label as well as the conditional distribution changes is referred to as *generalized target shift* (GeTarS) and has been considered in Zhang et al. (2013); Gong et al. (2016); Tachet des Combes et al. (2020). Arguably the most realistic problem setting, GeTarS has received relatively little attention in the literature. In our third article, we consider a special instance of GeTarS, where the conditional shift is encountered in online deployment of a black-box AI model in different locations, each associated with a particular label-distribution.

Domain generalization. The primary difference between domain generalization and domain adaptation is that one does not presume access to an unlabeled test set up front. Instead, we assume access to multiple domains at training time, each with its unique characteristics, but such that the underlying task relies upon specific features that are invariant across all training domains (Blanchard et al., 2011). These invariant features are presumed to also exist in any new test domain, thus justifying the learning of a predictor that learns invariant features across all training domains. Although the problem settings are somewhat different, one can usually extend the approaches from the domain adaptation literature to the problems in domain generalization. For example, while in domain adaptation we were concerned with learning similar features across the source and target domains, now we can modify this objective to that of learning similar features across the multiple training domains.

Datasets in domain generalization benchmarks for Vision tasks have broadly fallen into two categories. In one type, the domains correspond to different datasets with the same objects represented in the same "style" – for example, the VLCS dataset collection (Fang et al., 2013) curates images from four different datasets of web-crawled images for five objects. In another type, the domains correspond to different stylistic representations. For example, in PACS (Li et al., 2017), Office-Home (Venkateswara et al., 2017), and DomainNet (Peng et al., 2019), the different domains in each benchmark consist of the same objects but captured in different image domains, for example, *art*, *sketches*, *cartoons*, or *real images*. These two types of benchmarks correspond to different real-world problems, and might call for different

approaches. When the difference lies only in the data collection scheme, the distributional-shift issues tend to be a combination of lower-level factors like the particular image processing methods used to downsample the images, as well as higher-level factors such as the selection and capture biases discussed previously. Depending on the application one is interested in, one might prefer one type or the other (or both) for evaluation. Apart from these, there are also some synthetic datasets with specific goals, such as the MNIST-R/S, ETH80P/Y datasets developed in Ghifary et al. (2015), for the purpose of testing generalization to novel poses. Another example of a synthetic dataset is IMAGENET-C (Hendrycks and Dietterich, 2019), where the goal is to learn to be robust to artificial corruptions added to IMAGENET images, for example, through blurring or noise-addition.

One can consider several other types of distributional shifts, aligned with real life circumstances. For example, one can encounter *sub-population shift*, where a particular subset in a heterogeneous dataset can change in relative frequency in the test-set (group shifts, as in Oren et al. (2019)) or undergo a change in "sub-type" – for example, in the training set dogs might have been represented by poodles, but at test-time, they could be German shepherds (Santurkar et al., 2020). Depending on the specific downstream application in mind, data from such sub-type shifts might either be treated as anomalies or not. One might argue that while a classifier trained on poodles should recognize German Shepherds as a form of dog (assuming there are no closer categories in the training set), this classifier ought to exhibit reduced confidence for this prediction, so that one can set a threshold for predictive confidence that allows for either anomaly detection or generalization, depending on intended use. *Distributionally robust optimization* (DRO) (Ben-Tal et al., 2013) has been shown to be useful at handling certain group-shifts, where an unstable feature-label correlation, existing in the larger subgroup in training data, does not manifest in a much smaller subgroup. At test-time, performance on samples from the smaller subgroup can be far worse relatively (Sagawa et al., 2020). DRO aims to reduce expected loss over worst-case distributions, by searching for worst-performing distributions in a ball around the training distribution, for example. However, this can lead to overly pessimistic models, optimizing for unrealistic distributional shifts (Duchi and Namkoong, 2018). *Group DRO* (Hu et al., 2018; Oren et al., 2019; Sagawa et al., 2020) instead approximately minimizes worst-group risk among training sub-groups, instantiated practically as a mixture over the sub-groups, such that the mixture weights are a function of training losses. Another alternative is to simply reweight losses over the groups in inverse proportion of their frequencies (King, 2014), if one were aware of the sub-groups of interest. We shall revisit these techniques in our second article.

Causality and invariance. Domain generalization can also be viewed through the lens of *causality*. Peters et al. (2016) discussed that a model's predictions can be expected to be more robust to potential interventions when the model makes predictions based on causal covariates. With the presumption that causal mechanisms are invariant across environments associated with different interventions (and an invariant noise variable, if any), one can expect the conditional distribution of the target variable given the immediate parent causes to remain invariant under interventions on other covariates. A predictor that represents such a conditional distribution is called an *invariant predictor*. While their discussion was for linear models, more recently, Arjovsky et al. (2019) extend such ideas to the deep learning context, proposing *invariant risk minimization* (IRM). If we consider a deep predictor to consist of two "stages" – a feature-extractor stage $f_\theta(x)$, followed by a linear predictor w – IRM specifies the objective of a deep feature extractor to be that of learning features that leads to a predictor which is simultaneously optimal across all training environments. This is equivalent to learning features $f_\theta(x)$ that correlate in a stable manner with the target variable across different environments (i.e. $\mathbb{E}[Y^e \mid f_\theta(x^e) = h] = \mathbb{E}[Y^{e'} \mid f_\theta(x^{e'}) = h]$ for any pair of data-environments e, e'), since capturing stable feature-target correlations corresponds to modelling the presumed invariant conditional distribution $\mathbb{P}(Y \mid f_\theta(x))$.

One might wonder what the advantage of IRM is over the methods discussed earlier for matching feature distributions across environments, especially if such matching is performed after conditioning upon labels. One possible advantage is that IRM can be more resistant to label-noise. For example, if we assume significantly high label-noise in our data, then matching features conditioned on these noisy labels can promote confusing feature representations. Since IRM seeks to match the invariant conditional output distribution, it can account for invariant label-noise as well. We ought to note that: (1) the presumption of invariant label-noise across environments can be unrealistic in a lot of real-life data collection schemes, i.e. label-noise can very much be a function of the environment; (2) current empirical evidence seems to suggest that existing instantiations of IRM under-perform at most domain generalization tasks (Gulrajani and Lopez-Paz, 2020; Koh et al., 2021). This failure could be due to multiple reasons, such as undesirable behavior in the non-asymptotic regime of training, or due to difficulties in optimization for the specific instantiations of the objectives (Zhang et al., 2022).

More generally, causal inference from purely observational data (i.e. a static dataset without any controlled intervention) is only achievable under several specific assumptions (Hernan and Robins, 2023). Two such key assumptions are *ignorability* (Rubin, 1978), which means there are no unmeasured confounders in the data, and *positivity* (Rosenbaum and Rubin, 1983), which implies that all possible interventions have a non-zero probability of being administered for all possible values of covariates. Note that it is not possible to test for ignorability, given

an observational dataset. While positivity can be estimated by analyzing the dataset, for a specified set of covariates and interventions, in high-dimensional feature-learning settings, one does not really have a clear sense of what the covariates are, in order to reliably perform such tests.

In this thesis, we are concerned with classification problems on static datasets (more particularly visual object recognition problems), while the discussions in the classical causal inference literature are framed within the nomenclature and structure of problems involving the estimation of average treatment effects of a drug. While analogies between the two experimental setups are not immediate, one can intuitively sense connections, which have been developed more formally in the literature (Chalupka et al., 2014; Schölkopf et al., 2012; Arjovsky et al., 2019; Ilse et al., 2021). In such settings, an intervention can correspond to the choice of environment to collect data from, which can involve curating images from a different location of origin or via different collection schemes, or simply performing data transformations. The assumption of ignorability would now correspond to the notion that there is no latent confounding variable in the data generating and collection process. The assumption of positivity implies that all possible environments have been sampled, for all object categories. One can imagine that such assumptions are less likely to be satisfied in a high-dimensional problem of the sort deep learning applications are concerned with, given the prevalence of the dataset biases discussed previously. Large-scale curation spanned across multiple diverse environments can potentially improve the likelihood of satisfying some of these assumptions.

Since we are concerned with static datasets in the deep learning context, with no guarantees of key assumptions being met or being testable, we do not discuss causal inference further in our articles, although recovering a true and complete causal model might be expected to provide optimal behavior under distributional shift (Schölkopf et al., 2012). Our perspective is rather that one can make useful predictions in many real-life applications based on robust observational associations, without necessarily communicating a causal interpretation. We shall interpret "robust associations" to imply that feature-label correlations are observed to be stable across environments of interest. Environments of interest may be inferred from existing datasets (Creager et al., 2020), or provided by meta-data (Koh et al., 2021), or artificially introduced through data transformations (Gulrajani and Lopez-Paz, 2020).

Hyper-parameter selection. The final aspect we will touch upon in this chapter is the question of hyper-parameter tuning (or model selection) when developing methods for OOD generalization. As mentioned in the introductory chapter, the standard recipe for IID settings is to hold out a fraction of training data to estimate test error upon. When we wish to

estimate OOD error, we have the problem of not quite knowing what the OOD data would look like. This problem seems somewhat ill-posed, but one sensible approach can be to collect an OOD validation set which we believe changes in ways from the training set that might be considered representative of similar, but likely different, changes at deployment. While one still cannot reliably estimate OOD test error, one can adopt a pragmatic perspective and assume that if, given data from a set of environments, average generalization improves on held-out environments (one at a time), then it is intuitively likely that corresponding improvements may be achieved on another yet-unseen environment. Evidently, none of this reasoning can come with any guarantees or formal development without placing strong assumptions on the available data distributions, and the ones likely to be encountered in the wild, which we have presumed to be unknown or difficult to anticipate in the problem setting.

Prologue to the first article

Detecting semantic anomalies. Faruk Ahmed and Aaron Courville. *Proceedings of the 34th AAAI Conference on Artificial Intelligence, 2020 (presented as a Spotlight Talk).*

Abstract. We critically appraise the recent interest in out-of-distribution (OOD) detection and question the practical relevance of existing benchmarks. While the currently prevalent trend is to consider different datasets as OOD, we argue that out-distributions of practical interest are ones where the distinction is semantic in nature for a specified context, and that evaluative tasks should reflect this more closely. Assuming a context of object recognition, we recommend a set of benchmarks, motivated by practical applications. We make progress on these benchmarks by exploring a multi-task learning based approach, showing that auxiliary objectives for improved semantic awareness result in improved semantic anomaly detection, with accompanying generalization benefits.

Key Words: *out-of-distribution detection, systematic generalization, anomaly detection.*

Context. At ICLR 2017, Hendrycks and Gimpel (2017) introduced the task of OOD detection, along with a set of benchmarks which were (and to an extent still are) widely adopted by the community. The motivation behind the task was that being able to detect when inputs to an AI model come from a distribution different to the training distribution would allow us to screen such inputs which could potentially lead to unpredictable predictive behavior. The proposed benchmarks involved a selection of different datasets for every in-distribution training set; for example, one typical test-suite involved CIFAR-10 as the in-distribution set while the OOD sets were {Scene Understanding Dataset (SUN), Gaussian noise}. This style of dataset-shift detection rapidly became widely adopted as standard, receiving some augmentations of the OOD set in later works such as Liang et al. (2018), which added Tiny-Imagenet to the set.

Article contributions. We suggested that one ought to take into account the different ways distributions might be different. In particular, distributional shift might be semantic in

nature or non-semantic. While semantic differences should be acknowledged, non-semantic shifts are typically ones we wish our models to be robust to, otherwise we start to move away from the desired ML goal of generalization. Non-semantic differences (such as low-level differences in images) are also typically easier to detect, perhaps the reason behind the flattering performances reported in the literature. Arguing for the relevance of detecting semantic shift, we recommended restricting the OOD suite to held-out categories, as in the open-set recognition framework. While the previous benchmarks did involve semantic differences, they also included overwhelming non-semantic differences, due to the use of differently curated datasets for different tasks. By way of supporting arguments, (1) we found existing high-performant methods to significantly under-perform on our benchmarks compared to dataset-shift detection, and (2) we showed that a trivial baseline consisting of a pixel-wise Gaussian likelihood model performed competitively with SOTA methods on dataset-shift detection for CIFAR-10. We also curated a set of small-scale, fine-grained semantic anomaly detection benchmarks based off of ImageNet. We showed how one might improve performance at these proposed benchmarks by using auxiliary self-supervised tasks during training, both at detecting semantic distributional shift as well as improving generalization. Source code is available at `https://github.com/Faruk-Ahmed/detecting_semantic_anomalies`.

Subsequent developments. Our critiques have motivated subsequent development of OOD detection benchmarks more aligned with detecting semantic distributional shift, along with adoption of our recommended evaluation settings. New benchmarks and evaluation protocols focused on semantic distributional shift citing our arguments appear in Hendrycks et al. (2021); Arora et al. (2021); Doorenbos et al. (2022); Yang et al.; Bhaskhar et al. (2022). Our benchmarks were used for evaluating methods in Sastry and Oore (2020); Deecke et al. (2020; 2021). Furthermore, while we had demonstrated that auxiliary self-supervised tasks can improve semantic anomaly detection by potentially improving semanticity of feature spaces, Deecke et al. (2021) show how performance can be improved significantly further if one adopts large-scale pre-training followed by training with parameter-drift penalties.

Author contributions. The contributions of the authors are the following.

- Aaron Courville had initially suggested using held-out categories to evaluate semantic understanding in generative models, which we had explored in my Masters thesis (Ahmed, 2018). Aaron supervised the project at all stages.
- I developed further context and motivation for such benchmarks with regards to building reliable classifiers, proposed the idea for multi-task training with self-supervised objectives, designed and implemented the experiments, and wrote the paper.

Chapter 3

Detecting semantic anomalies

3.1. Introduction

In recent years, concerns have been raised about modern neural network based classification systems providing incorrect predictions with high confidence (Guo et al., 2017). A possibly-related finding is that classification-trained CNNs find it much easier to "overfit" to low-level properties such as texture (Geirhos et al., 2019), canonical pose (Alcorn et al., 2019), or contextual cues (Beery et al., 2018) rather than learning globally coherent characteristics of objects. A subsequent worry is that such classifiers, trained on data sampled from a particular distribution, are likely to be misleading when encountering novel situations in deployment. For example, silent failure might occur due to equally confident categorization of unknown objects into known categories (due to shared texture, for example). This last concern is one of the primary motivating reasons for wanting to be able to detect when test data comes from a different distribution than that of the training data. This problem has been recently dubbed *out-of-distribution (OOD) detection* (Amodei et al., 2016; Hendrycks and Gimpel, 2017), but is also referred to as anomaly/novelty/outlier detection in the contemporary machine learning context. Evaluation is typically carried out with benchmarks of the style proposed in Hendrycks and Gimpel (2017), where different datasets are treated as OOD after training on a particular in-distribution dataset. This area of research has been steadily developing, with some additions of new OOD datasets to the evaluation setup (Liang et al., 2018), and improved results.

Current benchmarks are ill-motivated. Despite such tasks rapidly becoming the standard benchmark for OOD detection in the community, we suggest that, taken as a whole, they are not very well-motivated. For example, the object recognition dataset CIFAR-10 (consisting

of images of objects placed in the foreground), is typically trained and tested against noise, or different datasets such as downsampled LSUN (a dataset of scenes), or SVHN (a dataset of house numbers), or TINY-IMAGENET (a different dataset of objects). For the simpler cases of noise, or datasets with scenes or numbers, low-level image statistics are sufficient to tell them apart. While choices like TINY-IMAGENET might seem more reasonable, it has been noted that particular datasets have particular biases related to specific data collection and curation quirks (Torralba and Efros, 2011; Tommasi et al., 2017), which renders the problem of treating different datasets for OOD detection questionable. It is possible we are only getting better at distinguishing such idiosyncrasies. As an empirical illustration, we show in Appendix A.3 that very trivial baselines can perform reasonably well at existing benchmarks.

***Semantic* distributional shift is relevant.** We call into question the practical relevance of these evaluative tasks which are currently treated as standard by the community. While they might have some value as very preliminary reliability certification or as a testbed for diagnosing peculiar pathologies (for example, undesired behaviours of unsupervised density models, as in Nalisnick et al. (2019)), their significance as benchmarks for practical OOD detection is less clear. The implicit goal for the current style of benchmarks is that of detecting one or more of a wide variety of distributional shifts, which mostly consist of irrelevant factors when high-dimensional data has low-dimensional semantics. We argue that this is misguided; in a realistic setting, distributional shift across non-semantic factors (for example, camera and image-compression artefacts) is something we want to be robust to, while shift in semantic factors (for example, object identity) should be flagged down as anomalous or novel. Therefore, OOD detection is well-motivated only when the distributional shift is semantic in nature.

Context determines semantic factors. In practical settings, OOD detection becomes meaningful only after acknowledging context, which identifies relevant semantic factors of interest. These are the factors of variation whose unnatural deviation are of concern to us in our assumed context. For example, in the context of scene classification, a kitchen with a bed in the middle is an anomalous observation. However, in the context of object recognition, the primary semantic factor is not the composition of scene-components anymore, but the identity of the foreground object. Now the unusual context should not prevent correct object recognition. If we claim that our object recognition models should be less certain of identifying an object in a novel context, it amounts to saying that we would prefer our models to be biased. In fact, we would like our models to systematically generalize (Fodor and Pylyshyn, 1988) in order to be trustworthy and useful. We would like them to form predictions from a globally coherent assimilation of the relevant semantic factors for the task, while being robust to their composition with non-semantic factors.

Without context, OOD detection is too broad to be meaningful. The problem of OOD detection then, as currently treated by the community, suffers from imprecision due to context-free presumption and evaluation. Even though most works assume an underlying classification task, the benchmark OOD datasets include significant variation over non-semantic factors. OOD detection with density models are typically presented as being unaware of a downstream module, but we argue that such a context must be specified in order to determine what shifts are of concern to us, since we typically do not care about all possible variations. Being agnostic of context when discussing OOD detection leads to a corresponding lack of clarity about the implications of underlying methodologies in proposed approaches. The current benchmarks and methods therefore carry a risk of potential misalignment between evaluative performance and field performance in practical OOD detection problems. Henceforth, we shall refer to such realistic OOD detection problems, where the concerned distributional shift is a semantic variation for a specified context, by the term *anomaly detection*.

Contributions and overview. Our contributions in this paper are summarized as follows.

1. *Semantic shifts are interesting, and benchmarks should reflect this more closely:* We provided a grounded discussion about the relevance of semanticity in the context of a task for realistic OOD (anomaly) detection. Under the view of regarding distributional shifts as being either semantic or non-semantic for a specified context, we concluded that semantic shifts are of practical interest. If we want to deploy reliable models in the real world, we typically wish to achieve robustness against non-semantic shift.

2. *More practical benchmarks for anomaly detection:* Although our discussion applies generally, in this paper we assume the common context of object recognition. In this context, unseen object categories may be considered anomalous at the "highest level" of semanticity. Anomalies corresponding to intermediate levels of semantic decomposition can also be relevant; for example, a liger should result in 50-50 uncertainties if the training data contains only lions and tigers. However, such anomalies are significantly harder to curate, requiring careful interventions at collection-time. Since detection of novel categories is a compelling anomaly detection task in itself, we recommend benchmarks that reflect such applications in section 2.

3. *Auxiliary objectives for improved semantic representation improves anomaly detection:* Following our discussion about the relevance of semanticity, in sections 4 and 5 we investigate the effectiveness of multi-task learning with auxiliary self-supervised objectives. These have been shown to result in semantic representations, measured through linear separability by object categories. Our experimental results are indicative that such augmented objectives lead to improved anomaly detection, with accompanying improvements in generalization.

TABLE 1. Sizes of proposed benchmark subsets from ILSVRC2012. The training set consists of roughly 1300 images per member, and 50 images per member in the test set (which come from the validation set images in the ILSVRC2012 dataset).

Subset	Number of members	Total training images	Total test images
Dog (*hound dog*)	12	14864	600
Car	10	13000	500
Snake (*colubrid snake*)	9	11700	450
Spider	6	7800	300
Fungus	6	7800	300

3.2. Motivation and proposed tasks

In order to develop meaningful benchmarks, we begin by considering some practical applications where being able to detect anomalies, in the context of classification tasks, would find use.

Nature studies and monitoring: Biodiversity scientists want to keep track of variety and statistics of species across the world. Online tools such as *iNaturalist* [1] enable photo-based classification and subsequent cataloguing in data repositories from pictures uploaded by naturalists. In such automated detection tools, a potentially novel species should result in a request for expert help rather than misclassification into a known species, and detection of undiscovered species is in fact a task of interest. A similar practical application is camera-trap monitoring of members in an ecosystem, notifying caretakers upon detection of invasive species (Fedor et al., 2009; Willi et al., 2019). Taxonomy of collected specimens is often backlogged due to the human labour involved. Automating digitization and identification can help catch up, and often new species are brought to light through the process (Carranza-Rojas et al., 2017), which obviously depends on effective detection of novel specimens.

Medical diagnosis and clinical microbiology: Online medical diagnosis tools such as *Chester* (Cohen et al., 2019) can be impactful at improving healthcare levels worldwide. Such tools should be especially adept at being able to know when faced with a novel pathology rather than categorizing into a known subtype. Similar desiderata applies to being able to quickly detect new strains of pathogens when using machine learning systems to automate clinical identification in the microbiology lab (Zieliński et al., 2017).

AI safety: Amodei et al. (2016) discuss the problem of distributional shift in the context of autonomous agents operating in our midst, with examples of actions that do not translate well

[1] https://www.inaturalist.org

across domains. A similar example in that vein, grounded in a computer vision classification task, is the contrived scenario of encountering a novel vehicle (that follows different dynamics of motion), which might lead to a dangerous decision by a self-driving car which fails to recognize unfamiliarity.

Having compiled the examples above, we can now try to come up with an evaluative setting more aligned with realistic applications. The basic assumptions we make about possible evaluative tasks are: (i) that anomalies of practical interest are semantic in nature; (ii) that they are relatively rare events whose detection is of more primary relevance than minimizing false positives; and (iii) that we do not have access to examples of anomalies. These assumptions guide our choice of benchmarks and evaluation.

Recommended benchmarks A very small number of recent works (Akcay et al., 2018; Zenati et al., 2018) have considered a case that is more aligned with the goals stated above. Namely, for a choice of dataset, for example MNIST, train as many versions of classifiers as there are classes, holding out one class every time. At evaluation time, score the ability of being able to detect the held out class as anomalous. This is a setup more clearly related to the task of being able to detect semantic anomalies, holding dataset-bias factors invariant to a significantly greater extent. In this paper, we shall explore this setting with CIFAR-10 and STL-10, and recommend this as the default benchmark for evaluating anomaly detection in the context of object recognition. Similar setups apply to different contexts. We discourage the recently-adopted practice of treating one category as in-distribution and many other categories as out-distributions (as in Pidhorskyi et al. (2018) and Golan and El-Yaniv (2018), for example). While this setting is not aligned with the context of multi-object classification, it relies on a dataset constructed for such a purpose. Moreover, practical situations calling for one-class modelling typically consider anomalies of interest to be (often subtle) variations of the same object, and not a set of very distinct categories.

While the hold-out-class setting for CIFAR-10 and STL-10 is a good setup for testing anomaly detection of disparate objects, a lot of applications, including some of the ones we described earlier, require detection of more fine-grained anomalies. For such situations, we propose a suite of tasks comprised of subsets of ILSVRC2012 (Russakovsky et al., 2015), with fine-grained subcategories. For example, the SPIDER subset consists of members *tarantula*, *Argiope aurantia*, *barn spider*, *black widow*, *garden spider*, *wolf spider*. We also propose FUNGUS, DOG, SNAKE, and CAR subsets. These subsets have varied sizes, with some of them being fairly small (see table 1). Although this is a significantly harder task, we believe this setting aligns with the practical situations we described above, where sometimes large

quantities of labelled data are not always available, and a particular fine-grained selection of categories is of interest. See Appendix A.1 for more details about our construction.

Evaluation Current works tend to mainly use both Area under the Receiver-Operator-Characteristics (AUROC) and Area under Precision-Recall curve (AUPRC) to evaluate performance on anomaly detection. In situations where positive examples are not only much rarer, but also of primary interest for detection, AUROC scores are a poor reflection of detection performance; *precision* is more relevant than the false positive rate (Fawcett, 2006; Davis and Goadrich, 2006; Avati et al., 2018). We shall not inspect AUROC scores because in all of our settings, normal examples significantly outnumber anomalous examples, and AUROC scores are insensitive to skew, thus resulting in optimistic scores (Davis and Goadrich, 2006). Precision and recall are calculated as

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}, \tag{3.2.1}$$

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}, \tag{3.2.2}$$

and a precision-recall curve is then defined as a set of precision-recall points, for a varying threshold, t,

$$\text{PR curve} \triangleq \{\text{recall}(t), \text{precision}(t), -\infty < t < \infty\}. \tag{3.2.3}$$

The area under the precision-recall curve is calculated by varying the threshold t over a range spanning the data, and creating a finite set of points for the PR curve. One alternative is to interpolate these points, producing a continuous curve as an approximation to the true curve, and computing the area under the interpolation by, for example, the trapezoid rule. Interpolation in a precision-recall curve can sometimes be misleading, as studied in Boyd et al. (2013), who recommend a number of more robust estimators. Here we use the standard approximation to average precision as the weighted mean of precisions at thresholds, weighted by the increase in recall from the previous threshold.

$$\text{average precision} = \sum_k \text{precision}_k(\text{recall}_k - \text{recall}_{k-1}). \tag{3.2.4}$$

3.3. Related work

Evaluative tasks As discussed earlier, the style of benchmarks widely adopted today follows the recommendation in Hendrycks and Gimpel (2017). Among follow-ups, the most significant successor has been Liang et al. (2018) which augmented the suite of tests with slightly more

reasonable choices: for example, TINY-IMAGENET is considered as out-of-distribution for in-distribution datasets such as CIFAR-10. However, on closer inspection, we find that TINY-IMAGENET shares semantic categories with CIFAR-10, such as species of {dogs, cats, frogs, birds}, so it is unclear how such choices of evaluative tasks correspond to realistic anomaly detection problems. Work in the area of *open-set recognition* is closer to a realistic setup in terms of evaluation; in Bendale and Boult (2016), detection of novel categories is tested with a set of images corresponding to different classes that were discontinued in subsequent versions of Imagenet, but later work (Dhamija et al., 2018) relapsed into treating very different datasets as novel. We do not encourage using one particular split of a collection of unseen classes as anomalous. This is because such a one-time split might favour implicit biases in the predefined split, and the chances of this happening is reduced with multiple hold-out trials. As mentioned earlier, a small number of works have already used the hold-out-class style of tasks for evaluation. Unfortunately, due to a lack of a motivating discussion, the community at large continues to adopt the tasks in Hendrycks and Gimpel (2017) and Liang et al. (2018).

Approaches to OOD detection In Hendrycks and Gimpel (2017), the most natural baseline for a trained classifier is presented, where the detection score is simply given by the predictive confidence of the classifier (MSP). Follow-up work in Liang et al. (2018) proposed adding a small amount of adversarial perturbation, followed by temperature scaling of the softmax (ODIN). Methodologically, the approach suffers from having to pick a temperature and perturbation weight per anomaly-dataset. Complementary methods such as confidence calibration of DeVries and Taylor (2018), have been shown to improve performance of MSP and ODIN.

Using auxiliary datasets as surrogate anomalies has been shown to improve performance on existing benchmarks in Hendrycks et al. (2019). This approach is limited, due to its reliance on other datasets, but a more practical variant in Lee et al. (2018) uses a GAN to generate negative samples. However, Lee et al. (2018) suffers from the methodological issue of hyperparameters being optimized per anomaly-dataset. We believe that such contentious practices arise from a lack of a clear discussion of the nature of the tasks we should be concerned with, and a lack of grounding in practical applications which would dictate proper methodology. The primary goal of our paper is to help fill this gap.

Shalev et al. (2018) augment the training set with semantically similar labels, but it is not always practical to assume access to a corpora providing such labels. In the next part of the paper, we explore a way to potentially induce more semantic representation, with the

TABLE 2. Multi-task augmentation with the self-supervised objective of predicting rotation improves generalization.

	CIFAR-10	STL-10
Classification only	95.87 ± 0.05	85.51 ± 0.17
Classification+rotation	96.54 ± 0.08	88.98 ± 0.30

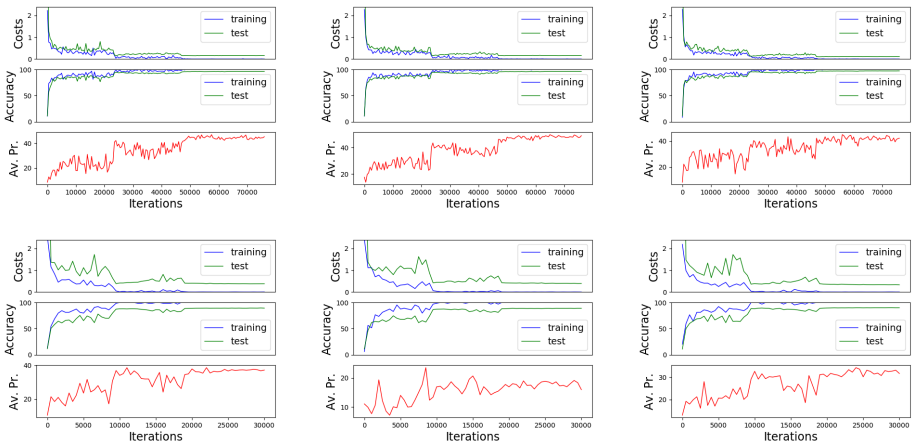


FIG. 1. Plots of costs, accuracies, and average precision for hold-out-class experiments with 3 categories each from CIFAR-10 (top) and STL-10 (bottom), using the MSP method (Hendrycks and Gimpel, 2017). While classification performance is not correlated with performance at anomaly detection (compare test accuracy numbers with average precision scores), the "pattern" of improvement at anomaly detection appears roughly related to generalization (compare the coarse shape of test accuracy curves with that of average precision curves).

hope that this would lead to corresponding improvements in semantic anomaly detection and generalization.

3.4. Encouraging semantic representations with auxiliary self-supervised objectives

We hypothesize that classifiers that learn representations which are more oriented toward capturing semantic properties would naturally lead to better performance at detecting semantic anomalies. "Overfitting" to low-level features such as colour or texture without consideration of global coherence might result in potential confusions in situations where the training data is biased and not representative. For a lot of existing datasets, it is quite possible to achieve good generalization performance without learning semantic distinctions, a possibility that spurs the search for removing algorithmic bias (Zemel et al., 2013), and which

is often exposed in embarrassing ways. As a contrived example, if the training and testing data consists of only one kind of animal which is furry, the classifier only needs to learn about fur-texture, and can ignore other meaningful characteristics such as the shape. Such a system would fail to recognize another furry, but differently shaped creature as novel, while achieving good test performance. Motivated by this line of thinking, we ask the question of how we might encourage classifiers to learn more meaningful representations.

Multi-task learning with auxiliary objectives. Caruana (1993) describes how sharing parameters for learning multiple tasks, which are related in the sense of requiring similar features, can be a powerful tool for inducing domain-specific inductive biases in a learner. Hand-design of inductive biases requires complicated engineering, while using the training signal from a related task can be a much easier way to achieve similar goals. Even when related tasks are not explicitly available, it is often possible to construct one. We explore such a framework for augmenting object recognition classifiers with auxiliary tasks. Expressed in notation, given the primary loss function, ℓ_{primary}, which is the categorical cross-entropy loss in the case of classification, and the auxiliary loss $\ell_{\text{auxiliary}}$ corresponding to the auxiliary task, we aim to optimize the combined loss

$$\ell_{\text{combined}}(\theta; \mathcal{D}) = \ell_{\text{primary}}(\theta; \mathcal{D}) + \lambda \ell_{\text{auxiliary}}(\theta; \mathcal{D}), \tag{3.4.1}$$

where θ are the shared parameters across both tasks, $\mathcal{D}$ is the dataset, λ is a hyper-parameter we learn by optimizing for classification accuracy on the validation set. In practice, we alternate between the two updates rather than taking one global step; this balances the *training rates* of the two tasks.

Auxiliary tasks. Recently, there has been strong interest in self-supervision applied to vision (Doersch et al., 2015; Pathak et al., 2016; Noroozi and Favaro, 2016; Zhang et al., 2017; van den Oord et al., 2018; Gidaris et al., 2018; Caron et al., 2018), exploring tasks that induce representations which are linearly separable by object categories. These objectives naturally lend themselves as auxiliary tasks for encouraging inductive biases towards semantic representations. First, we experiment with the recently introduced task in Gidaris et al. (2018), which asks the learner to predict the orientation of a rotated image. In table 2, we show significantly improved generalization performance of classifiers on CIFAR-10 and STL-10 when augmented with the auxiliary task of predicting rotation. Details of experimental settings, and performance on anomaly detection, are in the next section. We also perform experiments on anomaly detection with contrastive predictive coding (van den Oord et al., 2018) as the auxiliary task and find that similar trends continue to hold. The addition of such auxiliary objectives is complementary to the choice of scoring anomalies. Additionally, it enables further augmentation with more auxiliary tasks (Doersch and Zisserman, 2017).

TABLE 3. We train ResNet classifiers on CIFAR-10 holding out each class per run, and score detection with average precision for the maximum softmax probability (MSP) baseline in (Hendrycks and Gimpel, 2017) and ODIN (Liang et al., 2018). We find that augmenting with rotation results in improved anomaly detection as well as generalization (contrast columns in the right half with the left).

CIFAR-10	Classification-only			Rotation-augmented		
Anomaly	MSP	ODIN	Accuracy	MSP	ODIN	Accuracy
airplane	43.30 ± 1.13	48.23 ± 1.90	96.00 ± 0.16	46.87 ± 2.10	49.75 ± 2.30	96.91 ± 0.02
automobile	14.13 ± 1.33	13.47 ± 1.50	95.78 ± 0.12	17.39 ± 1.26	17.35 ± 1.12	96.66 ± 0.03
bird	46.55 ± 1.27	50.59 ± 0.95	95.90 ± 0.17	51.49 ± 1.07	54.62 ± 1.10	96.79 ± 0.06
cat	38.06 ± 1.31	38.97 ± 1.43	97.05 ± 0.12	53.12 ± 0.92	55.80 ± 0.76	97.46 ± 0.07
deer	49.11 ± 0.53	53.03 ± 0.50	95.87 ± 0.12	50.35 ± 2.57	52.82 ± 2.96	96.76 ± 0.09
dog	25.39 ± 1.17	24.41 ± 1.05	96.64 ± 0.13	32.11 ± 0.82	32.46 ± 1.39	97.36 ± 0.06
frog	40.91 ± 0.81	42.21 ± 0.48	95.65 ± 0.09	52.39 ± 4.58	54.44 ± 5.80	96.51 ± 0.12
horse	36.18 ± 0.77	36.78 ± 0.82	95.64 ± 0.08	39.93 ± 2.30	39.65 ± 4.31	96.27 ± 0.07
ship	28.35 ± 0.81	30.61 ± 1.46	95.70 ± 0.15	29.36 ± 3.16	28.82 ± 4.63	96.66 ± 0.17
truck	27.17 ± 0.73	28.01 ± 1.06	96.04 ± 0.24	29.22 ± 2.87	29.93 ± 3.86	96.91 ± 0.12
Average	34.92 ± 0.41	36.63 ± 0.61	96.03 ± 0.00	40.22 ± 0.16	41.56 ± 0.15	96.83 ± 0.02

3.5. Evaluation

We study the two existing representative baselines of maximum softmax probability (MSP) (Hendrycks and Gimpel, 2017), and ODIN (Liang et al., 2018) on the proposed benchmarks. For ODIN, it is unclear how to choose the hyperparameters for temperature scaling and the weight for adversarial perturbation without assuming access to anomalous examples, an assumption we consider unrealistic in most practical settings. We fix $T = 1000, \epsilon =$ 5e-5 for all experiments, following the most common setting.

3.5.1. Experimental settings

Settings for CIFAR-10 and STL-10. Our base network for all CIFAR-10 experiments is a Wide ResNet (Zagoruyko and Komodakis, 2016) with 28 convolutional layers and a widening factor of 10 (WRN-28-10) with the recommended dropout rate of 0.3. Following Zagoruyko and Komodakis (2016), we train for 200 epochs, with an initial learning rate of 0.1 which is scaled down by 5 at the 60th, 120th, and 160th epochs, using stochastic gradient descent with Nesterov's momentum at 0.9. We train in parallel on 4 Pascal V100 GPUs with batches of size 128 on each. For STL-10, we use the same architecture but append an extra group of 4 residual blocks with the same layer widths as in the previous group. We use a widening factor

TABLE 4. Average precision scores for hold-out-class experiments with STL-10. We observe that the same trends in improvements hold as with the previous experiments on CIFAR-10.

STL-10	Classification-only			Rotation-augmented		
Anomaly	MSP	ODIN	Accuracy	MSP	ODIN	Accuracy
airplane	19.21 ± 1.05	23.46 ± 1.65	85.18 ± 0.20	22.21 ± 0.76	23.37 ± 1.71	89.24 ± 0.12
bird	29.05 ± 0.69	33.51 ± 0.36	85.91 ± 0.36	36.12 ± 2.08	40.08 ± 3.30	89.91 ± 0.29
car	14.52 ± 0.37	16.14 ± 0.83	84.32 ± 0.55	15.95 ± 2.20	16.87 ± 2.94	89.52 ± 0.44
cat	25.21 ± 0.93	27.92 ± 0.84	86.95 ± 0.36	29.34 ± 1.30	31.35 ± 1.88	90.89 ± 0.26
deer	24.29 ± 0.53	25.94 ± 0.49	85.34 ± 0.35	27.60 ± 2.22	29.71 ± 2.55	89.20 ± 0.17
dog	23.42 ± 0.60	23.44 ± 1.18	87.78 ± 0.45	26.78 ± 0.71	26.14 ± 0.62	91.37 ± 0.33
horse	21.31 ± 1.01	22.19 ± 0.75	85.52 ± 0.21	23.79 ± 1.46	23.59 ± 1.63	89.60 ± 0.11
monkey	23.67 ± 0.83	21.98 ± 0.91	86.66 ± 0.31	28.43 ± 1.67	28.32 ± 1.20	90.07 ± 0.23
ship	14.61 ± 0.12	13.78 ± 0.63	84.65 ± 0.21	16.79 ± 1.20	15.37 ± 1.22	89.33 ± 0.15
truck	15.43 ± 0.17	14.35 ± 0.12	85.34 ± 0.17	17.05 ± 0.50	16.59 ± 0.60	90.08 ± 0.38
Average	21.07 ± 0.25	22.27 ± 0.29	85.77 ± 0.13	24.41 ± 0.23	25.14 ± 0.45	89.92 ± 0.08

TABLE 5. Averaged average precisions for the proposed subsets of Imagenet, with rotation-prediction as the auxiliary task. Each row shows averaged performance across all members of the subset. A random detector would score at the skew rate.

		Classification-only			Rotation-augmented		
Subset	Skew	MSP	ODIN	Accuracy	MSP	ODIN	Accuracy
dog	8.33	23.92 ± 0.49	25.85 ± 0.09	85.09 ± 0.14	24.66 ± 0.58	25.73 ± 0.87	85.25 ± 0.17
car	10.00	21.54 ± 0.62	22.49 ± 0.54	77.17 ± 0.10	21.66 ± 0.19	22.38 ± 0.46	76.72 ± 0.19
snake	11.11	18.62 ± 0.93	19.18 ± 0.79	69.74 ± 1.63	20.23 ± 0.18	21.17 ± 0.12	70.51 ± 0.48
spider	16.67	21.20 ± 0.56	24.15 ± 0.72	68.40 ± 0.21	22.90 ± 1.29	25.10 ± 1.78	68.68 ± 0.77
fungus	16.67	42.56 ± 0.49	44.59 ± 1.46	88.23 ± 0.45	44.19 ± 1.86	46.86 ± 1.13	88.47 ± 0.43

of 4 instead of 10, and batches of size 64 on each of the 4 GPUs, and train for twice as long. We use the same optimizer settings as with CIFAR-10. In both cases, we apply standard data augmentation of random crops (after padding) and random horizontal reflections.

Settings for Imagenet. For experiments with the proposed subsets of IMAGENET, we replicate the architecture we use for STL-10, but add a downsampling average pooling layer after the first convolution on the images. We do not use dropout, and use a batch size of 64, train for 200 epochs; otherwise all other details follow the settings for STL-10. The standard data augmentation steps of random crops to a size of 224×224 and random horizontal reflections are applied.

Predicting rotation as an auxiliary task. For adding rotation-prediction as an auxiliary task, all we do is append an extra linear layer alongside the one that is responsible for object

TABLE 6. Averaged average precisions for the proposed subsets of Imagenet where CPC is the auxiliary task.

		Classification-only			CPC-augmented		
Subset	Skew	MSP	ODIN	Accuracy	MSP	ODIN	Accuracy
dog	8.33	20.84 ± 0.50	22.77 ± 0.74	83.12 ± 0.26	21.43 ± 0.63	24.08 ± 0.63	84.16 ± 0.07
car	10.00	19.86 ± 0.21	21.42 ± 0.48	75.42 ± 0.11	22.21 ± 0.44	23.61 ± 0.57	78.88 ± 0.15
snake	11.11	18.20 ± 0.76	18.67 ± 1.07	66.15 ± 1.89	18.78 ± 0.40	20.39 ± 0.60	68.02 ± 0.85
spider	16.67	22.03 ± 0.68	24.08 ± 0.70	66.65 ± 0.42	22.28 ± 0.60	23.37 ± 0.68	68.67 ± 0.36
fungus	16.67	39.19 ± 1.26	41.71 ± 1.94	87.05 ± 0.06	42.08 ± 0.57	45.05 ± 1.11	88.91 ± 0.46

recognition. λ is tuned to 0.5 for CIFAR-10, 1.0 for STL-10, and a mix of 0.5 and 1.0 for IMAGENET. The optimizer and regularizer settings are kept the same, with the learning rate decayed along with the learning rate for the classifier at the same scales.

We emphasize that this procedure is not equivalent to data augmentation, since we do not optimize the linear classification layer for rotated images. Only the rotation prediction linear layer gets updated for inputs corresponding to the rotation task, and only the linear classification layer gets updated for non-rotated, object-labelled images. Asking the classifier to be rotation-invariant would require the auxiliary task to develop a disjoint subset in the shared representation that is not rotation-invariant, so that it can succeed at predicting rotations. This encourages an internally split representation, thus diminishing the potential advantage we hope to achieve from a shared, mutually beneficial space.

CPC as an auxiliary task. We also experimented with contrastive predictive coding van den Oord et al. (2018) as an auxiliary task. Since this is a patch-based method, the input spaces are different across the two tasks: that of predicting encodings of patches in the image, and that of predicting object category from the entire image. We found that two tricks are very useful for fostering co-operation: (i) replacing the normalization layers with their conditional variants de Vries et al. (2017) (conditioning on the task at hand), and (ii) using symmetric-padding instead of zero-padding. Since CPC induces significant computational overhead, we resorted to a lighter-weight base network. While this comes at the cost of a drop in performance, we still find, in table 6, that similar patterns of improvements continue to hold. We provide further details in Appendix A.2.

TABLE 7. Improving test set performance might not help.

Method	Accuracy	Av. Prec. with MSP
Base model	96.03 ± 0.00	34.92 ± 0.41
Random-center-masked	96.27 ± 0.05	34.41 ± 0.74
Rotation-augmented	96.83 ± 0.02	40.22 ± 0.16

3.5.2. Discussion

Self-supervised multi-task learning is effective. In tables 3 and 4 we report average precision scores on CIFAR-10 and STL-10 for the baseline scoring methods MSP (Hendrycks and Gimpel, 2017) and ODIN (Liang et al., 2018). We note that ODIN, with fixed hyperparameter settings across all experiments, continues to outperform MSP most of the time. When we augment our classifiers with the auxiliary rotation-prediction task, we find that anomaly detection as well as test set accuracy are markedly improved for both scoring methods. As we have remarked earlier, a representation space with greater semanticity should be expected to bring improvements on both fronts. All results report mean ± standard deviation over 3 trials. In table 5, we repeat the same process for the much harder Imagenet subsets. Taken together, our results indicate that multi-task learning with self-supervised auxiliary tasks can be an effective approach for improving anomaly detection, with accompanying improvements in generalization.

Improved test set accuracy is not enough. Training methods developed solely to improve generalization, without consideration of the affect on semantic understanding, might perform worse at detecting semantic anomalies. This is because it is often possible to pick up on low-level or contextual discriminatory patterns, which are almost surely biased in relatively small datasets for complex domains such as natural images, and perform reasonably well on the test set. To illustrate this, we run an experiment where we randomly mask out a 16×16 region in CIFAR-10 images from within the central 21×21 region. In table 7, we show that while this leads to improved test accuracies, anomaly detection suffers (numbers are averages across hold-out-class trials). This suggests that while the masking strategy is effective as a regularizer, it might come at the cost of less semantic representation. Certain choices can therefore result in models with seemingly improved generalization but which have poorer representation for tasks that require a more coherent understanding. For comparison, the rotation-augmented network achieves both a higher test set accuracy as well as an improved average precision. This example serves as a caution toward developing techniques that might achieve reassuring test set performance, while inadvertently following an internal *modus*

operandi that is misaligned with the pattern of reasoning we hope they discover. This can have unexpected consequences when such models are deployed in the real world.

3.6. Conclusion

We provided a critical review of the current interest in OOD detection, concluding that realistic applications involve detecting semantic distributional shift for a specified context, which we regard as anomaly detection. While there is significant recent interest in the area, current research suffers from questionable benchmarks and methodology. In light of these considerations, we suggested a set of benchmarks which are better aligned with realistic anomaly detection applications in the context of object classification systems.

We also explored the effectiveness of a multi-task learning framework with auxiliary objectives. Our results demonstrate improved anomaly detection along with improved generalization under such augmented objectives. This suggests that inductive biases induced through such auxiliary tasks could have an important role to play in developing more trustworthy neural networks.

We note that the ability to detect semantic anomalies also provides us with an indirect view of semanticity in the representations learned by our mostly opaque deep models.

Acknowledgements

We thank Rachel Rolland, Ishaan Gulrajani, Tim Cooijmans, and anonymous reviewers for useful discussions and feedback. This work was enabled by the computational resources provided by Compute Canada and funding support from the Canadian CIFAR AI chair and NSERC Discovery Grant.

Prologue to the second article

Systematic generalisation with group invariant predictions. Faruk Ahmed, Yoshua Bengio, Harm van Seijen, and Aaron Courville. *Proceedings of the 9th International Conference on Learning Representations, 2021 (presented as a Spotlight Talk).*

Abstract. We consider situations where the presence of dominant simpler correlations with the target variable in a training set can cause an SGD-trained neural network to be less reliant on more persistently correlating complex features. When the non-persistent, simpler correlations correspond to non-semantic background factors, a neural network trained on this data can exhibit dramatic failure upon encountering systematic distributional shift, where the correlating background features are recombined with different objects. We perform an empirical study on three synthetic datasets, showing that group invariance methods across inferred partitionings of the training set can lead to significant improvements at such test-time situations. We also suggest a simple invariance penalty, showing with experiments on our setups that it can perform better than alternatives. We find that even without assuming access to any systematically shifted validation sets, one can still find improvements over an ERM-trained reference model.

Key Words: *systematic generalization, out-of-distribution, invariance, semantic anomaly detection.*

Context. In the previous article, we suggested that under a compositional perspective of data with semantic and non-semantic factors, anomaly detection and systematic generalisation were two sides of the same coin. Specifically, while anomaly detection requires sensitivity to semantic distributional shift, generalization requires robustness to non-semantic distributional shift, and we would ideally aim for feature extraction to improve at both tasks simultaneously in order to build trustworthy classifiers. Building on this perspective, I had subsequently developed the notion of viewing non-semantic distributional shift as being compositional either in systematic or non-systematic ways – where systematic composition is a recombination of seen factors, and non-systematic composition is a combination of seen factors with unseen ones.

At the same time, Arjovsky et al. (2019) released their draft on Invariant Risk Minimization, sparking off a strong and renewed interest in OOD generalization within the ML community. The time was ripe for contributing to a conversation about all things OOD.

Article contributions. In order to perform controlled experiments, we created three synthetic setups where we could create four test sets for every training set – in-distribution, systematically-shifted, non-systematically-shifted, and semantically-shifted (corresponding to semantic anomaly detection). We used colors or scene backgrounds to bias a majority group of a training set with a high degree of background-label correlations. The key difference between the synthetic construction in Arjovsky et al. (2019) and ours was that we used no label-noise to amplify failure modes of ERM; rather, we excluded any counterfactuals to biases in the training set, under which condition ERM sees no reason to promote the unlearning of majority-group biases in later stages of training. We believe this to be a more plausible model of ERM-failure in real life datasets.

We also proposed a simple new invariance penalty based on matching average predictive distributions across majority and minority groups with a KL-divergence. We found this simple alternative to perform better than, or competitively with, existing invariance penalties on our synthetic testbeds.

Source code is available at https://github.com/Faruk-Ahmed/predictive_group_invariance.

Subsequent developments. Our synthetic datasets and the general framework of construction have been adopted in a number of recent papers, such as in Zhang et al. (2021); Zhou et al. (2022;); Xu and Jaakkola (2021); Shrestha et al. (2022); Saranrittichai et al. (2022). Citing our perspectives about controlling for semantic and non-semantic shift separately when assessing robustness to non-semantic shifts as well as sensitivity to semantic shift, Deecke et al. (2021) used a similar framework of synthetic experiments in the context of semantic anomaly detection.

Creager et al. (2021) pointed out that our KL penalty is closely related to the objective of equalized odds in the fairness literature (Hardt et al., 2016), a connection we were unaware of when writing the paper. Hu et al. (2022) also used our trick of freezing the last layer when applying an invariance penalty on the output space, in work about encouraging robustness to spurious correlations in a multi-task learning setting.

Author contributions. The contributions of the authors are the following.

- With the motivation of providing a compositional perspective, I conceptualized and created the novel experimental setup, developed the invariance penalty, performed the experiments, and wrote the paper.
- Yoshua Bengio had initially suggested exploring the question of recovering useful partitions of a dataset for invariant learning. Yoshua Bengio and Harm van Seijen participated in discussions and provided feedback on the draft.
- Aaron Courville supervised and provided feedback at all stages of the project.

Chapter 4

Systematic generalisation with group invariant predictions

4.1. Introduction

If a training set is biased such that an easier-to-learn feature correlates with the target variable throughout the training set, a modern neural network trained with SGD will use that factor to perform predictions, ignoring co-occurring harder-to-learn complex predictive features (Shah et al., 2020). Without any other criteria, this is arguably desirable behaviour, reflecting Occam's razor. We consider the situation where although such a simpler correlation is a dominant bias in the training set, a minority group exists within the dataset where the bias does not manifest. In such cases, relying on more complex predictive features which more pervasively explain the data can be preferable to simpler ones that only explain most of it. For example, if all chairs are red, redness ought to be a predictive rule for chairhood (without any other criteria for predictions). However, if some chairs are not red, and all chairs have backs and legs, then one can infer that redness is less relevant.

In this paper, we will study object recognition tasks, where the objects correlate strongly with simpler non-semantic background information for a majority of the images, but not for a minority group. There is evidence in the literature that modern CNNs tend to fixate on simpler features such as texture (Geirhos et al., 2019; Brendel and Bethge, 2019), canonical pose (Alcorn et al., 2019), or contextual background cues (Beery et al., 2018). We are assuming that semantic features in a classification context (ones that humans would agree contribute to their labelling of objects) are more likely to persistently correlate with the target variable, while simpler non-semantic background biases are more likely to exhibit

TABLE 1. For a coloured MNIST dataset with every digit correlated with a colour 80% of the time, we see poor performance at systematically varying tasks. Performance improves if the minority group combines colours from other biased digits - this provides corrective gradients that promote invariance to colour. Non-systematic shifts are when unseen colours are used, and anomaly detection is measured by decreased predictive confidence for an unseen digit (see Section 2 for more details).

Minority colours	In-distribution	Non-systematic shift	Systematic shift	Anomaly detection
Different	99.60 ± 0.02	53.26 ± 1.89	38.72 ± 2.27	7.70 ± 0.23
Recombinations	98.67 ± 0.39	85.05 ± 1.89	97.56 ± 0.05	46.59 ± 6.93

non-persistent correlations in real-life data collection processes. Based on this assumption, we will use combinations of objects and backgrounds to compare test-time performances corresponding to particular distributional shifts.

Consider coloured MNIST digits such that there is a dominant, but not universal, correlation between colour and digit identity for a majority of the images. In the situation we are considering, if the biasing colours in the majority group are not recombined with different digits in the minority group, then there is no signal for the model to disregard these biasing factors, which are retained as important predictive rules. This can lead to poor performance at *systematic generalisation* (Lake and Baroni, 2018), where an object occurs with another object's biasing factor, and at *semantic anomaly detection* (Ahmed and Courville, 2020), where a novel object appears with one of the biasing factors. In our example with coloured MNIST, if we colour the minority group digits with the colours used to bias (different) digits in the majority group, we find a marked improvement at systematically shifted tests over the case when the colours in the minority group are different colours altogether (see Table 1).

We investigate the role of encouraging robust predictive behaviour across such groups in terms of improved performance at tasks with such distributional shifts. Our experiments suggest that training with cross-group invariance penalties can result in models that have learned to be more reliant on persistent complex correlations without being overwhelmed by simpler, yet less stable features, as indicated by improved performance at systematic generalisation and semantic anomaly detection on our synthetic setups.

We find that a recently proposed method (Creager et al., 2020) can be effective at inferring the majority and minority groups along a learned feature-bias, and we use this inferred partition to provide us with groups in the training set in our comparative study. We also suggest a new method for encouraging predictions that rely on persistent correlations across

(A) T_r (B) T_g (C) T_s (D) T_n (E) T_a

FIG. 1. COLOURED MNIST training and test sets for evaluating generalisation under non-semantic marginal shift and systematic shift, and anomaly detection. (a) Training set; (b) *In-distribution generalisation set* T_g, where the test set is coloured following the same scheme as for T_r; (c) *Systematic-shift generalisation set* T_s, where we colour the test set with the biasing colours, but such that no digit is coloured with its own biasing colour; (d) *Non-systematic-shift generalisation set* T_n, where the test is coloured with random colours that are different from any of the colours seen in the training set; and (e) *Semantic anomaly detection set* T_a, where we colour the held-out digits of the test set randomly with the biasing colours.

such groups, with the intuition that similar predictive behaviour across the groups should be promoted throughout training. With experiments on three synthetic datasets, we compare the performance of recently proposed invariance penalties and methods, and find that our variant can often perform better at tasks involving such test-time distributional shifts.

4.2. Systematic and non-systematic generalisation

If we assume that data x is generated via a composition $\mathcal{C}$ of semantic factors h_s and non-semantic factors h_n, we can use this decomposition, $x = \mathcal{C}(h_s, h_n)$, to generate test datasets to capture different scenarios. While h_n is actually independent of y, we shall have the independence property $p_{\mathcal{D}}(h_n \mid y) = p_{\mathcal{D}}(h_n)$ to not hold when there is bias in the dataset $\mathcal{D}$ due to $h_n - y$ correlations.

We can evaluate, for a particular target y and our system's prediction of the target $\hat{y}(x)$, the average accuracy $\mathbb{E}\Big[\mathbf{1}\{\hat{y}(\mathcal{C}(h_s, h_n)) = y\}\Big]$, as a measure of generalisation for the following different cases.

In-distribution generalisation. $h_s \sim p(h_s \mid y)$ and $h_n \sim p(h_n \mid y)$: The validation and test sets are assumed to possess the same biases as the training set, in that the class-conditional distribution of non-semantic features in the test set match that of the training set, $p(h_n \mid y)$.

Generalisation under non-systematic-shift. $h_s \sim p(h_s \mid y)$ and $h_n \not\sim p(h_n)$[1]: This estimates a form of generalisation under distributional shift, where the non-semantic factors are sampled from outside the marginal distribution of h_n as present in the training set.

Generalisation under systematic-shift. $h_s \sim p(h_s \mid y)$ and $h_n \sim p(h_n \mid y')$ where $y' \sim p(y)$ $s.t.$ $y' \neq y$: This estimates another form of generalisation under distributional shift but one where non-semantic factors are sampled with intent to confuse: non-semantic factors for x are sampled from the marginal distribution of a randomly picked different target, $y' \neq y$. Although systematicity, as discussed in Fodor and Pylyshyn (1988), and systematic generalisation, as discussed in the NLP literature (Lake and Baroni, 2018; Bahdanau et al., 2019) consider recombinations of intra-semantic factors as well, here, in the context of background-agnostic object recognition tasks, we only consider $h_s - h_n$ recombinations.

Semantic anomaly detection. $h_s \not\sim p(h_s)$ and $h_n \sim p(h_n)$: Such a datapoint should not be confidently categorised as a known y, even if non-semantic features are shared (Ahmed and Courville, 2020). We can use these x to evaluate anomaly detection, as indicated by decreased predictive confidence, and measured by the area under the precision-recall curve (Hendrycks and Gimpel, 2017).

COLOURED MNIST: Consider an illustrative dataset with coloured MNIST digits. For the training set, T_r, MNIST digits are coloured with a set of digit-correlated "biasing" colours 80% of the time, and with ten random colours that are different from the biasing colours the remaining 20% of the time. One digit is held out, for testing semantic anomaly detection. See Figure 1 for examples of the four test sets corresponding to this setting, and also Appendix B.1 for more details on the construction.

Improving performance for such scenarios involving distributional shift might come at a cost for in-distribution performance, since more robust features might be harder to learn than simpler dominant correlations that hold in-distribution. In real-world deployments where one is likely to encounter unexpected situations, such as in a self-driving car, it can often be preferable to find appropriate trade-offs such that classifiers can indicate reduced confidence upon encountering anomalous objects, or continue to operate in changing environments, while continuing to achieve a desirable degree of in-distribution predictive performance.

[1]In this paper, we imply sampling from outside the support of p when we say $h \not\sim p(h)$.

4.3. Predictive group invariance across inferred splits

In general, we do not expect to have direct knowledge of majority and minority groups corresponding to the biasing non-semantic features in a dataset. We will later show how one might infer such groups from the data, but we first describe an invariance penalty assuming we have access to the groups.

Learning features that are group invariant would require us to match the (class-conditioned) distribution of features from the majority and minority groups (Ganin et al., 2016; Li et al., 2018). In terms of predictive performance, we can alternatively ask for the class-conditioned distributions of features to match in the sense that they lead to the same softmax distributions on average as training progresses, without modifying the last linear layer. This implementation has the advantage of doing away with an adversarial network, and the issues that tend to accompany the training of such models. We shall refer to this objective as *predictive group invariance* (PGI). Intuitively, encouraging matched predictive distributions across the groups with a fixed last layer pushes for over-emphasis on minority-group features in the representation, thus acting as an implicit re-weighting of features in both groups (leading to demoting the relevance of colour in the MNIST case, for example). When a persistent feature does exist in both groups, using that feature can lead to equal training rates in regularised networks, satisfying the penalty.

Consider a classifier that extracts a feature vector $f_\theta(x)$, where θ are the parameters of a convolutional neural network for example, with a linear layer w on top. The predictive distribution is then

$$p_w(y \mid x) = \sigma(w^\top f_\theta(x)), \tag{4.3.1}$$

where σ is a softmax, and predictions are made by performing an $\arg\max$.

Given a partition scheme for splitting the images x in our dataset $\mathcal{D}$ such that every i-th image $x^{(i)}$ is associated with a partition-label $\alpha^{(i)}$, we define distributions $\mathbb{P}^c, \mathbb{Q}^c$ for the subsets in class c:

$$x^{(i)} \sim \mathbb{P}^c \text{ if } \alpha^{(i)} = 0, y^{(i)} = c, \tag{4.3.2}$$

$$x^{(j)} \sim \mathbb{Q}^c \text{ if } \alpha^{(j)} = 1, y^{(j)} = c. \tag{4.3.3}$$

We want to minimize empirical risk under the constraint that our feature extractor causes similar predictive distributions on average for pictures of the same object in both partitions.

Formally, we want to optimise

$$\min_{\theta,w} \ell(\theta, w \mid \mathcal{D}), \tag{4.3.4}$$

$$\text{s.t. } \theta \in \arg\min_{\Theta} d\Big(\mathbb{E}_{x\sim\mathbb{P}^c}[p_w(y \mid x)], \mathbb{E}_{x\sim\mathbb{Q}^c}[p_w(y \mid x)]\Big), \ \forall\, c, \tag{4.3.5}$$

where ℓ is the standard loss function for ERM training, for example, the categorical cross-entropy. A softened objective for stochastic optimisation can be approximated as

$$L(w, \theta|\mathcal{D}, \alpha) = \ell(\theta, w \mid \mathcal{D}) + \lambda \left[\sum_c d\Big(\mathbb{E}_{x\sim\mathbb{P}^c}[p_{\tilde{w}}(y \mid x)], \mathbb{E}_{x\sim\mathbb{Q}^c}[p_{\tilde{w}}(y \mid x)]\Big) \right]_{\tilde{w}=w \text{ (fixed)}}. \tag{4.3.6}$$

Since we are comparing distributions, we make the simplest natural choice of d to be the KL-divergence,

$$d\Big(\mathbb{E}_{x\sim\mathbb{P}^c}[p_{\tilde{w}}(y \mid x)], \mathbb{E}_{x\sim\mathbb{Q}^c}[p_{\tilde{w}}(y \mid x)]\Big) = \sum \mathbb{E}_{x\sim\mathbb{Q}^c}[p_{\tilde{w}}(y \mid x)] \log \frac{\mathbb{E}_{x\sim\mathbb{Q}^c}[p_{\tilde{w}}(y \mid x)]}{\mathbb{E}_{x\sim\mathbb{P}^c}[p_{\tilde{w}}(y \mid x)]}. \tag{4.3.7}$$

We use this particular ordering of $\mathbb{Q}\|\mathbb{P}$ because with our grouping, $\mathbb{P}$ consists of examples that are "easy" due to a particular bias, and so the mean predictive distribution for $\mathbb{P}$ tends to be correct and low-entropy, while that for $\mathbb{Q}$ is more high-entropy and inaccurate. We take advantage of the zero-forcing property of this KL divergence, encouraging the mean predictive distribution for $\mathbb{Q}$ to closely match that of $\mathbb{P}$. It is likely that different choices for d would be better suited for different settings.

Partitioning the dataset Recently, Creager et al. (2020) have considered the question of finding worst-case partitions for invariant learning given a collection of data. The key intuition is that an invariant learning objective, as formulated by IRM (Arjovsky et al., 2019), is maximally violated by splitting along a spurious correlation when predictions rely exclusively on it in a reference model (see Theorem 1 in Creager et al. (2020) for details). In our case, this would consist of partitioning into the majority and minority groups given our ERM-trained model early on in training as reference.

A soft-partition predicting network is used, $g(x, y)$, conditioned on the input and the target, to *maximise* the IRMv1 penalty (Arjovsky et al., 2019), which gives us soft partition-predictions, $\hat{\beta}$, for the examples,

$$\begin{aligned}\hat{\beta} = \max_{\beta} & \sum_{e\in\{0,1\}} \frac{1}{\sum_{i'} \beta^{(i)}(e)} \sum_i \beta^{(i)}(e) \ell(\sigma(\Phi(x^{(i)})), y^{(i)}) \\ & + \sum_{e\in\{0,1\}} \gamma \left|\left| \nabla_{\mu|\mu=1.0} \frac{1}{\sum_{i'} \beta^{(i)}(e)} \sum_i \beta^{(i)}(e) \ell(\sigma(\mu \circ \Phi(x^{(i)})), y^{(i)}) \right|\right|^2,\end{aligned} \tag{4.3.8}$$

where $\Phi(x_i) = w^\top f_\theta(x)$ are the logits from the reference model, $e \in \{0, 1\}$ indexes the partition, $\beta^{(i)}(e) \in [0, 1]$ signifies the predicted probability for the i-th example being in partition e, such that $\beta^{(i)}(e = 0) + \beta^{(i)}(e = 1) = 1$, and γ is a hyper-parameter. We can then compute the partition $\alpha^{(i)} = \arg\max_e \beta^{(i)}(e)$. In our implementation, we condition the partition predicting network g on the features $f_\theta(x)$ instead of the input x, and use separate networks for each category, *i.e.* $\beta^{(i)} = g_{y^{(i)}}(f_\theta(x^{(i)}))$. We find this to perform better in preliminary experiments, improving training and enabling more light-weight g networks. This also ensures that the same features as the ones used by our ERM-trained reference model are used to predict partitions, resulting in partitions corresponding to more consistent learned-feature biases. We provide more details in Appendix B.2.3.

4.4. Related work

The dominant perspective towards the issue of unreliable behaviour in novel domains has consisted of treating the problem as that of *domain generalisation* (Blanchard et al., 2011). One hopes to recover stable features by encouraging invariance across data sampled from different domains, so that performance at test-time *out-of-distribution* (OoD) scenarios is less likely to be unstable.

Approaches along such lines typically resemble a cross-domain distribution-matching penalty applied to the features being learned, augmenting the usual ERM term (Ganin et al., 2016; Sun and Saenko, 2016; Heinze-Deml and Meinshausen, 2017; Li et al., 2018; Li et al., 2018;), and evaluated on datasets that consist of data in different modalities (Li et al., 2017; Peng et al., 2019; Venkateswara et al., 2017), or collected through different means (Fang et al., 2013), or in different contexts (Beery et al., 2018).

Works with the perspective of *distributionally robust optimisation* (DRO) have generally considered using uncertainty sets around training data (Ben-Tal et al., 2013; Duchi and Namkoong, 2018) to minimise worst-case losses, which can often have a regularising effect by effectively up-weighting harder examples. More relevant to our discussion, group DRO methods have considered uncertainty sets in terms of different groups of data, for example with different cross-group distributions of labels (Hu et al., 2018), or groups collected differently (Oren et al., 2019), similarly to domain generalisation datasets.

More recently, methods promoting the learning of stable features across data from different *environments*, or sources, have been proposed by using gradient penalties (Arjovsky et al.,

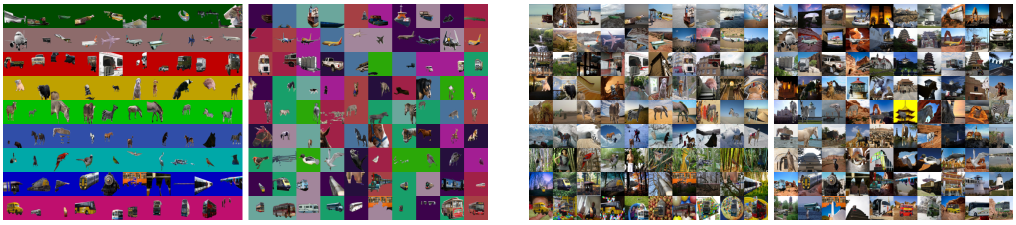


FIG. 2. (left) COCO-ON-COLOURS; left block is the majority group, right block is the "unbiased" minority group; (right) COCO-ON-PLACES.

2019), risk-based extrapolation (Krueger et al., 2020), and masking gradients with opposing signs (Parascandolo et al., 2020).

The typical datasets in such existing works are not curated with testing performance under systematic distributional shift in mind, most often not characterising the specific shift in distribution. In recent times, a commonly adopted synthetic dataset is the coloured MNIST variant used in Arjovsky et al. (2019) – since this particular dataset uses flipped colours for the minority group, which is less of a problem with ERM-training, the true digit labels were flipped at a sufficiently high frequency to incapacitate ERM performance by forcing reliance on colour. We believe setups such as ours can be better synthetic testbeds for developing ideas, where it is not necessary to alter ground truth labels to expose a failure mode. In general, using better models of dataset bias implies a narrower disconnect with realistic settings, with higher chances of the conclusions carrying over.

4.5. Experiments

We compare performance with our four test sets - in-distribution, non-systematically shifted, systematically shifted, semantic anomalies - for a range of recently proposed methods for a set of three synthesised datasets. Appendix B.2 describes architectural details and training choices.

4.5.1. Methods

We compare recent methods aimed at robust predictions across groups, and which do not require changes to network capacity or additional adversaries to impose invariance penalties. We also do not include methods based on advances in self-supervised feature learning, such

as Carlucci et al. (2019), since such methods are developed with prior knowledge of the desired invariances, and are thus limited in their generality.

Baseline: This is our reference model, trained via ordinary (regularised) empirical risk minimisation (ERM) without any invariance penalties added. The choices for architecture and regularisers were made to conform to the way modern networks are typically trained with in-distribution performance in mind (details in Appendix B.2).

IRMv1, **REx**, **GroupDRO**: IRMv1 (Arjovsky et al., 2019) and REx (Krueger et al., 2020) are two methods that augment the standard ERM term with invariance penalties across data from different sources. GroupDRO (Sagawa et al., 2020) is an algorithm for distributional robustness, which works by weighting groups of data as a function of their relative losses. See Appendix B.3 for more details about these methods.

cIRMv1, **cREx**, **cGroupDRO**: We implement label-conditional variants of the above algorithms, which, to our knowledge, has not been explored. In the context of multi-class classification it is reasonable to expect that performances might have multi-modal distributions along different categories earlier in training, which suggests stratification by class might improve performance.

Reweight: We weight the losses in the biased group down. This is a heuristic form of re-balancing the dataset, while choosing a hyper-parameter for the weight using the validation set, with the weight serving to downweight the losses for the biased group. In preliminary experiments we found this re-weighting variant (King and Zeng, 2001) to significantly outperform oversampling the minority group, as suggested in Buda et al. (2018), or weighting the grouped losses using their population ratios, as performed for imbalanced classes in Cui et al. (2019).

cMMD: Following Li et al. (2018), we match the MMD (Gretton et al., 2012) of the distribution of features. In preliminary experiments, we find a conditional version (as done with adversarial models in (Li et al., 2018)) to perform significantly better, so we only report cMMD results here.

TABLE 2. Generalisation results on COLOURED MNIST.

Methods	In-distribution	Non-systematic shift	Systematic shift	Anomaly detection
Base (ERM)	**99.60 ± 0.02**	53.26 ± 1.89	38.72 ± 2.27	7.70 ± 0.23
IRMv1	99.47 ± 0.05	63.24 ± 3.04	55.19 ± 1.07	11.54 ± 1.18
REx	98.95 ± 0.11	72.12 ± 1.90	71.18 ± 3.27	15.54 ± 2.05
GroupDRO	89.47 ± 4.52	70.53 ± 1.79	79.17 ± 1.64	35.15 ± 10.83
Reweight	98.51 ± 0.12	75.01 ± 1.28	84.85 ± 0.61	28.60 ± 1.11
cIRMv1	99.36 ± 0.25	65.78 ± 3.53	61.09 ± 5.30	14.16 ± 2.12
cREx	98.56 ± 0.12	74.35 ± 2.09	80.01 ± 2.11	22.02 ± 2.52
cGroupDRO	95.65 ± 3.23	75.41 ± 3.45	81.14 ± 2.41	26.61 ± 6.61
cMMD	99.40 ± 0.03	97.17 ± 0.59	97.86 ± 0.16	78.32 ± 4.15
PGI	99.05 ± 0.08	**98.58 ± 0.06**	**98.48 ± 0.05**	**89.42 ± 1.95**

4.5.2. Datasets

Evaluating performance in an unambiguous manner for the specific kinds of generalisation that we aim to study necessitates controlled test-beds. In order to model these tasks, we use 3 synthetic datasets of progressively higher complexity, approaching photo-realism.

COLOURED MNIST: This is the simplest setting, where the background information exists as part of the object.

COCO-ON-COLOURS: We superimpose 10 segmented COCO (Lin et al., 2014) objects on coloured backgrounds. The training set has 800 images per category, with nine in-distribution categories and one held-out category for anomaly detection. Validation and test sets have 100 each images per category. See Figure 2 (left). This is the most extreme dataset in our experiments in terms of the contrast in complexity between the non-semantic correlating factor (background colour) vs. stable features (objects).

COCO-ON-PLACES: Here we superimpose the same COCO objects on scenes from the PLACES dataset (Zhou et al., 2017), with the place-scenes acting as the bias (figure 2, right). See Appendix B.1 for more details about how these datasets are constructed. While the backgrounds in this dataset are more complex than colour, they still act as biasing factors, as indicated in the relatively poorer performance at systematic generalisation, and were selected due to visually obvious and distinct colour or texture.

4.5.3. Results

In all cases, we have used the partition predictor to infer the two groups. The partition accuracies for the three datasets at the end of one epoch of training the base models are in the table below. We tested a more naïve approach by applying K-Means clustering to the losses, but found it to under-perform, possibly because it cannot account for a consistent feature bias learned by our reference model.

Coloured MNIST	COCO-on-Colours	COCO-on-places
97.26 ± 0.71	98.22 ± 1.05	80.43 ± 1.41

In Tables 2,3,4, we find that significant improvements can be achieved using group invariance methods. All hyper-parameters for the results in this set are picked on a validation set consisting of a subset of colours or backgrounds that are different from both the training and test sets, and an equally sized subset of systematically varying colours or backgrounds from the biased majority group. In all cases, the split is learned after one epoch of training, and the various penalties dropped in at this point with a linearly ramped-in penalty co-efficient. Details about hyper-parameter selection are in Appendix B.3.

While conditional variants perform better at systematic generalisation for coloured MNIST, perhaps owing to our hyper-parameter selection procedure of using a mixed-shift validation set, performance at systematic shift appears to be traded off with non-systematic shift in some cases for the more complex datasets. All aggregates are over 5 trials.

4.5.4. Practical considerations for hyper-parameter selection

While we find that with the use of group invariance penalties it is possible to encourage reliance upon complex persistent correlations in the presence of dominant simple biases, this can sometimes come at a cost to in-distribution performance when picking hyper-parameters using validation sets with specific distributional shift. One might reasonably expect that this can be mis-aligned with real-life situations: in practice, one typically does not have access to data corresponding exactly to unexpected scenarios, besides not expecting to encounter

TABLE 3. Generalisation performance on COCO-ON-COLOURS.

Methods	In-distribution	Non-systematic shift	Systematic shift	Anomaly detection
Base (ERM)	90.57 ± 1.28	26.81 ± 4.93	1.10 ± 0.36	5.47 ± 0.08
IRMv1	91.61 ± 0.38	32.30 ± 4.52	2.11 ± 0.30	5.81 ± 0.17
REx	**91.69 ± 0.50**	36.57 ± 4.03	2.69 ± 0.81	5.73 ± 0.14
GroupDRO	43.06 ± 2.26	41.32 ± 4.39	43.24 ± 2.89	20.05 ± 3.08
Reweight	42.42 ± 3.47	47.56 ± 2.27	49.12 ± 1.63	18.15 ± 3.81
cIRMv1	91.53 ± 0.31	31.11 ± 4.51	1.74 ± 0.40	5.87 ± 0.16
cREx	74.75 ± 14.14	32.29 ± 7.71	29.75 ± 5.16	19.77 ± 14.98
cGroupDRO	41.10 ± 2.37	41.83 ± 2.96	42.10 ± 2.15	**21.81 ± 5.40**
cMMD	89.87 ± 1.13	55.02 ± 2.29	27.36 ± 1.57	8.82 ± 0.70
PGI	78.23 ± 2.01	**55.57 ± 4.60**	**51.62 ± 3.09**	18.84 ± 2.11

TABLE 4. Generalisation performance on COCO-ON-PLACES.

Methods	In-distribution	Non-systematic shift	Systematic shift	Anomaly detection
Base (ERM)	81.06 ± 1.01	45.25 ± 0.96	29.18 ± 1.24	9.21 ± 0.21
IRMv1	80.93 ± 0.71	45.17 ± 0.92	28.78 ± 0.73	9.39 ± 0.60
REx	**81.55 ± 0.70**	45.35 ± 0.92	29.56 ± 0.77	9.46 ± 0.51
GroupDRO	76.05 ± 0.87	43.72 ± 0.43	31.83 ± 0.54	9.61 ± 0.55
Reweight	81.14 ± 0.80	45.84 ± 0.70	30.37 ± 1.16	9.75 ± 0.69
cIRMv1	80.08 ± 1.90	44.96 ± 2.88	30.06 ± 2.07	9.64 ± 0.94
cREx	81.50 ± 0.76	45.44 ± 0.96	29.12 ± 0.97	9.17 ± 0.59
cGroupDRO	78.25 ± 0.31	41.69 ± 0.08	28.16 ± 0.91	9.45 ± 0.22
cMMD	79.64 ± 0.73	**49.44 ± 0.99**	35.86 ± 0.66	9.80 ± 0.45
PGI	75.00 ± 0.85	46.10 ± 0.79	**36.25 ± 0.42**	**11.12 ± 0.85**
cMMD (oracle split)	**75.05 ± 0.98**	47.88 ± 1.03	37.40 ± 1.07	10.76 ± 0.61
PGI (oracle split)	70.63 ± 0.48	**48.11 ± 0.82**	**42.69 ± 0.84**	**12.56 ± 1.20**

situations outside the training distribution nearly as often as situations for which a model has been trained and deployed. A practitioner might wish to aim for a clearer trade-off with such situations, with prior knowledge of how often they might arise compared to in-distribution situations, and with a surrogate validation set to model distributional shift. Here, we will simply show that picking hyper-parameters without assuming access to validation sets consisting of systematic distributional shift can still provide improvements over the baseline reference model. We consider three cases.

NS: Hyper-parameters are picked using only the validation set for non-systematic distributional shift (which consists of backgrounds that are different from those in the training set and test sets). This models the situation where we have access to some data that is different

TABLE 5. Hyper-parameters with different validation sets for COLOURED MNIST

Validation	In-distribution	Non-systematic shift	Systematic shift	Anomaly detection
NS+S (PGI)	99.05 ± 0.08	98.58 ± 0.06	98.48 ± 0.05	89.42 ± 1.95
NS (PGI)	99.31 ± 0.05	98.21 ± 0.26	97.54 ± 0.41	76.00 ± 4.06
NS+ID (PGI)	99.30 ± 0.07	98.31 ± 0.27	97.48 ± 0.45	76.07 ± 5.67
ID only (PGI)	99.69 ± 0.03	63.62 ± 2.05	58.18 ± 2.05	11.81 ± 1.89
Base (ERM)	99.60 ± 0.02	53.26 ± 1.89	38.72 ± 2.27	7.70 ± 0.23

from our training data, and is also considered somewhat representative of any shifts we might encounter.

NS + ID: Hyper-parameters are picked using an (equally-weighted) average of the NS and the in-distribution validation sets. If we have prior knowledge of the likelihood of encountering data from out-distributions in the wild, we could use this prior to use an appropriately sampled validation set for hyper-parameter optimisation.

ID ONLY: Hyper-parameters are picked using only the in-distribution validation set.

We show results for the different schemes for our method in Tables 5, 6, 7. While the accuracies under distributional shift are, as expected, less strong than in the previous set of results (NS+S in the tables), we still find improvements over the reference model, indicating that one can still achieve an improved classifier.

In Appendix B.4, we show similar results with all methods, and include only the best performing method for both generalisation under systematic and non-systematic shift corresponding to the different validation strategies in the tables in this section.

TABLE 6. Hyper-parameters with different validation sets for COCO-ON-COLOURS

Validation	In-distribution	Non-systematic shift	Systematic shift	Anomaly detection
NS+S (PGI)	78.23 ± 2.01	55.57 ± 4.60	51.62 ± 3.09	18.84 ± 2.11
NS (PGI)	85.78 ± 1.45	51.02 ± 2.32	38.85 ± 2.29	15.71 ± 3.25
NS+ID (PGI)	85.78 ± 1.45	51.02 ± 2.32	38.85 ± 2.29	15.71 ± 3.25
ID only (cMMD)	92.51 ± 0.41	44.59 ± 3.28	10.48 ± 0.98	6.05 ± 0.23
Base (ERM)	90.57 ± 1.28	26.81 ± 4.93	1.10 ± 0.36	5.47 ± 0.08

TABLE 7. Hyper-parameters with different validation sets for COCO-on-PLACES

Validation	In-distribution	Non-systematic shift	Systematic shift	Anomaly detection
NS+S (cMMD)	79.64 ± 0.73	49.44 ± 0.99	35.86 ± 0.66	9.80 ± 0.45
NS (cMMD)	79.64 ± 0.73	49.44 ± 0.99	35.86 ± 0.66	9.80 ± 0.45
NS+ID (cMMD)	79.64 ± 0.73	49.44 ± 0.99	35.86 ± 0.66	9.80 ± 0.45
ID only (PGI)	80.99 ± 0.52	47.63 ± 0.90	31.91 ± 0.89	9.59 ± 0.89
Base (ERM)	81.06 ± 1.01	45.25 ± 0.96	29.18 ± 1.24	9.21 ± 0.21

4.6. Conclusion

Our experiments investigate the potential usefulness of invariance penalties and methods at improving performance under distributional shift, such as systematic generalisation and semantic anomaly detection.

While our exploratory experiments are conducted in disambiguated synthetic setups, next steps would involve investigating the potential for extending these approaches to real datasets used in the field. Since such methods cannot work when spurious correlations are completely pervasive, it is important to include sufficient diversity of data sources and curation in order to be able to reap the advantages such techniques can afford us in real world applications. We note that peculiarities in datasets and problems might give rise to different potential failings at robustness, calling for more targeted invariance methods.

We find that our method of learning features that result in matched predictive behaviour throughout training appears to hold promise at handling certain distributional shifts, although it does not always perform best across different validation schemes. A practical line of inquiry would be the question of how to make trade-offs in performance between in-distribution and unexpected situations.

Acknowledgments

We thank Ishaan Gulrajani, Tim Cooijmans, David Krueger, and Phong Nguyen for useful discussions, and anonymous reviewers for constructive feedback. We acknowledge the computational resources provided by Compute Canada and Mila, and the financial support of Hitachi, Microsoft Research, CIFAR, and the Natural Sciences and Engineering Research Council of Canada (NSERC Discovery Grant).

Prologue to the third article

Online adaptation to black-box label-shift in the presence of conditional-shift. Faruk Ahmed and Aaron Courville. *Unpublished*.

Abstract. We consider an out-of-distribution setting where trained predictive models are deployed online in new locations (inducing *conditional-shift*), such that these locations are also associated with differently skewed target distributions (*label-shift*). While approaches for online adaptation to label-shift have recently been discussed by Wu et al. (2021), the potential presence of concurrent conditional-shift has not been considered in the literature, although one might anticipate such distributional shifts in realistic deployments. In this article, we empirically explore the effectiveness of online adaptation methods in such situations on three synthetic and two realistic datasets, comprising both classification and regression problems. We show that it is possible to improve performance in these settings by learning additional hyper-parameters to account for the presence of conditional-shift by using appropriate validation sets.

Key words: *label-shift, generalized target-shift, out-of-distribution generalization, Bayesian.*

Context. In this article, we adopt a somewhat narrow focus, based on considerations of how AI applications are used in the real world, and some of the practical constraints that accompany this usage. While adapting to distributional-shift in a particular test location using techniques such as domain adaptation is likely to improve performance, doing this at scale is challenging. For example, a company servicing tens of thousands of clients with large proprietary models over an API (for example, the cloud-based AI services provided by tech companies) would find it impractical to make and re-train multiple copies of such models in order to personally adapt to every client during online usage. However, adjusting the predictive distribution for a black-box model in a particular deployment location can provide an efficient way to adapt to changes in *label-distributions* (Lipton et al., 2018). Such practical motivations have resulted in significant exploration in the literature for black-box label-shift adaptation, and recently extended to an online setting by Wu et al. (2021).

Existing work on black-box label-shift estimation and correction has relied on a crucial assumption: that there is no *conditional shift* accompanying the label-shift, i.e. $\mathbb{P}(x \mid y)$ does not change across training to test environments. We suggest that this is unlikely to be satisfied in practice – in realistic OOD scenarios, everything can, and likely will, change. Another assumption made in a significant portion of later literature in this area involves the ability to reliably estimate a confusion matrix, used in recent approaches to label-shift problems. Specifically, the literature holds out significantly large-sized validation sets to estimate such confusion matrices. In real life settings, one often ends up with highly imbalanced datasets, and it is not always feasible to hold out a large fraction of training data, particularly in a post-hoc settings when a model has already been trained. Given such considerations, it seems relevant to revisit approaches to label-shift under less presumptive experimental settings.

Article contributions. We consider an online, post-hoc, novel-deployment setting, which includes both label-shift (per unique test-time deployment location) as well as a distributional shift in features (due to the change in location). We conduct an empirical study of the methods introduced in Wu et al. (2021), with the positive finding that the methods generally continue to improve numbers over a base classifier. We explore some heuristic modifications to these methods – particularly in terms of selection of a validation set for estimating confusion matrices and scaling hyper-parameters – and find that we are often able to improve performance further in our experimental settings. Finally, we reinterpret one of the existing baselines under a Bayesian perspective, allowing us to derive an equivalent method for analogous regression problems.

Author contributions. The contributions of the authors are the following.

- I proposed the online black-box experimental setup, derived the adaptation method, implemented the experiments, and wrote the draft.
- Aaron Courville supervised and provided feedback on the project.

Chapter 5

Online black-box adaptation to label-shift in the presence of conditional-shift

5.1. Introduction

We consider a setting where we have black-box access to a predictive model which we are interested in deploying online in different places with skewed label distributions. For example, such situations can arise when a cloud-based, proprietary service trained on large, private datasets (like the Vision APIs served by tech companies) serves several clients real-time in different locations. Every new deployment can be associated with label-shift. Recently, Wu et al. (2021) discuss the problem of online adaptation to label-shift, proposing two variants based on classical adaptation strategies – *Online Gradient Descent* (OGD) and *Follow The Leader* (FTH). Adapting the output of a model to a new label-distribution without an accompanying change in the label-conditioned input distribution only requires an adjustment to the predictive distribution (in principle). Therefore, both methods lend themselves to online black-box adaptation to label-shift, which makes on-device, post-hoc adjustments to the predictive distribution feasible under resource constraints.

In this article, we empirically explore such methods when the underlying assumption of an invariant conditional distribution is broken. Such situations are likely to arise in reality. For example, in healthcare settings there are often differing rates of disease-incidence (label-shift) across different regions (Vos et al., 2020) accompanied by conditional-shift in input features at different deployment locations, for example in diagnostic radiology Cohen et al. (2021). In notation, for input variable x and target variable y, we have that $P^{\text{new}}(x \mid y) \neq P(x \mid y)$

and $P^{\text{new}}(y) \neq P(y)$, for a training distribution P and a test distribution P^{new} in a new deployment location.

Contributions. Our contributions are as follows.

- We conduct an empirical study of the FTH and OGD methods introduced by Wu et al. (2021) in black-box label-shift settings with concurrent conditional-shift, a situation likely to arise in realistic deployments.
- We explore the question of how to potentially improve performance in such practical settings by computing confusion matrices on OOD validation sets, and show that adding extra hyper-parameters can contribute to further improvements.
- We reinterpret a simplified variant of FTH under a more general Bayesian perspective, enabling us to develop an analogous baseline for online adaptation in regression problems.

5.2. Background

We begin with a brief review of online adaptation methods for label-shift for classification problems, based on the recent discussion in Wu et al. (2021). While their motivation is temporal drift in label-distributions, we consider the case where a single model is serving several clients online in different locations, each with their own skewed label-distribution that does not change even further with time. Windowed or temporally-attenuated versions of the methods can be expected to be applicable for temporal shifts in label-distributions. If the training set label-distribution is $P(y)$ and the label-distribution in the new location is $P^{\text{new}}(y)$, and if we assume $P^{\text{new}}(x \mid y) = P(x \mid y)$, then the following holds

$$P^{\text{new}}(y \mid x) = \frac{P(x \mid y)P^{\text{new}}(y)}{P^{\text{new}}(x)} = \frac{P(y \mid x)P(x)}{P(y)}\frac{P^{\text{new}}(y)}{P^{\text{new}}(x)} \propto \frac{P^{\text{new}}(y)}{P(y)}P(y \mid x), \tag{5.2.1}$$

i.e., the location-adjusted output distribution is simply a reweighting of the output distribution from the base underlying predictive model. Wu et al. (2021) follow along past work on label-shift adaptation by restricting the hypothesis space for f to be that of re-weighted classifiers, since Eq. 5.2.1 implies that one only needs to re-weight the predictive distribution to account for label-shift. The parameter vector for this classifier is simply the vector of probabilities in $P^{\text{new}}(y)$, henceforth referred to as $\boldsymbol{p}$, and we will similarly use $\boldsymbol{q}$ to represent the training-set probability distribution, $P(y)$. Given an underlying predictive model f, the adjusted classifier

rule is therefore given by

$$g(x; f, \boldsymbol{q}, \boldsymbol{p}) = \arg\max_{y \in [K]} \frac{\boldsymbol{p}[y]\, P_f(y \mid x)}{\boldsymbol{q}[y]}, \tag{5.2.2}$$

where $P_f(y \mid x)$ is the predictive distribution produced by an underlying base model f; for example, a softmax distribution produced by a neural network, and there are K classes in our dataset.

5.2.1. Online adaptation algorithms

Wu et al. (2021) present two online updating methods to estimate $\boldsymbol{p}$ – Online Gradient Descent (OGD) and Follow The History (FTH).

If we assume knowledge of a confusion matrix for a classifier f in a new location, $C^{\text{new}}(f) \in \mathcal{R}^{K \times K}$, such that $C_f^{\text{new}}[i, j] = P_{x \sim P^{\text{new}}(x|y=i)}(f(x) = j)$, then Wu et al. (2021) show that the expected error rate in this new location can be derived as a function of the label-distribution $P^{\text{new}}(y)$. If we represent $P^{\text{new}}(y)$ as a K-dimensional probability vector $\boldsymbol{q}^{\text{new}}$, the expected error rate is given as

$$\ell^{\text{new}}(f) = \sum_{i=1}^{K} \Big(1 - P_{x \sim P^{\text{new}}(x|y=i)}(f(x) = i)\Big) \cdot \boldsymbol{q}^{\text{new}}[i] = \langle \mathbf{1} - \text{diag}(C_f^{\text{new}}), \boldsymbol{q}^{\text{new}} \rangle, \tag{5.2.3}$$

where $\mathbf{1}$ is the all-ones vector. Since we have assumed no conditional-shift so far, $C_f^{\text{new}} = C_f$, i.e. the confusion matrix remains invariant under label-shift. This implies one can optimize the expected error rate in the new deployment location using a confusion matrix estimated from a large in-distribution validation set, C_f, in place of C_f^{new} in Eq. 5.2.3.

Online Gradient Descent (OGD). Assuming that $\text{diag}(C_f)$ is differentiable $wrt f$, we can update f to minimize the expected error rate. We would typically not be aware of the true label-distribution in the new deployment location. However, when the confusion matrix C_f is invertible, we can compute an unbiased estimate of this distribution, given as $\hat{\boldsymbol{q}}^{\text{new}} = \left(C_f^{\top}\right)^{-1} \boldsymbol{e}$, where $\boldsymbol{e}$ is a one-hot vector for the predicted category. Using this, Wu et al. (2021) present an unbiased gradient of $\ell^{\text{new}}(f)$,

$$\nabla_f \hat{\ell}^{\text{new}}(f) = \mathbb{E}_{P^{\text{new}}}\Big[\frac{\partial}{\partial f}[\mathbf{1} - \text{diag}(C_f)]^{\top} \cdot \hat{\boldsymbol{q}}^{\text{new}}\Big]. \tag{5.2.4}$$

When the hypothesis space is restricted to the space of re-weighted classifiers g (Eq. 5.2.2) this gradient is only over $\boldsymbol{p}$. Wu et al. (2021) show how we might use effective numerical methods

to estimate this gradient. In the online setting, $\boldsymbol{p}$ is updated after seeing new examples, hence the $t+1$-th gradient update is performed by computing the gradient at the current point $\boldsymbol{p}_t$, followed by a projection to the probability simplex,

$$\nabla_{\boldsymbol{p}} \hat{\ell}^{\text{new}}(\boldsymbol{p})\Big|_{\boldsymbol{p}=\boldsymbol{p}_t} = \mathbb{E}_{P^{\text{new}}}\Big[\frac{\partial}{\partial \boldsymbol{p}}[\mathbf{1} - \text{diag}(C_g)]^\top \cdot \hat{\boldsymbol{q}}^{\text{new}}\Big]\Big|_{\boldsymbol{p}=\boldsymbol{p}_t} \tag{5.2.5}$$

$$\boldsymbol{p}_{t+1} = \text{Proj}_{\Delta^{K-1}}\left(\boldsymbol{p}_t - \eta \cdot \nabla_{\boldsymbol{p}} \hat{\ell}^{\text{new}}(\boldsymbol{p})\Big|_{\boldsymbol{p}=\boldsymbol{p}_t}\right), \tag{5.2.6}$$

where η is the learning rate and Proj is the projection operator.

Follow The History (FTH). The update rule for $\boldsymbol{p}_t$ in FTH is simpler and more efficient (in terms of memory and time complexity), given by

$$\boldsymbol{p}_{t+1} = \frac{1}{t}\sum_{\tau=1}^{t} \hat{\boldsymbol{q}}_\tau^{\text{new}}, \tag{5.2.7}$$

where $\hat{\boldsymbol{q}}_\tau^{\text{new}}$ is the estimate for the label distribution at the τ-th iteration. Empirical evidence in Wu et al. (2021) suggests that FTH performs very competitively with OGD, and might be preferred in highly resource-constrained settings.

5.3. Unmet assumptions in practice

We now consider applying the above strategies in cases where some of the assumptions in the above section are broken. While it is difficult to make conclusive theoretical statements in situations when these assumptions break, we propose some heuristics which we evaluate empirically.

5.3.1. The assumption of invariant $P(x \mid y)$ can break

In realistic deployments in new locations, it is likely that along with a differently skewed label-distribution, the conditional distribution will change as well, i.e. $P^{\text{new}}(x \mid y) \neq P(x \mid y)$. In our study, we will assume that this distributional shift only takes place within the same domain, and along (potentially spuriously-correlated) non-semantic features, leaving the semantic features intact, a setting likely to be manifested in different deployment locations.

Heuristic 1. One possibility to adapt the above methods to settings with concurrent conditional-shift is to estimate the confusion matrix on an OOD validation set. Intuitively, an

IID-estimated confusion matrix is likely to be over-confident, and a surrogate-OOD validation set can better reflect performance at test-time OOD settings.

Heuristic 2. We propose to add extra scaling hyper-parameters in the decision rule in Eq. 5.2.2. Specifically, we add the scaling hyper-parameters λ_u and λ_y before making a test prediction,

$$\tilde{g}(x; f, \boldsymbol{q}, \boldsymbol{p}) = \arg\max_{y \in [K]} \; \log P_f(y \mid x) + \lambda_u \log \boldsymbol{p}[y] - \lambda_y \log \boldsymbol{q}[y], \tag{5.3.1}$$

where we have rewritten the rule in log-space. In this formulation, $\log P_f(y \mid x) = \text{logit}[y] - Z(x)$, so we can drop the normalizing term. This results in a predictive rule that is a form of *logit-adjustment* (Menon et al., 2021). Intuitively, these hyper-parameters play the role of determining how much of the training prior to "subtract", and how much weight to assign to the pseudo-label based re-adjustment. When these magnitudes are learned on validation sets representing a combination of label-shift and conditional-shift, one can hope to further improve at novel test-time deployments.

5.3.2. Confusion matrices can be non-invertible

Existing work on label-shift based on confusion matrices rely on a significantly large held-out validation set to estimate a robust confusion matrix. When the underlying dataset is highly class-imbalanced with several categories and limited-size validation sets, one can easily end up with a non-invertible confusion matrix. Lipton et al. (2018) suggests two main possibilities – use of a soft confusion matrix, or a pseudo-inverse. In our experiments on a large-scale realistic dataset, we find both choices to lead to degraded performance. We find that simply using an identity matrix approximation can recover some of the performance drops (see Appendix C.4). When using FTH with an identity C_f, this corresponds to simply using the pseudo-labels up to time t to estimate the label-distribution. However, naively using the identity matrix in Eq. 5.2.7 might lead to a practical problem: after seeing the first data-point, $\boldsymbol{p}$ would be a one-hot vector, and thus enforce the same prediction at the next iteration when using Eq. 5.2.2. A fix would be to use a "pseudo-count" to smooth initial conditions, which is reminiscent of Bayesian posterior updates. In the next section, we use this realization as a starting point to suggest a simpler as well as more general framework. This framework then enables us to develop an equivalent online label-shift adaptation method for regression problems.

5.4. A Bayesian perspective

If we use the vector $\boldsymbol{\alpha}$ to keep online counts of predictions, with an initialized $\boldsymbol{\alpha}_0$, such that

$$\boldsymbol{\alpha}_t[k] = \sum_{\tau=1}^{t} \mathbf{1}[\hat{y}_\tau = k] + \boldsymbol{\alpha}_0 = \mathbf{1}[\hat{y}_t = k] + \boldsymbol{\alpha}_{t-1}[k], \tag{5.4.1}$$

then using an identity confusion matrix in Eq. 5.2.7 corresponds to the following update rule,

$$\boldsymbol{p}_{t+1}[k] = \frac{\boldsymbol{\alpha}_t[k]}{\sum_{k'=1}^{K} \boldsymbol{\alpha}_t[k']}. \tag{5.4.2}$$

We recognize that this update-rule corresponds exactly to the posterior predictive distribution computed using a Categorical likelihood with a Dirichlet prior, and using a recursive rule for updating the posterior. More precisely, if we use

$$\phi \sim \mathrm{Dir}(\boldsymbol{\alpha}), \tag{5.4.3}$$

$$y \mid \phi \sim \mathrm{Cat}(\phi), \tag{5.4.4}$$

where $\phi \in \Delta^{K-1}$ are the parameters of the Categorical distribution, in the following update equations

$$P_t(\phi) \propto P(y_t \mid \phi)\, P_{t-1}(\phi), \tag{5.4.5}$$

$$P_{t+1}(y) = \int_\phi P(y \mid \phi)\, P_t(\phi)\, d\phi, \tag{5.4.6}$$

then we arrive at Eq. 5.4.2 using Eq. 5.4.6, and Eq. 5.4.1 using Eq. 5.4.5. See Appendix C.1 for a derivation of Eq. 5.4.5. In practice, y_t is not available to us, and we use the pseudo-label $\hat{y}_t$ instead, as in FTH.

5.4.1. Extension to regression problems

While adaptation for regression problems has been discussed more generally (Cortes and Mohri, 2011; 2014; Zhang et al., 2013), an analogous discussion for online black-box label-shift adaptation is missing for regression. We adapt the general online update rules in Eq. 5.4.5, 5.4.6 for regression problems undergoing similar concurrent test-time distributional shifts. A natural choice is to use Gaussians to model the distributions over the continuous target variable,

$$P_f(y \mid x) \propto \exp\Big(-\frac{\lambda_x}{2}\Big(y - f(x)\Big)^2\Big), \tag{5.4.7}$$

$$P(y) \propto \exp\Big(-\frac{\lambda_y}{2}\Big(y-m\Big)^2\Big), \tag{5.4.8}$$

where λ_x, λ_y are the precision parameters and m is the training set mean. The parameters ϕ in Eq. 5.4.5 are now the mean and precision parameters for y in the new deployment location. We use the Normal-Gamma distribution to model the posterior over these parameters, since this is the conjugate distribution for Gaussians with unknown mean and precision (DeGroot, 2004),

$$P(\mu^{\text{new}}, \lambda^{\text{new}}) = \mathcal{N}\Big(\mu^{\text{new}} \mid \mu, \frac{1}{\kappa\lambda^{new}}\Big)\,\text{Ga}(\lambda^{new} \mid a, b). \tag{5.4.9}$$

Combined with the Gaussian likelihood in Eq. 5.4.6, this yields $P^{\text{new}}(y)$ in the form of a Student's t-distribution,

$$P^{\text{new}}(y) \propto \left(1 + \frac{L}{2a}(y-\mu)^2\right)^{-\frac{2a+1}{2}}, \tag{5.4.10}$$

where $2a$ is the number of degrees of freedom, and $L = \frac{a\kappa}{b(\kappa+1)}$. Using these, our predictive function (in log-space) takes the form

$$\arg\min_y \ \frac{\lambda_x}{2}\Big(y - f(x)\Big)^2 - \frac{\lambda_y}{2}\Big(y-m\Big)^2 + \frac{2a+1}{2}\log\Big(1 + \frac{L}{2a}(y-\mu)^2\Big). \tag{5.4.11}$$

Setting the derivative *wrt* y to zero yields a cubic equation (see Appendix C.2.1), which we can solve to find roots. A positive sign of the second derivative of the objective tells us if a solution is a (local) minima. When we have one real solution with a positive second derivative, we use this; when we have multiple real solutions with positive second derivatives, we pick the one that corresponds to the smallest objective; when we have no real solutions with positive second derivatives, we do not update $\mathbb{P}(y \mid x)$, retaining $f(x)$ as the solution. Empirically, we find that the condition for no local minima does not arise for optimal choices of hyper-parameters (also see Appendix C.2.2).

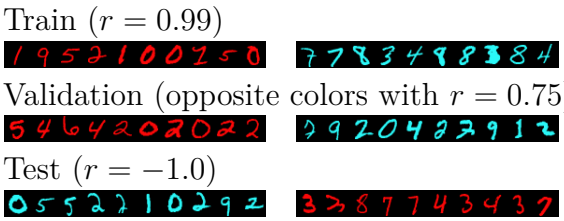
The update equations at the t-th step follow from the computation of the posterior using Eq. 5.4.5 (see Murphy (2007), for example, for the derivation of these update steps) and are given as:

$$a_{t+1} = a_t + 1/2, \tag{5.4.12}$$

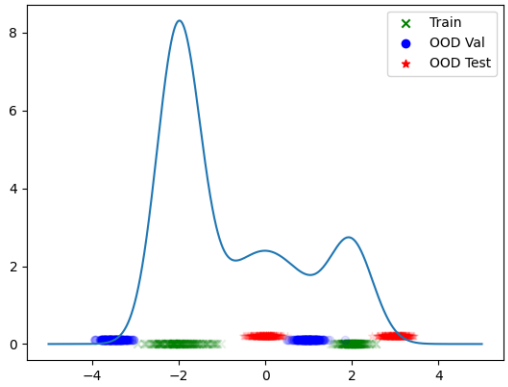
$$\kappa_{t+1} = \kappa_t + 1, \tag{5.4.13}$$

$$\mu_{t+1} = \frac{\kappa_t\mu_t + \hat{y}_{t+1}}{\kappa_t + 1}, \tag{5.4.14}$$

$$b_{t+1} = b_t + \frac{\kappa_t(\hat{y}_{t+1} - \mu_t)^2}{2(\kappa_t+1)}. \tag{5.4.15}$$



(A) Synthetic variant of the MNIST dataset constructed by using colors to correspond to sources with skewed label-distributions. The colors are flipped for validation and test with different correlation strengths, corresponding to (almost completely) reversing the label-skew at the sources at test-time.



(B) Synthetic Mix-of-Gaussians data. Differently colored regions along the x-axis correspond to training, validation and test samples, with different regions of the same color corresponding to different sources/locations.

Fig. 1. Synthetic MNIST and Gaussian datasets.

The hyper-parameters λ_x (output precision) and κ (equivalent of the smoothing pseudo-count α_0 in classification) are picked on the validation set, along with a scaling pre-multiplier for the precision λ_y (analogous to the classification setup). $\hat{y}_{t+1}$ is the prediction, treated as a pseudo-label. In order to place uniform priors over the output range, we will simulate a uniform set of samples over the output range. $\mu = \mathbb{E}[y^{\text{pseudo}}]$ is the mean of the pseudo-samples, and β is initialized as $0.5(\kappa - 1)\text{Var}(y^{\text{pseudo}})$ (see Appendix C.2.3 for details).

5.5. Experiments

We compare variants of online label-shift methods based on our discussion above on a mix of synthetic and realistic datasets to the un-adjusted model performance (BASE).

- FTH and OGD: These are the variants proposed in Wu et al. (2021). We evaluate both for two choices of confusion matrices each – computed using the in-distribution validation set, and using the out-of-distribution validation set (our HEURISTIC 1). We refer to these two alternatives as (C-IID) and (C-OOD).
- FTH-H and OGD-H: These are our modifications of FTH and OGD using the scaling hyper-parameters proposed in HEURISTIC 2. For both variants, we again evaluate two versions each, using (C-IID) and (C-OOD).
- FTH-H-B: This is our modification of FTH, with an additional pseudo-count hyper-parameter added for smoothing. The hyper-parameters are learned on the OOD validation sets. We call the regression variant FTH-H-B (R).
- OPTIMAL FIXED CLASSIFIERS: These oracle methods are derived by replacing $\mathbf{p}$ in Eq. 5.2.2 with the empirical location-wise label distributions, providing a sense of achievable gains if one were aware of the true label-distributions from the get-go. We include two variants – OFC, which uses Eq. 5.2.2, and OFC-H, which uses the modified update rule in Eq. 5.3.1 where the hyper-parameters are oracle hyper-parameters learned on the test-set.

When using OGD, we use the surrogate loss implementation in Wu et al. (2021) since it is both better-performing as well as much faster. This variant involves using a smooth approximation of the 0-1 loss allowing for direct gradient computation instead of a numerical approximation.

5.5.1. Classification problems

Synthetic: Skewed-MNIST. We split MNIST classes into two subsets: [0, 1, 2, 5, 9] and [3, 4, 6, 7, 8]. We use different colors to correspond to different deployment locations, similar to Arjovsky et al. (2019). In the training set, we color digits in a particular subset a particular color 99% of the time. This corresponds to a 99% skew in label-distributions across the two locations. The 1% cross-over instructs some color-invariance but not strongly enough to completely overcome the bias. The validation set uses opposing colours for the subsets, but with a 75% correlation – this represents a scenario where the class-distributions in different

FIG. 2. Skewed COCO-on-Places: Synthetic dataset constructed by superimposing COCO objects (Lin et al., 2014) on scenes from the Places dataset (Zhou et al., 2017). The 5 columns correspond to 5 sources of data, where the backgrounds correspond to examples of particular scenes, and the skew in number of examples per row correspond to the skew in label distribution we impose. Different background scenes are used for training, validation, and test sets.

locations change from that in training. Finally, the test set uses completely flipped colors in the two subsets compared to the training set – this implies reversed label-distributions, resulting in poorer baseline performance.

Since the overall class frequencies are balanced in the training set, we drop the $P(y)$ from the update rule in Eq. 5.2.2 and 5.3.1. With a 3-layer CNN trained for 20 epochs to 100% training set accuracy and 99.6% in-distribution test set accuracy, we find, in Table 1, that using online adjustments at test-time can lead to marked improvements for the base model in the test set. The numbers are averaged over 5 independent rounds of base-model training, with validation and test sets randomly shuffled for 5 trials for each round of training. (More details about dataset construction in Appendix C.3.1)

Synthetic: Skewed-COCO-on-Places. We construct a second, more photo-realistic, synthetic dataset by superimposing segmented objects from COCO Lin et al. (2014) on to scenes from the PLACES dataset Zhou et al. (2017), as in Ahmed et al. (2021). The scenes correspond to the notion of a deployment location, albeit with significant intra-location variation. For every such scene-represented source, we use a different class-distribution to simulate source-specific skews in the label distribution. In Fig. 2 the relative number of images per row represent the relative frequency of a particular class at a specific source. There are a total of $\sim 10K$ training images, $\sim 2.5K$ validation images (each for seen and unseen sources), and $\sim 6K$ test images (each for seen and unseen sources).

The validation and test sets are constructed similarly. For in-distribution validation and test sets, the same set of scenes as for training is used (with different instances), and for new-location validation and test sets, different sets of scenes are used. See Appendix C.3.3 for details about dataset construction. We train a ResNet-50 for 400 epochs with SGD+Momentum for

TABLE 1. Classification problems: Average accuracy on SKEWED-MNIST, SKEWED-COCO-ON-PLACES, and WILDS-IWILDCAM (also reporting macro F1-score for IWILDCAM). Overall trends indicate that our heuristics are helpful, and FTH-H-B is competitive or better without needing a confusion matrix.

Method	S-MNIST	S-COCO-ON-PLACES	IWILDCAM (Avg.)	IWILDCAM (F1)
BASE	82.59 ± 1.82	56.09 ± 0.66	73.10 ± 3.26	32.70 ± 0.16
FTH (C-IID)	93.12 ± 1.57	58.50 ± 0.55	71.41 ± 4.91	29.57 ± 0.93
FTH (C-OOD)	96.04 ± 1.03	58.94 ± 0.63	71.41 ± 4.91	29.57 ± 0.93
OGD (C-IID)	88.32 ± 2.06	57.37 ± 0.51	71.66 ± 4.56	32.56 ± 0.27
OGD (C-OOD)	95.75 ± 0.70	57.75 ± 0.29	73.11 ± 3.05	32.49 ± 0.41
FTH-H (C-IID)	98.21 ± 0.47	56.72 ± 0.84	73.75 ± 3.77	32.46 ± 0.31
FTH-H (C-OOD)	98.69 ± 0.31	57.81 ± 0.74	73.75 ± 3.77	32.46 ± 0.31
OGD-H (C-IID)	96.07 ± 1.76	57.58 ± 0.79	72.89 ± 3.30	31.74 ± 0.51
OGD-H (C-OOD)	98.91 ± 0.20	57.12 ± 0.15	73.36 ± 3.51	31.36 ± 0.41
FTH-H-B	97.46 ± 0.64	58.42 ± 0.49	74.10 ± 3.56	33.33 ± 1.31
OFC	99.24 ± 0.20	75.88 ± 0.33	79.19 ± 1.76	48.61 ± 0.27
OFC-H	99.26 ± 0.20	75.88 ± 0.33	81.07 ± 0.79	48.61 ± 0.27

the underlying model, achieving an in-distribution test accuracy of $\sim 75\%$. Since the overall distribution of classes is close to uniform, we again drop the marginal $P(y)$ term in Eq. 5.2.2 and 5.3.1. In Table 1 we again find improved performance over the unadjusted base model for all variants. Accuracy is aggregated across 20 random orderings of the test set (since the test-sets are smaller for this specific dataset), for 3 rounds of base-model training each.

WILDS-iWildCam. We use the variant of the IWILDCAM 2020 dataset Beery et al. (2021) curated by the WILDS set of benchmarks for out-of-distribution (OOD) generalization Koh et al. (2021). The data consists of burst images taken at camera traps, triggered by animal motion. The task is to identify the species in the picture, and the locations correspond to the unique camera trap the pictures are from. There are a total of 182 species in this version of the dataset across a total of 323 camera traps. There is significant skew in terms of species distribution across different camera traps, as well as the number of images available for each trap. The training set consists of $\sim 130K$ images from 243 traps; the in-distribution validation set consists of $\sim 7.3K$ images from the same traps as that in the training set but on different dates; the OOD validation set consists of $\sim 15K$ images taken at 32 traps that are different from the ones in the training set; the in-distribution test set consists of $\sim 8.1K$ images taken by the same camera traps as in the training set, but on different dates from both training and validation; finally, the OOD test set consists of $\sim 43K$ images taken at 48 camera traps that are different from those for all other splits.

TABLE 2. Regression problems: For the GAUSSIANS dataset the metric is mean squared error (lower is better), and for the PovertyMap folds the metric is Pearson's correlation co-efficient (higher is better), computed separately for average (ALL) and worst-group (WG) performance.

Dataset	BASE	FTH-H-B (R)
MIX-OF-GAUSSIANS	9.17 ± 2.17	4.35 ± 1.48

POVERTYMAP Fold	BASE	FTH-H-B (R)
A (ALL)	0.84	0.84 ± 0.00
B (ALL)	0.83	0.82 ± 0.00
C (ALL)	0.80	0.83 ± 0.00
D (ALL)	0.77	0.77 ± 0.00
E (ALL)	0.75	0.75 ± 0.00

POVERTYMAP Fold	BASE	FTH-H-B (R)
A (WG)	0.42	0.43 ± 0.00
B (WG)	0.52	0.50 ± 0.01
C (WG)	0.42	0.56 ± 0.01
D (WG)	0.50	0.56 ± 0.01
E (WG)	0.34	0.37 ± 0.00

Koh et al. (2021) trained ResNet-50 based models along with their curation of this dataset, also evaluating several methods for OOD generalization and releasing all models. We use their models trained with the domain generalization method CORAL (Sun and Saenko, 2016), since this model has improved performance over the ERM baseline. They released three sets of weights, trained with three random seeds. We evaluate all variants for each of the three seeds, with 3 random orderings each of the test set, and report aggregates in Table 1. Koh et al. (2021) recommend evaluation with both average accuracy as well as macro-F1 (since some species in the dataset are rare). We perform evaluation with both metrics, but use our own trained models for average accuracy – this is because Koh et al. (2021) trained their models optimizing for macro F1. We similarly trained CORAL-augmented base models optimizing the penalty coefficient and choice of early stopping.

We replace the confusion matrix with an identity matrix for evaluating methods on this dataset (for methods where a validation-set estimated confusion matrix is required). Confusion matrices evaluated on the validation sets are non-invertible for this dataset due to sparse class-representation and we found common alternatives to perform poorly (see Appendix C.4).

5.5.2. Regression problems

Synthetic: Mix-of-Gaussians. We create a synthetic regression dataset by constructing a curve from a mixture of Gaussians. We pick regions on the x-axis to correspond to training, validation, and test sets, such that every set samples data from two regions each, corresponding to two locations (see Appendix C.3.2). In Figure 1b, we depict the curve, along with sampling indicators for the different sets and sources. The points have been placed at

different heights for clearer visualization of overlaps. 500 points are sampled from the two training regions, and 250 each for the validation and test sets from their assigned regions. We train a 3-layer MLP with BatchNorm and ReLU activations and a mean squared loss for 100 epochs, yielding an in-distribution test mean squared error (MSE) of ~ 0.15. In Table 2 we find that online updating reduces the OOD test MSE significantly. Results are aggregates over five trials, with a different random sampling of all data, followed by training and validation each time. Full results and more experimental details are in Appendix C.3.2).

WILDS-PovertyMap. We use the WILDS variant of a *poverty mapping* dataset (Yeh et al., 2020). This is a dataset for estimating average household economic conditions in a region through satellite imagery, measured by an asset wealth index computed from survey data. The data comprises 8-channel satellite images with data from 23 African countries. The locations here correspond to different countries. Due to the smaller size of the dataset, Koh et al. (2021) recommend a five-fold evaluation, where every fold is approximately constructed as follows – 10K images from 13-14 countries in the training set; 1K images from the same countries for in-distribution validation; 1K images from these countries for in-distribution testing; 4K images from 4-5 countries not in the training set for OOD validation; and 4K images from 4-5 countries in neither training nor validation sets for OOD test.

The evaluation metric is Pearson's correlation between predicted economic index vs. actual index, as is standard in the literature (Yeh et al., 2020). Following Koh et al. (2021), we split the assessment into overall average as well as worst-group performance, which picks the worst performance across rural/urban subgroups. As with iWildCam, we use the CORAL-augmented base networks and weights released by Koh et al. (2021), but with our retrained versions for average correlation coefficient (since the validation choices for the released weights were for worst group performance). We evaluate separately for each fold (which have quite a bit of variance in base performance) with 5 random orderings of each of the test sets. In Table 2, we find that while there seems generally little to no improvement for average correlation, there are more significant improvements for three of five folds in terms of worst-group performance. As noted in Koh et al. (2021), a wide range of differences along many dimensions such as infrastructure, agriculture, development, cultural aspects play a role not only in determining wealth-distribution, but also in terms of how the features manifest in different places. Such real-world issues imply that validating for OOD performance is bound to be sensitive to problem types and the specific choices of validation sets used to tune hyper-parameters, and the differences that may arise between an OOD validation set and an OOD test set. This issue extends generally to all attempts at OOD generalization.

5.5.3. Takeaways

Our experiments are generally suggestive of the following takeaways.

- While invertible confusion matrices are not always achievable due to data scarcity (as modelled in our experiments with WILDS-iWildCam), a practitioner can adopt confusion-matrix free methods such as FTH-H-B, which we find to provide competitive or improved performance. Using OOD validation sets to estimate confusion matrices can improve results relative to using an IID validation set, although confusion matrices estimated on smaller-sized sets can be noisy.
- Learning additional scaling hyper-parameters can be useful for further improvements. We find this trend to not hold for Skewed-COCO-on-Places (FTH outperforms FTH-H and FTH-H-B). This might be due to the OOD validation set being farther from the OOD test set relative to the IID validation set, or an artifact of noise due to the relatively smaller size of the validation set – when picking oracle scaling hyper-parameters on the test set, we achieve an accuracy of 59.37 ± 0.89. In Table 3 we compare performance when learning hyper-parameters on different validation sets – IID/OOD/test (oracle). In general, OOD validation seems a better choice than IID validation.

5.6. Related work

Label-shift for classifiers. Saerens et al. (2002) provides a seminal discussion about adapting the output distribution of a classifier when the test set undergoes label-shift. This approach presumes access to the entire test set up front, or a sufficiently representative sample. More recent works have investigated other ways to estimate label-shift (Lipton et al., 2018; Azizzadenesheli et al., 2019) using confusion matrices, which partially inspired the methods in Wu et al. (2021) that we use as our foundation. It has been recently suggested (Alexandari et al., 2020; Garg et al., 2020) that the simple correction method in Saerens et al. (2002) often outperforms these later methods when combined with calibration. While Alexandari et al. (2020) perform their calibration using a held-out IID validation set for their iterative method, we adapt this strategy to the out-of-distributions setting by picking scaling hyper-parameters on an OOD validation set.

Test-time training. Another emerging line of literature focuses on updating neural network parameters using test data without being able to match training statistics with test statistics,

TABLE 3. (top) Classification problems: Performance when picking hyper-parameters on IID, OOD validation sets, or on (Oracle) test sets. (bottom) Regression problems: Performance when picking hyper-parameters on IID, OOD validation sets, or on (Oracle) test sets. For MIX-OF-GAUSSIANS, we use mean squared error as the metric (lower is better), while for POVERTYMAP the metric is the Pearson's correlation co-efficient (higher is better).

Datasets	Methods	IID validation	OOD validation	Oracle
S-MNIST	FTH-H	82.67 ± 1.79	98.69 ± 0.30	98.69 ± 0.30
	OGD	82.75 ± 1.77	95.75 ± 0.70	95.75 ± 0.70
	OGD-H	82.59 ± 1.82	98.91 ± 0.20	98.91 ± 0.20
	FTH-H-B	83.00 ± 1.79	97.46 ± 0.64	98.35 ± 0.52
S-COCO-ON-PLACES	FTH-H	57.42 ± 0.53	57.81 ± 0.74	59.05 ± 0.53
	OGD	57.72 ± 0.31	57.75 ± 0.29	57.75 ± 0.29
	OGD-H	57.31 ± 0.68	57.12 ± 0.15	58.10 ± 0.85
	FTH-H-B	58.59 ± 1.02	58.42 ± 0.49	59.37 ± 0.89
IWILDCAM (AVG)	FTH-H	73.52 ± 3.36	73.75 ± 3.77	74.13 ± 3.54
	OGD	69.42 ± 5.10	73.11 ± 3.05	73.16 ± 3.15
	OGD-H	73.41 ± 3.42	73.36 ± 3.51	73.53 ± 3.29
	FTH-H-B	73.90 ± 3.93	74.10 ± 3.56	74.41 ± 3.65
IWILDCAM (F1)	FTH-H	31.93 ± 1.56	32.46 ± 0.31	33.81 ± 0.30
	OGD	29.37 ± 2.15	32.49 ± 0.41	32.72 ± 0.06
	OGD-H	32.09 ± 0.29	31.36 ± 0.41	32.72 ± 0.15
	FTH-H-B	32.73 ± 2.78	33.33 ± 1.31	33.33 ± 1.31

Datasets	IID validation	OOD validation	Oracle validation
MIX-OF-GAUSSIANS	9.24 ± 2.76	4.35 ± 1.48	1.76 ± 0.59
POVERTYMAP-A (ALL)	0.80 ± 0.00	0.84 ± 0.00	0.84 ± 0.00
POVERTYMAP-B (ALL)	0.82 ± 0.00	0.82 ± 0.00	0.83 ± 0.00
POVERTYMAP-B (ALL)	0.82 ± 0.00	0.83 ± 0.00	0.83 ± 0.00
POVERTYMAP-B (ALL)	0.78 ± 0.01	0.77 ± 0.00	0.78 ± 0.00
POVERTYMAP-B (ALL)	0.72 ± 0.01	0.75 ± 0.00	0.75 ± 0.00
POVERTYMAP-A (WG)	0.43 ± 0.00	0.43 ± 0.00	0.45 ± 0.02
POVERTYMAP-A (WG)	0.33 ± 0.03	0.50 ± 0.01	0.52 ± 0.00
POVERTYMAP-A (WG)	0.50 ± 0.01	0.56 ± 0.01	0.58 ± 0.02
POVERTYMAP-A (WG)	0.46 ± 0.04	0.56 ± 0.01	0.57 ± 0.02
POVERTYMAP-A (WG)	0.36 ± 0.02	0.37 ± 0.00	0.37 ± 0.00

due to the potential lack of access to training data for the same topical reasons – data privacy and large datasets. Some examples include updating the Batch-Norm statistics optimizing for minimum test-time entropy Wang et al. (2021), or using self-supervised pseudo-labels to adapt the feature extraction part of the network Liang et al. (2020). Our setup here can be viewed as a form of test-time training, but in a more constrained setting, with inaccessible

model parameters and no resources to replicate an onsite-model by querying the black-box model, e.g. using distillation (Hinton et al., 2015).

Out-of-distribution generalization. There has been a recent surge in interest for methods aiming to learn stable or invariant features across different domains/environments/groups Sun and Saenko (2016); Arjovsky et al. (2019); Krueger et al. (2020); Sagawa et al. (2020). Such approaches have been demonstrated to be useful for certain types of distributional shifts, such as with improved minority group robustness Sagawa et al. (2020) and systematic generalization Ahmed et al. (2021). Our discussion in this article is complementary to this set of methods in OOD generalization research. One can use an underlying model trained with cross-group penalties that result in improved OOD generalization, and further improve performance by factoring in useful contextual information.

5.7. Conclusion

In this article, we empirically investigated the effectiveness of online black-box adaptation methods for label-shift when a key underlying assumption of invariant class-conditional input distributions is broken. We found that while existing methods can be effective to an extent regardless of conditional-shift, performance can be improved by adopting intuitive heuristics – in particular, estimating confusion matrices on OOD validation sets, and learning additional scaling hyper-parameters in the output adjustment step to account for shifting distributions.

Acknowledgements

We thank Aditya Menon, Ishaan Gulrajani, Chin-Wei Huang, Tim Cooijmans, and Olexa Bilaniuk for useful comments and feedback. We acknowledge the computational resources provided by Mila, and the financial support from Hitachi and CIFAR.

Chapter 6

Conclusion

In the articles presented, we broadly explored questions of how to study and improve predictive behaviour in novel settings. Under a compositional view of data generation, this involves two aspects – we would like our AI models to express reduced confidence when facing unfamiliar things; and we would like them to generalize to familiar things in unfamiliar settings. We presented perspectives about these problems with both pragmatic and philosophical considerations, particularly with regards to meaningful benchmarking and grounded-ness in real-life scenarios. We discussed different aspects of what makes for practical AI systems that can be deployed with some expectation of robustness, while being able to efficiently adapt in real-time using available information.

In the first article, we took a critical view of the problem of OOD detection, a problem statement primarily developed with AI safety in mind. We pointed out that the perspective of treating all possible variations as unsafe might lead to impractical directions since we realistically wish to be robust to non-semantic distributional shift while being sensitive to semantic anomalies, where the context of an application determines semanticity. Since this motivation ought to be reflected in benchmarking, we made the case for measuring semantic anomaly detection on held-out categories, when evaluating an object recognition model. Our arguments have been cited in the development of a larger scale semantic anomaly benchmark called ImageNet-O in Hendrycks et al. (2021).

With the intuition that a classifier trained on richer and more relevant features can be expected to exhibit greater uncertainty when facing unfamiliar things as well as inherit improvements at generalization, we showed that adding self-supervised multi-tasking objectives improve both semantic anomaly detection as well as test set accuracy. This intuition has been further

developed recently by Deecke et al. (2021), who showed that pre-training the feature extractor on larger datasets improves performance significantly beyond our results on our recommended benchmarks.

In the second article, we explored questions of both generalization to unfamiliar contexts as well as recognizing the novelty of unfamiliar things in familiar contexts. With a compositional perspective, we developed a synthetic test bed to empirically study how some existing robustness-methods perform at different variants of out-of-distribution settings. This also led us to developing a simple penalty that is promising in comparison to existing methods. Related to the first article, we improved upon our notion of evaluating semantic anomaly detection – rather than pit a held-out category against in-distribution images, we tested for lower predictive confidences on held-out objects in familiar contexts relative to familiar objects in unfamiliar contexts. Coupled with evaluating for OOD test accuracy, this can be a more meaningful, albeit synthetic, test of a classifier's alignment with the twin goals of robustness to non-semantic distributional shift and sensitivity to semantic shift. Our dataset construction framework has been adopted in a number of recent works (Zhang et al., 2021; Deecke et al., 2021; Zhou et al., 2022; Roburin et al., 2022).

Finally, in the third article, we considered a specific practical scenario when people wish to deploy AI models in the real world. Such attempts often come with several constraints, dictated by concerns of privacy or hardware limitations. We assumed an online setting with a black-box model deployed in novel environments, encountering a combination of label-shift and conditional-shift at each new location. We found that in such challenging circumstances it is nevertheless possible to find improvements over base performance using efficiently-computed adjustments to the output distribution, although there is still significant room for improvement.

By way of concluding remarks, a few pertinent lines of thought are presented below, which might make for interesting future exploration.

OOD generalization in pre-trained models. It is now common to adopt self-supervised pre-training on massive unlabeled data (Bommasani et al., 2021). Some work has suggested that this could alleviate OOD generalization failures – for example, Hendrycks et al. (2020) showed that models pre-trained on larger datasets can be more robust when fine-tuned on a task-specific dataset and tested on OOD data. While it is certainly plausible to assume that exposure to vast quantities of unlabelled data makes for richer, more "extrapolative spaces", we posit that this can only take care of certain non-systematic distributional shifts. Higher-order predictive rules inferred from task-specific small-size training sets can still

be prone to picking up non-robust/domain-specific features with high-level combinatorial interactions. Investigating such pathological modes empirically might be an interesting avenue for future exploration. Such directions can inspire both new benchmarks in the context of contemporary practice, as well as methods for enforcing robustness for such distributional shifts.

Trade-offs between IID vs. OOD performances. An aspect that seems to go ignored in current academic discourse is the pragmatic question of how to trade off IID vs. OOD performances. While the dominant narrative is that learning robust features is better, attempting to learn robust models can sometimes come at a cost to IID performance, since robust models can be harder to learn or require more specific data. This suggests we might be able to consistently achieve higher performances when we are guaranteed IID-operating conditions, since any domain-specific feature-correlations or modality can be expected to be preserved. In such cases, estimating both non-semantic and semantic distributional shift in real-time can be useful to determine which subset of predictive rules to rely upon. Research avenues here could include real-time distributional shift detection with sequential context (Bhaskhar et al., 2022) for specific shifts, and architectural design choices for enabling multiple predictive rules, perhaps using techniques for conditional computing (Bengio et al., 2013). In some cases, continual learning (Parisi et al., 2019) might be a practical approach for staying up to date on temporally shifting data distributions.

Online updates with human-in-the-loop feedback. Consumer-facing applications are usually trained on data obtained from a set of clients, and then deployed across a larger number of different clients. Unfamiliar situations can arise in such deployments, and one possibility is to update the parameters of the serving model on specific data points encountered at deployment. A mechanism for efficient choice of examples for such updates is to simply solicit user feedback interactively for uncertain or failure cases. This format might also be preferable at the front-end: consider a client-facing software that asks the operator to clarify if they meant X or Y, instead of making a false decision that requires later correction, possibly incurring greater expense than the inconvenience. Since clients are often concerned with data privacy, naive central-updating policies are likely to encounter resistance in certain domains, such as healthcare. Advances in *federated learning* (Li et al., 2020) might inspire tools for maintaining privacy when updating a global model with information from multiple sources in a trustworthy manner.

References

[1] Faruk Ahmed. Generative models for natural images. *Thèses et mémoires électroniques de l'Université de Montréal*, 2018.

[2] Faruk Ahmed and Aaron Courville. Detecting semantic anomalies. *AAAI*, 2020.

[3] Faruk Ahmed, Yoshua Bengio, Harm van Seijen, and Aaron Courville. Systematic generalisation with group invariant predictions. In *9th International Conference on Learning Representations (ICLR)*, 2021.

[4] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. *ArXiv e-prints*, 2018.

[5] Michael Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei shinn Ku, and Anh Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. *CVPR*, 2019.

[6] Amr Alexandari, Anshul Kundaje, and Avanti Shrikumar. Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In *International Conference on Machine Learning*, pages 222–232. PMLR, 2020.

[7] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016.

[8] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant Risk Minimization. *arXiv e-prints*, Jul 2019.

[9] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *CoRR*, 2019.

[10] Udit Arora, William Huang, and He He. Types of out-of-distribution texts and how to detect them. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10687–10701, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.835. URL https://aclanthology.org/2021.emnlp-main.835.

[11] Anand Avati, Kenneth Jung, Stephanie Harman, Lance Downing, Andrew Ng, and Nigam H. Shah. Improving palliative care with deep learning. *BMC Medical Informatics and Decision Making*, 18(4):122, Dec 2018.

[12] Shekoofeh Azizi, Laura Culp, Jan Freyberg, Basil Mustafa, Sebastien Baur, Simon Kornblith, Ting Chen, Patricia MacWilliams, S Sara Mahdavi, Ellery Wulczyn, et al. Robust and efficient medical imaging with self-supervision. *arXiv preprint arXiv:2205.09723*, 2022.

[13] Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. *arXiv preprint arXiv:1903.09734*, 2019.

[14] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[15] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. End-to-end attention-based large vocabulary speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4945–4949. IEEE, 2016.

[16] Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. Systematic generalization: What is required and can it be learned? In *ICLR*, 2019.

[17] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. *CoRR*, 2018.

[18] Sara Beery, Arushi Agarwal, Elijah Cole, and Vighnesh Birodkar. The iwildcam 2021 competition dataset. *arXiv preprint arXiv:2105.03494*, 2021.

[19] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

[20] Aharon Ben-Tal, Dick den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. Robust solutions of optimization problems affected by uncertain probabilities. 59(2), 2013. ISSN 0025-1909.

[21] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016.

[22] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[23] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019.

[24] Nandita Bhaskhar, Daniel L Rubin, and Christopher Lee-Messer. Trust-lapse: An explainable & actionable mistrust scoring framework for model monitoring. *arXiv*

preprint arXiv:2207.11290, 2022.

[25] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2178–2186. 2011.

[26] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[27] Kendrick Boyd, Kevin H. Eng, and C. David Page. Area under the precision-recall curve: Point estimates and confidence intervals. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 451–466. Springer Berlin Heidelberg, 2013.

[28] Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *International Conference on Learning Representations*, 2019.

[29] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

[30] Fabio Maria Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, 2019.

[31] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

[32] Jose Carranza-Rojas, Herve Goeau, Pierre Bonnet, Erick Mata-Montero, and Alexis Joly. Going deeper in the automated identification of herbarium specimens. *BMC Evolutionary Biology*, 17(1):181, Aug 2017.

[33] Rich Caruana. Multitask learning: A knowledge-based source of inductive bias. In *ICML*, 1993.

[34] Krzysztof Chalupka, Pietro Perona, and Frederick Eberhardt. Visual causal feature learning. *arXiv preprint arXiv:1412.2309*, 2014.

[35] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[36] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[37] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*,

pages 15750–15758, 2021.

[38] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[39] Noam Chomsky. *Syntactic Structures*. Mouton and Co., 1957.

[40] Joseph Paul Cohen, Paul Bertin, and Vincent Frappier. Chester: A web delivered locally computed chest x-ray disease prediction system. *CoRR*, abs/1901.11210, 2019.

[41] Joseph Paul Cohen, Tianshi Cao, Joseph D Viviano, Chin-Wei Huang, Michael Fralick, Marzyeh Ghassemi, Muhammad Mamdani, Russell Greiner, and Yoshua Bengio. Problems in the deployment of machine-learned models in health care. *CMAJ*, 193(35): E1391–E1394, 2021.

[42] Corinna Cortes and Mehryar Mohri. Domain adaptation in regression. In *International Conference on Algorithmic Learning Theory*, pages 308–323. Springer, 2011.

[43] Corinna Cortes and Mehryar Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, 2014.

[44] Elliot Creager, Jörn-Henrik Jacobson, and Richard Zemel. Environment inference for invariant learning. *ICML Workshop on Uncertainty and Robustness*, 2020.

[45] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR, 2021.

[46] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019.

[47] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. pages 233–240, 2006.

[48] Harm de Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C. Courville. Modulating early visual processing by language. *NIPS*, 2017.

[49] Lucas Deecke, Lukas Ruff, Robert A Vandermeulen, and Hakan Bilen. Deep anomaly detection by residual adaptation. *arXiv preprint arXiv:2010.02310*, 2020.

[50] Lucas Deecke, Lukas Ruff, Robert A Vandermeulen, and Hakan Bilen. Transfer-based semantic anomaly detection. In *International Conference on Machine Learning*, pages 2546–2558. PMLR, 2021.

[51] Morris H. DeGroot. *Optimal Statistical Decisions*, chapter 9, pages 155–189. John Wiley Sons, Ltd, 2004.

[52] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[53] Terrance DeVries and Graham W. Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

[54] Akshay Raj Dhamija, Manuel Günther, and Terrance Boult. Reducing network agnostophobia. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9157–9168. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/8129-reducing-network-agnostophobia.pdf.

[55] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2070–2079, 2017.

[56] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. *ICCV*, 2015.

[57] Lars Doorenbos, Raphael Sznitman, and Pablo Márquez-Neila. Data invariants to understand unsupervised out-of-distribution detection. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 133–150, Cham, 2022. Springer Nature Switzerland.

[58] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[59] John Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *arXiv preprint arXiv:1810.08750*, 2018.

[60] Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000. URL https://proceedings.neurips.cc/paper/2000/file/44968aece94f667e4095002d140b5896-Paper.pdf.

[61] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1657–1664, 2013.

[62] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1657–1664, 2013.

[63] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861 – 874, 2006.

[64] Peter Fedor, Jaromír Vanhara, Josef Havel, Igor Malenovsky, and Ian Spellerberg. Artificial intelligence in pest insect monitoring. *Systematic Entomology*, 34(2):398–400, 2009.

[65] Jerry A. Fodor. *The Language of Thought*. Harvard University Press, 1975.

[66] Jerry A. Fodor and Zenon W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3 – 71, 1988.

[67] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, January 2016.

[68] Saurabh Garg, Yifan Wu, Sivaraman Balakrishnan, and Zachary C Lipton. A unified view of label shift estimation. *arXiv preprint arXiv:2003.07554*, 2020.

[69] 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. *ICLR*, 2019.

[70] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pages 2551–2559, 2015.

[71] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *ICLR*, 2018.

[72] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[73] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. *NeuRIPS*, 2018.

[74] Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *International conference on machine learning*, pages 2839–2848. PMLR, 2016.

[75] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. 13(null), 2012. ISSN 1532-4435.

[76] 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. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[77] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

[78] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. *ICML*, pages 1321–1330, 2017.

[79] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[80] J. Hadamard. *Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées*. Mémoires présentés par divers savants à l'Académie des sciences de l'Institut de France: Éxtrait. Imprimerie nationale, 1908. URL http://books.google.com.au/books?id=BTEPAAAAIAAJ.

[81] Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/9d2682367c3935defcb1f9e247a97c0d-Paper.pdf.

[82] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[83] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[84] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[85] Christina Heinze-Deml and Nicolai Meinshausen. Conditional variance penalties and domain shift robustness. *CoRR*, 2017.

[86] Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

[87] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[88] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[89] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021.

[90] Daniel Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ICLR*, 2017.

[91] Daniel Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *ICLR*, 2019.

[92] Daniel Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. Pretrained transformers improve out-of-distribution robustness. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, July 2020.

[93] M.A. Hernan and J.M. Robins. *Causal Inference*. Chapman & Hall/CRC Monographs on Statistics & Applied Probab. CRC Press, 2023. ISBN 9781420076165. URL https://books.google.ca/books?id=_KnHIAAACAAJ.

[94] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[95] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[96] Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning*, pages 2029–2037. PMLR, 2018.

[97] Ziniu Hu, Zhe Zhao, Xinyang Yi, Tiansheng Yao, Lichan Hong, Yizhou Sun, and Ed H Chi. Improving multi-task generalization via regularizing spurious correlation. 2022.

[98] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer. *arXiv preprint arXiv:1809.04281*, 2018.

[99] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[100] Maximilian Ilse, Jakub M Tomczak, and Patrick Forré. Selecting data augmentation for simulating interventions. In *International Conference on Machine Learning*, pages 4555–4562. PMLR, 2021.

[101] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.

[102] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *VLDB*, volume 4, pages 180–191. Toronto, Canada, 2004.

[103] Gary King and Langche Zeng. Logistic regression in rare events data. *Political analysis*, 9(2):137–163, 2001.

[104] Simon King. Measuring a decade of progress in text-to-speech. *Loquens*, 1(1), 1 2014. ISSN 2386-2637.

[105] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[106] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2014.

[107] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International*

Conference on Machine Learning, pages 5637–5664. PMLR, 2021.

[108] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[109] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). *CoRR*, 2020.

[110] Jeremy Lai, Faruk Ahmed, Supriya Vijay, Tiam Jaroensri, Jessica Loo, Saurabh Vyawahare, Saloni Agarwal, Fayaz Jamil, Yossi Matias, Greg S Corrado, et al. Domain-specific optimization and diverse evaluation of self-supervised models for histopathology. *arXiv preprint arXiv:2310.13259*, 2023.

[111] Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *ICML*, 2018.

[112] Aleksandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of the 2003 SIAM international conference on data mining*, pages 25–36. SIAM, 2003.

[113] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436–444, 2015.

[114] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *ICLR*, 2018.

[115] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[116] H. Li, S. J. Pan, S. Wang, and A. C. Kot. Domain generalization with adversarial feature learning. pages 5400–5409, 2018.

[117] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3): 50–60, 2020.

[118] Y. Li, X. Tian, M. Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao. Deep domain generalization via conditional invariant adversarial networks. In *ECCV*, 2018.

[119] Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. Domain generalization via conditional invariant representations. 2018.

[120] Yitong Li, Michael Murias, Samantha Major, Geraldine Dawson, and David Carlson. On target shift in adversarial domain adaptation. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 616–625. PMLR, 2019.

[121] Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. In *Advances in Neural Information*

Processing Systems 32, pages 11674–11685. 2019.

[122] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020.

[123] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution detection in neural networks. *ICLR*, 2018.

[124] Tsung-Yi Lin, M. Maire, Serge J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. *ArXiv*, abs/1405.0312, 2014.

[125] R. Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

[126] Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3122–3130. PMLR, 10–15 Jul 2018.

[127] Si Liu, Risheek Garrepalli, Thomas G Dietterich, Alan Fern, and Dan Hendrycks. Open category detection with pac guarantees. *arXiv preprint arXiv:1808.00529*, 2018.

[128] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.

[129] Gary Marcus. *The algebraic mind: Integrating connectionism and cognitive science*. MIT Press, 2001.

[130] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[131] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *ICLR*, 2021.

[132] Tom Mitchell. *Machine learning*. McGraw Hill, 1997.

[133] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *CoRR*, 2018.

[134] Mary M Moya and Don R Hush. Network constraints and multi-objective optimization for one-class classification. *Neural networks*, 9(3):463–474, 1996.

[135] Kevin P Murphy. Conjugate bayesian analysis of the gaussian distribution. https://www.cs.ubc.ca/~murphyk/Papers/bayesGauss.pdf, 2007. [Online; accessed 19-January-2022].

[136] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? *ICLR*, 2019.

[137] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. *ECCV*, 2016.

[138] Graham Oppy and David Dowe. The Turing Test. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2021 edition, 2021.

[139] Yonatan Oren, Shiori Sagawa, Tatsunori Hashimoto, and Percy Liang. Distributionally robust language modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4218–4228, 2019.

[140] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[141] Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. Learning explanations that are hard to vary. *arXiv preprint arXiv:2009.00329*, 2020.

[142] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[143] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. *CVPR*, 2016.

[144] 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 International Conference on Computer Vision*, pages 1406–1415, 2019.

[145] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

[146] Stanislav Pidhorskyi, Ranya Almohsen, and Gianfranco Doretto. Generative probabilistic novelty detection with adversarial autoencoders. *NIPS*, 2018.

[147] Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal processing*, 99:215–249, 2014.

[148] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. *Advances in neural information processing systems*, 32, 2019.

[149] Herbert Robbins and Sutton Monro. A stochastic approximation method. *Ann. Math. Statist.*, 22(3):400–407, 09 1951.

[150] Simon Roburin, Charles Corbière, Gilles Puy, Nicolas Thome, Matthieu Aubry, Renaud Marlet, and Patrick Pérez. Take one gram of neural features, get enhanced group robustness. *arXiv preprint arXiv:2208.12625*, 2022.

[151] Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

[152] Peter J Rousseeuw and Mia Hubert. Robust statistics for outlier detection. *Wiley interdisciplinary reviews: Data mining and knowledge discovery*, 1(1):73–79, 2011.

[153] Donald B Rubin. Bayesian inference for causal effects: The role of randomization. *The Annals of statistics*, pages 34–58, 1978.

[154] David E. Rumelhart, James L. McClelland, and CORPORATE PDP Research Group, editors. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. MIT Press, Cambridge, MA, USA, 1986. ISBN 0-262-68053-X.

[155] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[156] Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Comput.*, 14(1): 21–41, jan 2002. ISSN 0899-7667.

[157] Shiori Sagawa, Pang Wei Koh, Tatsunori Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *ICLR*, 2020.

[158] Arthur L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3:210, 1959.

[159] Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. Breeds: Benchmarks for subpopulation shift. *arXiv preprint arXiv:2008.04859*, 2020.

[160] Piyapat Saranrittichai, Chaithanya Kumar Mummadi, Claudia Blaiotta, Mauricio Munoz, and Volker Fischer. Multi-attribute open set recognition. In Björn Andres, Florian Bernard, Daniel Cremers, Simone Frintrop, Bastian Goldlücke, and Ivo Ihrke, editors, *Pattern Recognition*, pages 101–115, Cham, 2022. Springer International Publishing.

[161] Chandramouli Shama Sastry and Sageev Oore. Detecting out-of-distribution examples with gram matrices. In *International Conference on Machine Learning*, pages 8491–8501. PMLR, 2020.

[162] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1757–1772, 2012.

[163] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[164] Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. *arXiv preprint arXiv:1206.6471*, 2012.

[165] John Searle. Minds, brains, and programs. *Behavioral and brain science*, 1980.

[166] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *CoRR*, 2020.

[167] Gabi Shalev, Yossi Adi, and Joseph Keshet. Out-of-distribution detection using multiple semantic label representations. *NeuRIPS*, 2018.

[168] Robik Shrestha, Kushal Kafle, and Christopher Kanan. Occamnets: Mitigating dataset bias by favoring simpler hypotheses. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 702–721, Cham, 2022. Springer Nature Switzerland.

[169] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. pages 443–450, 2016.

[170] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[171] Remi Tachet des Combes, Han Zhao, Yu-Xiang Wang, and Geoff Gordon. Domain adaptation with conditional distribution matching and generalized label shift. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[172] Tijmen Tieleman and Geoff Hinton. Lecture 6.5 - rmsprop, coursera: Neural networks for machine learning, 2012.

[173] Nenad Tomasev, Ioana Bica, Brian McWilliams, Lars Buesing, Razvan Pascanu, Charles Blundell, and Jovana Mitrovic. Pushing the limits of self-supervised resnets: Can we outperform supervised learning without labels on imagenet? *arXiv preprint arXiv:2201.05119*, 2022.

[174] Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. *A Deeper Look at Dataset Bias*, pages 37–55. 2017.

[175] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, pages 1521–1528, June 2011.

[176] A. M. Turing. I.—Computing Machinery and Intelligence. *Mind*, LIX(236):433–460, 10 1950. ISSN 0026-4423.

[177] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[178] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

[179] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[180] 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.

[181] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[182] Theo Vos et al. Global burden of 369 diseases and injuries in 204 countries and territories, 1990-2019: a systematic analysis for the global burden of disease study 2019. *Lancet*, 396(10258):1204–1222, 2020.

[183] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

[184] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615, 2016.

[185] Marco Willi, Ross T. Pitman, Anabelle W. Cardoso, Christina Locke, Alexandra Swanson, Amy Boyer, Marten Veldthuis, and Lucy Fortson. Identifying animal species in camera trap images using deep learning and citizen science. *Methods in Ecology and Evolution*, 10(1):80–91, 2019.

[186] Ruihan Wu, Chuan Guo, Yi Su, and Kilian Q Weinberger. Online adaptation to label distribution shift. *Advances in Neural Information Processing Systems*, 34, 2021.

[187] Yilun Xu and Tommi Jaakkola. Learning representations that support robust transfer of predictors. *arXiv preprint arXiv:2110.09940*, 2021.

[188] Jingkang Yang, Pengyun Wang, Dejian Zou, Zitang Zhou, Kunyuan Ding, Wenxuan Peng, Haoqi Wang, Guangyao Chen, Bo Li, Yiyou Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, Dan Hendrycks, Yixuan Li, and Ziwei Liu. Openood: Benchmarking generalized out-of-distribution detection.

[189] Christopher Yeh, Anthony Perez, Anne Driscoll, George Azzari, Zhongyi Tang, David Lobell, Stefano Ermon, and Marshall Burke. Using publicly available satellite imagery and deep learning to understand economic well-being in africa. *Nature communications*, 11(1):1–11, 2020.

[190] Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *CoRR*, 2017.

[191] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.

[192] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 325–333, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[193] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. Efficient gan-based anomaly detection. *CoRR*, abs/1802.06222, 2018.

[194] Dinghuai Zhang, Kartik Ahuja, Yilun Xu, Yisen Wang, and Aaron Courville. Can subnetwork structure be the key to out-of-distribution generalization? In *International Conference on Machine Learning*, pages 12356–12367. PMLR, 2021.

[195] Jianyu Zhang, David Lopez-Paz, and Léon Bottou. Rich feature construction for the optimization-generalization dilemma. *arXiv preprint arXiv:2203.15516*, 2022.

[196] Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, 2013.

[197] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. *CVPR*, 2017.

[198] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[199] Xiao Zhou, Yong Lin, Renjie Pi, Weizhong Zhang, Renzhe Xu, Peng Cui, and Tong Zhang. Model agnostic sample reweighting for out-of-distribution learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 27203–27221. PMLR, 17–23 Jul 2022.

[200] Xiao Zhou, Yong Lin, Renjie Pi, Weizhong Zhang, Renzhe Xu, Peng Cui, and Tong Zhang. Model agnostic sample reweighting for out-of-distribution learning. In *International Conference on Machine Learning*, pages 27203–27221. PMLR, 2022.

[201] Xiao Zhou, Yong Lin, Weizhong Zhang, and Tong Zhang. Sparse invariant risk minimization. In *International Conference on Machine Learning*, pages 27222–27244. PMLR, 2022.

[202] Bartosz Zieliński, Anna Plichta, Krzysztof Misztal, Przemysław Spurek, Monika Brzychczy-Włoch, and Dorota Ochońska. Deep learning approach to bacterial colony classification. *PLoS One*, 12(9), 2017.

Appendix A

Appendix for first article

A.1. Imagenet benchmarks

We first sorted all candidate subsets by the number of members. We then picked from among the list of top twenty subsets, with a preference for subsets that are more closely aligned with the theme of motivating practical applications we provided. We also manually inspected the data, to check for inconsistencies, and performed some pruning. For example, in the *beetle* subset, *leaf beetle* and *ladybug* appear to overlap sometimes. Finally, we settled on our choice of 5 subsets. In table 1, we list the members under every proposed subset. The sets are collected by first resizing such that the shorter side is of length 256 pixels, followed by a center crop. We treat 20% of the data in the training sets as validation, and the remaining 80% for training.

A.2. Experiments with CPC

CPC involves performing predictions for encodings of patches of an image from those above them. To avoid learning trivial codes, a contrastive loss is used which essentially trains the model to distinguish between correct codes and "noisy" ones. These negative samples are taken from patches within and across images in the batch.

We use the same network architecture as we used for the Imagenet experiments with rotation-prediction as the auxiliary task, but modify the first convolution layer to have a stride of 2. This reduces the computation overhead sufficiently for concurrent training with CPC

TABLE 1. Imagenet subset members

Dog (hound)	Car	Snake (colubrid)	Spider	Fungus
Ibizan hound	Model T	ringneck snake	tarantula	stinkhorn
bluetick	race car	vine snake	Argiope aurantia	bolete
beagle	sports car	hognose snake	barn spider	hen-of-the-woods
Afghan hound	minivan	thunder snake	black widow	earthstar
Weimaraner	ambulance	garter snake	garden spider	gyromitra
Saluki	cab	king snake	wolf spider	coral fungus
redbone	beach wagon	night snake		
otterhound	jeep	green snake		
Norweigian elkhound	convertible	water snake		
basset hound	limo			
Scottish deerhound				
bloodhound				

at reasonable batch-sizes (CPC training batch-sizes are 32), but at a minor expense of classifier performance. We use the first three blocks of the network for the patch encoder as in van den Oord et al. (178), and append the final layers for the classification task. Unlike with rotation, the auxiliary task works on patches while the primary classifier works on the entire image. This leads to differences in the operating receptive-fields, and differing proportions of boundary effects. To facilitate easier parameter sharing across the two tasks, we make the following changes. First, we replace all default zero-padding with reflected, symmetric padding. This removes the effect of having a different ratio of border-zeros to pixels when the spatial dimensions of the input changes. Second, we replace all normalization layers with conditional normalization variants (this means separate sets of scale and shift parameters are used depending on the current prediction task). Since batch-normalization allows trivial solutions to CPC for patches sampled from different images, we only use patches from within the same image, and find that we can continue using it to our advantage. We keep the same optimizer settings from the rotation experiments, but it is possible that different choices might lead to further improvements. λ is tuned to 10.0 for all experiments, following a coarse hyperparameter search for best validation-set classification accuracy over a range of $\{0.1, 1.0, 10.0, 20.0, 50.0\}$.

A.3. Trivial baseline for existing benchmarks

To demonstrate that the current benchmarks are trivial with very low-level information, we experiment with CIFAR-10 as in-distribution by simply looking at likelihoods under a mixture of 3 pixel-level Gaussians, trained channel-wise. We find that this simple baseline

compares very well with recent approaches at all but one of the benchmark OOD tasks in Liang et al. (123) for CIFAR-10.

OOD dataset	Average precision
TinyImagenet (crop)	96.84
TinyImagenet (resize)	99.03
LSUN	58.06
LSUN (resize)	99.77
iSUN	99.21

We see that this underperforms on LSUN. When we inspect LSUN, we find that the images are cropped patches from scene-images, and a majority of them are of uniform colour and texture, with little variation and structure in them. While this dataset is most obviously different from CIFAR-10, we believe that the appearance of the images results in the phenomenon reported in Nalisnick et al. (136), where one distribution that "sits inside" the other because of a similar mean but lower variance ends up being more likely under the wider distribution. In fact, thresholding on simply the "energy" of the edge-detection map gives us an average precision of around 87.5% for LSUN, thus indicating that the extremely trivial feature of a lower edge-count is already a strong indicator for telling apart such an obvious difference.

We found that the Gaussian baseline underperforms severely on the hold-out-class experiments on CIFAR-10, achieving an average precision of a mere 11.17% across the 10 experiments.

Appendix B

Appendix for second article

B.1. Dataset details

In this section, we provide more details about how we constructed our synthetic datasets.

B.1.1. Coloured MNIST

The training set, T_r, is constructed with an 80% colour-digit correlation per digit with nine RGB-colours (with the zero digit held out, for testing semantic anomaly detection).

1	(0,100,0)
2	(188, 143, 143)
3	(255, 0, 0)
4	(255, 215, 0)
5	(0, 255, 0)
6	(65, 105, 225)
7	(0, 225, 225)
8	(0, 0, 255)
9	(255, 20, 147)

TABLE 1. RGB codes used to bias the digits in the majority group.

The ten colours for the minority group were picked such that their L2 distance is at least 50 units away from the biasing colours. Prior to colouring, the digits were binarised to avoid grayscale tones potentially resulting in unintentionally similar colours.

For the non-systematic validation and test sets, ten colours each were chosen such that they were at least 50 units away from all other colours.

B.1.2. COCO-on-Colours

We use the following nine categories for in-distribution objects: *boat*, *airplane*, *truck*, *dog*, *zebra*, *horse*, *bird*, *train*, and *bus*. We hold out *motorcycle* for anomaly detection experiments. For background colours, we use the same colours from the coloured MNIST experiments, and also use an 80/20 split for the majority and minority groups.

In case of multiple instances of the same object in an image, we pick the largest one, and filter our dataset by mask area, such that only images with objects occupying at least 10K pixels are retained. All images are finally resized to 64×64.

The training set uses 800 such pictures per category, and the validation and test sets use 100 each. The colour backgrounds for the minority group, non-systematically shifted validation and test sets are picked using the same strategy as with the COLOURED MNIST dataset.

B.1.3. COCO-on-Places

This dataset follows the same procedure as COCO-ON-COLOURS, except using scenes from the Places dataset. In Table 2 we list the backgrounds from the corresponding scenes for the different categories.

Majority group		Minority group	Validation	Test
boat	beach	kasbah	oast house	water tower
airplane	canyon	lighthouse	orchard	waterfall
truck	building facade	pagoda	viaduct	zen garden
dog	staircase	rock arch		
zebra	desert (sand)			
horse	crevasse			
bird	bamboo forest			
train	broadleaf forest			
bus	ball pit			

TABLE 2. Background scenes for the in-distribution majority group, minority group, and the non-systematically shifted validation and test sets. (The mapping to categories only applies to the majority group in the training set.)

B.2. Network architectures and training details

B.2.1. Coloured MNIST

We use a 4-layer CNN with the first three layers being convolutional and the last layer linear. The convolutional layers have feature dimensions of $64 - 128 - 256$, and are all followed by a MAX POOL, BATCH NORM layer, and RELU activation. Before being fed into the final linear layer, there is a spatial mean-pooling operation. An L2 weight decay is added to all parameters with a co-efficient of $1e{-}4$.

Training is conducted for 30 epochs, with SGD + Momentum (0.9), using batch sizes of 512. The learning rate is cut by 10 from its initial value of 0.1 at epochs 9, 18, and 24.

B.2.2. COCO-on-backgrounds

For both COCO datasets, we use an architecture based off of Wide Resnet 28-10 (191). Since our images are 64×64, we append an extra group of 4 residual blocks with the same layer widths as in the previous group, and use a smaller widening factor of 4 instead of 10 to avoid memory overflow (starting base dimension = 64). An L2 weight decay regulariser is applied on all parameters with a coefficient of $5e{-}4$.

We train for 200 epochs with SGD + Momentum (0.9), using batch sizes of 384, with an initial learning rate of 0.1 which is cut by 10 at the 120th, 160th, 180th, and 190th epochs. We use the initially large learning rate for longer following prior works such as Li et al. (121) that have suggested annealing schedules with longer periods of higher learning rates can improve generalisation, which we do find to help the base network. In both cases, we apply data augmentation of random crops (after symmetric padding) and random horizontal reflections.

B.2.3. Partitioning network

We use the same MLP with three hidden layers for all our partitioning networks, with dimensions $64 - 32 - 16$. We use LAYER NORM (14) and RELU activations after each layer. To avoid merely memorising hard examples, it is necessary to regularise this network, so we also apply spectral normalisation (190); this involves spectrally normalising every linear

layer, and excluding the scaling term in the layer normalisation transforms, as in Miyato et al. (133).

We use a separate network for each class, training for 100 iterations each, with the same batch size as used for training the rest of the model. We use the Adam optimiser (105) with a learning rate of $1e{-}4$. In preliminary experiments we found a shared network for all categories to also work, using conditional layer normalisation (14; 48). We didn't investigate it further for all datasets, since in general a larger number of classes in a dataset might require larger capacity in the partition predictor to account for more features, and as the number of classes go up, a number of smaller matrices can have a lower footprint than one very large matrix.

Network architecture design for the partitioning network was done only on the COLOURED MNIST dataset, with access to true oracle group labels for a smaller set of in-distribution validation images (20 per category). The same network architecture was applied for the two COCO datasets. The γ hyper-parameter was learned separately for all datasets, using the smaller sets of validation images. We find, in preliminary experiments, that using random partitionings lead to much worse performance. This suggests that, although not performed for our present study, in more realistic situations one could potentially tune these hyper-parameters by validating over classification accuracy as for the invariance penalties.

B.2.4. Invariance penalties

In all cases, we pause training of the base network after 1 epoch of training, and learn a partitioning of the training set. This learned partition is used to drop in the invariance penalties as training proceeds, and as in prior work (9; 109), we find ramping in the penalty co-efficient over a number of epochs to be useful for stable training. For IRM and REx (and conditional variants), we find it helpful to scale the ERM term down by the penalty co-efficient when the optimal validation co-efficient is greater than 1, as implemented by Arjovsky et al. (9) and Krueger et al. (109).

B.3. Review of baselines and conditional variants

We briefly review the group invariance methods we compared.

B.3.1. IRMv1

In Arjovsky et al. (9), a risk regularisation method is described in order to encourage reliance on features that obey stable correlations with the target variable across data from different environments. The regularisation consists of a gradient penalty *wrt* a dummy multiplier on the logits, with the intuition that scaling up or shrinking the logits in different environments can only result in local improvements within each environment if the classifier uses features that correlate at different levels in the different environments. The objective function is

$$\min_{\Phi:\mathcal{X}\to\mathcal{Y}} \sum_{e\in\mathcal{E}} \mathcal{R}^e(\Phi) + \lambda||\nabla_{\mu|\mu=1}\mathcal{R}^e(\mu.\Phi)||^2. \tag{B.3.1}$$

Φ comprises the predictor, which in our case is $w^\top f_\theta(x)$. μ is a dummy multiplier, fixed at 1, and $\mathbb{R}^e$ is the environment risk, corresponding to the average loss for data in a particular environment when using Φ.

For our conditional variant (cIRMv1), we stratify the gradient penalty over classes, so that the penalty is applied separately per class in each environment.

The hyper-parameters we search over for this method include the penalty co-efficient λ and the number of epochs of training over which to linearly ramp up λ to its full value.

B.3.2. REx

Krueger et al. (109) proposed a risk regularisation method that aims to directly match training risks across environments, by imposing a penalty that minimises the variance of risks across environments (V-REx).

$$\min_{\Phi:\mathcal{X}\to\mathcal{Y}} \sum_{e\in\mathcal{E}} \mathcal{R}^e(\Phi) + \lambda\text{Var}(\{\cdots,\mathcal{R}^e,\cdots\}). \tag{B.3.2}$$

For our conditional variant (cREx), we apply the variance penalty stratified by class.

The hyper-parameters we search over for this method include the penalty co-efficient λ and the number of epochs of training over which to linearly ramp up λ to its full value.

B.3.3. GroupDRO

Sagawa et al. (157) suggest an online algorithm for group-based distributionally robust optimisation, which effectively re-weights group losses as a function of their evolving magnitudes, therefore putting more emphasis on groups that fare worse through training.

For our conditional variant (cGroupDRO), we compute the group weights per class, by using the losses belonging to the classes separately in each group.

The hyper-parameters we search over for this method include the learning rate for the online group-weights and the two group adjustment hyper-parameters. Additionally, we sample equally from both groups for this method, as suggested, finding it to improve results in preliminary experiments.

B.3.4. Reweight

We learn a hyper-parameter λ on validation, such that every example in the majority group is weighted with $1/(\lambda + 1)$ (because we only want to weight the majority group down).

The hyper-parameters we search over for this method include the penalty co-efficient λ and the number of epochs of training over which to linearly ramp up λ to its full value.

B.3.5. MMD feature matching

Maximum mean discrepancy based distributional matching of features across domains has been shown to be effective for domain generalisation (116), and conditional matching of distributions (usually with adversaries, for example, in Li et al. (118)) tends to work better. We found in preliminary experiments that conditional MMD significantly outperformed the unconditional variant, so we only ran full experiments and reported results using cMMD.

The group invariance penalty looks as follows

$$||\mathbb{E}\Big[\phi(f_\theta(x_{\text{group }0}))\Big] - \mathbb{E}\Big[\phi(f_\theta(x_{\text{group }1}))\Big]||^2, \tag{B.3.3}$$

where ϕ induces a kernel function K, which in our implementation is a mixture of 3 Gaussians with bandwidths $[1, 5, 10]$, which are the recommended set of bandwidths in Li et al.

(116). Adding sharper or flatter bandwidths appeared to hurt performance in preliminary experiments.

The hyper-parameters we search over for this method include the penalty co-efficient λ and the number of epochs of training over which to linearly ramp up λ to its full value.

B.3.6. Hyper-parameter grid search ranges

In all cases, λ is searched over a range of

$$\{1e{-}4, 1e{-}3, 1e{-}2, 0.1, 0.5, 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 10000, 100000\},$$

and the number of epochs over which to linearly ramp up λ is searched over $\{1, 5, 30\}$ for MNIST and $\{1, 10, 200\}$ for COCO. For the GroupDRO methods, we search over $\{0.001.0.01, 0.1, 1.0, 10\}$ for the learning rate of the group-weights, and over $\{0, 1, 2, 3, 4, 5\}$ for the group-adjustment hyper-parameters, as recommended in Sagawa et al. (157). We also average the losses group-wise as already done in IRMv1 and REx for cMMD and PGI, except for COCO-on-Places, where we find this choice to hurt performance.

B.4. Different validation sets

In this section, we report results for all the methods we compare, when picking hyper-parameters using different validation sets, as discussed in Section 4.5.4.

We note that contrary to what one would typically do in a real-world deployment, we do not augment the training sets with the validation sets for evaluating test time performance. This is because the presence of data with systematic distributional shift at training time improves performance significantly (as observed in Table 1), and our goal here is to perform an illustrative study about the potential effectiveness of invariance methods at learning to generalise systematically.

While we could have augmented the training set with validation data when we are not using validation sets with systematic distributional shift, we follow the same protocol in these cases of not augmenting the training set, in order to keep the numbers comparable with each other across different validation schemes.

TABLE 3. Picking hyper-parameters only using a validation set of non-systematic shifts for COLOURED MNIST.

Methods	In-distribution	Non-systematic shift	Systematic shift	Anomaly detection
Base (ERM)	99.60 ± 0.02	53.26 ± 1.89	38.72 ± 2.27	7.70 ± 0.23
IRMv1	**99.61 ± 0.05**	63.80 ± 3.58	55.38 ± 1.52	10.35 ± 0.43
REx	98.95 ± 0.11	72.12 ± 1.90	71.18 ± 3.27	15.54 ± 2.05
GroupDRO	98.70 ± 0.10	71.51 ± 2.61	77.95 ± 0.65	18.26 ± 2.11
Reweight	99.06 ± 0.06	77.03 ± 1.33	83.37 ± 0.61	17.10 ± 1.11
cIRMv1	99.36 ± 0.25	65.78 ± 3.53	61.09 ± 5.30	14.16 ± 2.12
cREx	99.20 ± 0.10	73.97 ± 1.07	76.06 ± 1.71	17.62 ± 2.29
cGroupDRO	97.89 ± 0.29	73.71 ± 3.21	76.90 ± 2.55	20.73 ± 4.63
cMMD	99.40 ± 0.07	97.36 ± 0.72	**97.91 ± 0.19**	**78.14 ± 3.79**
PGI	99.31 ± 0.05	**98.21 ± 0.26**	97.54 ± 0.41	76.00 ± 4.06

TABLE 4. Picking hyper-parameters using both a validation set of non-systematic shifts and the in-distribution set for COLOURED MNIST.

Methods	In-distribution	Non-systematic shift	Systematic shift	Anomaly detection
Base (ERM)	99.60 ± 0.02	53.26 ± 1.89	38.72 ± 2.27	7.70 ± 0.23
IRMv1	**99.61 ± 0.05**	63.80 ± 3.58	55.38 ± 1.52	10.35 ± 0.43
REx	98.95 ± 0.11	72.12 ± 1.90	71.18 ± 3.27	15.54 ± 2.05
GroupDRO	98.70 ± 0.10	71.51 ± 2.61	77.95 ± 0.65	18.26 ± 2.11
Reweight	99.06 ± 0.06	77.03 ± 1.33	83.37 ± 0.61	17.10 ± 1.11
cIRMv1	99.36 ± 0.25	65.78 ± 3.53	61.09 ± 5.30	14.16 ± 2.12
cREx	99.20 ± 0.10	73.97 ± 1.07	76.06 ± 1.71	17.62 ± 2.29
cGroupDRO	97.89 ± 0.29	73.71 ± 3.21	76.90 ± 2.55	20.73 ± 4.63
cMMD	99.49 ± 0.04	96.36 ± 0.53	**97.68 ± 0.17**	71.15 ± 2.65
PGI	99.30 ± 0.07	**98.31 ± 0.27**	97.48 ± 0.45	**76.07 ± 5.67**

TABLE 5. Picking hyper-parameters using only the in-distribution set for COLOURED MNIST.

Methods	In-distribution	Non-systematic shift	Systematic shift	Anomaly detection
Base (ERM)	99.60 ± 0.02	53.26 ± 1.89	38.72 ± 2.27	7.70 ± 0.23
IRMv1	99.69 ± 0.02	60.18 ± 1.34	53.20 ± 1.44	9.71 ± 0.76
REx	**99.71 ± 0.04**	60.71 ± 1.38	50.87 ± 2.79	10.02 ± 0.69
GroupDRO	99.61 ± 0.01	52.21 ± 2.03	40.27 ± 2.08	7.37 ± 0.44
Reweight	99.66 ± 0.04	63.36 ± 4.60	58.09 ± 0.52	11.41 ± 0.49
cIRMv1	99.69 ± 0.01	60.43 ± 2.71	52.98 ± 2.14	10.40 ± 0.91
cREx	99.70 ± 0.02	61.06 ± 1.20	50.83 ± 2.33	9.21 ± 0.97
cGroupDRO	99.63 ± 0.01	55.53 ± 3.63	45.25 ± 2.24	8.69 ± 1.02
cMMD	99.70 ± 0.02	61.10 ± 1.66	51.06 ± 1.87	9.62 ± 1.09
PGI	99.69 ± 0.03	**63.62 ± 2.05**	**58.18 ± 2.05**	**11.81 ± 1.89**

TABLE 6. Picking hyper-parameters only using a validation set of non-systematic shifts for COCO-ON-COLOURS.

Methods	In-distribution	Non-systematic shift	Systematic shift	Anomaly detection
Base (ERM)	90.57 ± 1.28	26.81 ± 4.93	1.10 ± 0.36	5.47 ± 0.08
IRMv1	91.61 ± 0.38	32.30 ± 4.52	2.11 ± 0.30	5.81 ± 0.17
REx	**91.69 ± 0.50**	36.57 ± 4.03	2.69 ± 0.81	5.73 ± 0.14
GroupDRO	40.31 ± 2.11	38.84 ± 3.78	**43.24 ± 2.84**	17.99 ± 3.68
Reweight	73.17 ± 2.48	48.98 ± 2.65	39.80 ± 2.61	18.20 ± 3.80
cIRMv1	91.53 ± 0.31	31.11 ± 4.51	1.74 ± 0.40	5.87 ± 0.16
cREx	91.45 ± 0.39	32.43 ± 2.03	1.98 ± 0.68	5.75 ± 0.13
cGroupDRO	43.61 ± 4.33	39.15 ± 4.79	36.63 ± 4.81	**18.21 ± 3.65**
cMMD	89.87 ± 1.13	**55.02 ± 2.29**	27.36 ± 1.57	8.82 ± 0.70
PGI	85.78 ± 1.45	51.02 ± 2.32	38.85 ± 2.29	15.71 ± 3.25

TABLE 7. Picking hyper-parameters using both a validation set of non-systematic shifts and the in-distribution set for COCO-ON-COLOURS.

Methods	In-distribution	Non-systematic shift	Systematic shift	Anomaly detection
Base	90.57 ± 1.28	26.81 ± 4.93	1.10 ± 0.36	5.47 ± 0.08
IRMv1	91.61 ± 0.38	32.30 ± 4.52	2.11 ± 0.30	5.81 ± 0.17
REx	**91.69 ± 0.50**	36.57 ± 4.03	2.69 ± 0.81	5.73 ± 0.14
GroupDRO	90.70 ± 0.56	33.10 ± 3.26	5.66 ± 0.95	6.60 ± 0.40
Reweight	90.25 ± 0.71	40.23 ± 3.32	10.60 ± 1.34	7.06 ± 0.52
cIRMv1	91.53 ± 0.31	31.11 ± 4.51	1.74 ± 0.40	5.87 ± 0.16
cREx	91.45 ± 0.39	32.43 ± 2.03	1.98 ± 0.68	5.75 ± 0.13
cGroupDRO	87.68 ± 0.59	36.40 ± 2.30	14.07 ± 2.47	9.82 ± 0.91
cMMD	89.87 ± 1.13	**55.02 ± 2.29**	27.36 ± 1.57	8.82 ± 0.70
PGI	85.78 ± 1.45	51.02 ± 2.32	**38.85 ± 2.29**	**15.71 ± 3.25**

TABLE 8. Picking hyper-parameters using only the in-distribution set for COCO-ON-COLOURS.

Methods	In-distribution	Non-systematic shift	Systematic shift	Anomaly detection
Base	90.57 ± 1.28	26.81 ± 4.93	1.10 ± 0.36	5.47 ± 0.08
IRMv1	91.54 ± 0.37	32.40 ± 3.62	1.93 ± 0.36	5.77 ± 0.23
REx	91.62 ± 0.38	31.89 ± 4.08	1.98 ± 0.37	5.74 ± 0.20
GroupDRO	91.44 ± 0.27	22.42 ± 3.00	0.56 ± 0.15	5.55 ± 0.19
Reweight	91.10 ± 0.50	38.63 ± 3.23	4.35 ± 1.13	**6.13 ± 0.22**
cIRMv1	91.31 ± 0.43	30.94 ± 3.73	1.65 ± 0.36	5.83 ± 0.17
cREx	91.70 ± 0.50	34.93 ± 4.58	2.24 ± 0.48	5.82 ± 0.19
cGroupDRO	91.75 ± 0.60	24.05 ± 3.44	0.94 ± 0.27	5.77 ± 0.13
cMMD	92.51 ± 0.41	**44.59 ± 3.28**	**10.48 ± 0.98**	6.05 ± 0.23
PGI	**91.86 ± 0.33**	32.46 ± 3.06	2.81 ± 0.53	5.88 ± 0.19

TABLE 9. Picking hyper-parameters only using a validation set of non-systematic shifts for COCO-ON-PLACES.

Methods	In-distribution	Non-systematic shift	Systematic shift	Anomaly detection
Base (ERM)	81.06 ± 1.01	45.25 ± 0.96	29.18 ± 1.24	9.21 ± 0.21
IRMv1	80.93 ± 0.71	45.17 ± 0.92	28.78 ± 0.73	9.39 ± 0.60
REx	81.25 ± 0.76	45.40 ± 0.95	29.20 ± 1.28	9.46 ± 0.98
GroupDRO	76.05 ± 0.87	43.72 ± 0.43	31.83 ± 0.54	9.61 ± 0.55
Reweight	80.90 ± 0.50	44.87 ± 1.26	29.34 ± 0.99	9.59 ± 0.54
cIRMv1	81.48 ± 0.67	45.59 ± 1.27	29.28 ± 0.96	9.80 ± 0.78
cREx	**81.50 ± 0.76**	45.44 ± 0.96	29.12 ± 0.97	9.17 ± 0.59
cGroupDRO	78.25 ± 0.31	41.69 ± 0.08	28.16 ± 0.91	9.45 ± 0.22
cMMD	79.64 ± 0.73	**49.44 ± 0.99**	**35.86 ± 0.66**	**9.80 ± 0.45**
PGI	80.99 ± 0.52	47.63 ± 0.90	31.91 ± 0.89	9.59 ± 0.89
cMMD (oracle split)	**80.04 ± 1.01**	**49.02 ± 1.18**	35.60 ± 0.72	10.55 ± 0.55
PGI (oracle split)	75.98 ± 0.75	47.50 ± 0.87	**37.27 ± 1.40**	**11.57 ± 0.71**

TABLE 10. Picking hyper-parameters using both a validation set of non-systematic shifts and the in-distribution set for COCO-ON-PLACES.

Methods	In-distribution	Non-systematic shift	Systematic shift	Anomaly detection
Base (ERM)	81.06 ± 1.01	45.25 ± 0.96	29.18 ± 1.24	9.21 ± 0.21
IRMv1	80.93 ± 0.71	45.17 ± 0.92	28.78 ± 0.73	9.39 ± 0.60
REx	81.25 ± 0.76	45.40 ± 0.95	29.20 ± 1.28	9.46 ± 0.98
GroupDRO	80.61 ± 0.44	41.96 ± 1.00	27.19 ± 0.67	9.05 ± 0.06
Reweight	80.90 ± 0.50	44.87 ± 1.26	29.34 ± 0.99	9.59 ± 0.54
cIRMv1	81.48 ± 0.67	45.59 ± 1.27	29.28 ± 0.96	9.80 ± 0.78
cREx	**81.50 ± 0.76**	45.44 ± 0.96	29.12 ± 0.97	9.17 ± 0.59
cGroupDRO	78.25 ± 0.31	41.69 ± 0.08	28.16 ± 0.91	9.45 ± 0.22
cMMD	79.64 ± 0.73	**49.44 ± 0.99**	**35.86 ± 0.66**	**9.80 ± 0.45**
PGI	80.99 ± 0.52	47.63 ± 0.90	31.91 ± 0.89	9.59 ± 0.89
cMMD (oracle split)	**79.56 ± 0.64**	46.74 ± 0.83	**34.78 ± 0.76**	9.78 ± 0.59
PGI (oracle split)	78.70 ± 0.86	**47.28 ± 1.05**	32.84 ± 0.89	**11.13 ± 0.90**

TABLE 11. Picking hyper-parameters using only the in-distribution set for COCO-ON-PLACES.

Methods	In-distribution	Non-systematic shift	Systematic shift	Anomaly detection
Base (ERM)	81.06 ± 1.01	45.25 ± 0.96	29.18 ± 1.24	9.21 ± 0.21
IRMv1	80.93 ± 0.71	45.17 ± 0.92	28.78 ± 0.73	9.39 ± 0.60
REx	81.25 ± 0.76	45.40 ± 0.95	29.20 ± 1.28	9.46 ± 0.98
GroupDRO	80.61 ± 0.44	41.96 ± 1.00	27.19 ± 0.67	9.05 ± 0.06
Reweight	**81.53 ± 0.66**	45.77 ± 1.33	29.39 ± 0.97	9.55 ± 0.79
cIRMv1	81.48 ± 0.67	45.59 ± 1.27	29.28 ± 0.96	9.80 ± 0.78
cREx	80.68 ± 0.69	44.80 ± 1.39	29.76 ± 1.05	**9.95 ± 0.79**
cGroupDRO	80.23 ± 0.13	41.86 ± 0.60	25.88 ± 1.20	9.43 ± 0.68
cMMD	81.11 ± 0.51	46.57 ± 0.97	31.54 ± 0.88	9.79 ± 0.79
PGI	80.99 ± 0.52	**47.63 ± 0.90**	**31.91 ± 0.89**	9.59 ± 0.89
cMMD (oracle split)	**81.59 ± 0.65**	**45.47 ± 1.40**	29.16 ± 0.96	9.15 ± 0.36
PGI (oracle split)	81.22 ± 1.09	45.16 ± 0.96	**29.24 ± 0.64**	**9.31 ± 0.67**

B.5. Measuring semantic anomaly detection

We use the test set with non-systematic distributional shift as the normal data, and the held-out class data combined systematically with the biasing colours or backgrounds as the anomalous data. For MNIST, this means there is 9 times more normal data than anomalous data, which reflects the typical situation of anomalies being rarer. Our choice of normal data makes this a harder task than usual, since we are assessing for higher (than the anomalies) predictive confidences for non-semantic shift with semantic factors kept the same, and reduced predictive confidence for semantic shift with non-semantic factors from the seen data. For the COCO datasets, we only sample 100 images from the held-out class to resemble the MNIST experimental setup.

Anomaly detection is measured using average precision, treating the anomalous class as positive, with the negative of the predictive softmax confidence as the score (90).

B.6. Algorithm

Algorithm 1: Algorithm for PGI

Initialise all classifier parameters θ, w and partition-predicting networks, g_c, $\forall c \in [C]$;
for *one epoch* **do**
  for *mini-batches* $\mathcal{D}_b \in \mathcal{D}$ **do**
    $grad_\theta := \nabla_\theta \ell(\theta, w|\mathcal{D}_b)$;
    $grad_w := \nabla_w \ell(\theta, w|\mathcal{D}_b)$;
    $\theta, w := optimizer(grad_\theta, grad_w)$;
  end
end
for *all classes* $c \in [C]$ **do**
  Learn a partition for images in $\mathcal{D}$ with labels c, (Eq. 8)
end
for $T - 1$ *epochs* **do**
  for *mini-batches* $\mathcal{D}_b \in \mathcal{D}$ **do**
    $grad_\theta := \nabla_\theta(\ell(\theta, w|\mathcal{D}_b) + \lambda.penalty)$ (Eq. 6,7) ;
    $grad_w := \nabla_w \ell(\theta, w|\mathcal{D}_b)$;
    $\theta, w := optimizer(grad_\theta, grad_w)$;
  end
end

Appendix C

Appendix for third article

C.1. Posterior update

We derive the posterior update equation (Eq. 5.4.5), specifying the conditions under which this rule holds. The key assumption is that in the new deployment location, categories are encountered in an IID manner in the location, i.e., $y_j \perp\!\!\!\perp y_k$.

$$
\begin{aligned}
P_t(\phi) &= P(\phi \mid y_1, \cdots, y_t), && && \text{(C.1.1)}\\
&= \frac{P(y_1, \cdots, y_t \mid \phi)\, P(\phi)}{\mathbb{P}(y_1, \cdots, y_t)}, && \text{(Bayes rule)} && \text{(C.1.2)}\\
&\propto P(y_1, \cdots, y_t \mid \phi)\, P(\phi), && \text{(dropping terms independent of } \phi) && \text{(C.1.3)}\\
&= \prod_{i=1}^{t} P(y_i \mid \phi)\, P(\phi) && \text{(using assumption } y_j \perp\!\!\!\perp y_k) && \text{(C.1.4)}\\
&= P(y_t \mid \phi) \left(\prod_{i=1}^{t-1} P(y_i \mid \phi)\, P(\phi) \right) && \text{(regrouping terms)} && \text{(C.1.5)}\\
&\propto P(y_t \mid \phi)\, P_{t-1}(\phi), && \text{(by definition)} && \text{(C.1.6)}
\end{aligned}
$$

C.2. Regression model

C.2.1. Finding the optimal solution from the predictive rule

The required distributions are defined as

$$P(y \mid x) \propto \exp\left(-\frac{\lambda_x}{2}\Big(y - f(x)\Big)^2\right), \tag{C.2.1}$$

$$P^{\text{new}}(y) \propto \left(1 + \frac{L}{2a}(y-\mu)^2\right)^{-\frac{2a+1}{2}}, \tag{C.2.2}$$

$$P(y) \propto \exp\left(-\frac{\lambda_y}{2}\Big(y - m\Big)^2\right), \tag{C.2.3}$$

$$\tag{C.2.4}$$

which gives us the objective $J = -\log P(y \mid x)$ expressed as

$$J = -\log P(y \mid x) - \log P^{\text{new}}(y) + \log P(y) \tag{C.2.5}$$

$$= \frac{\lambda_x}{2}\Big(y - f(x)\Big)^2 - \frac{\lambda_y}{2}\Big(y - m\Big)^2 + \frac{2a+1}{2}\log\left(1 + \frac{L}{2a}(y-\mu)^2\right) \tag{C.2.6}$$

The derivative of this objective wrt y is

$$\frac{\partial J}{\partial y} = \lambda_x(y - f(x)) - \lambda_y(y-m) + \frac{\frac{2a+1}{2}\frac{L}{2a}.2.(y-\mu)}{1 + \frac{L}{2a}(y-\mu)^2} \tag{C.2.7}$$

$$= \lambda_x(y - f(x)) - \lambda_y(y-m) + \frac{(2a+1)\frac{L}{2a}(y-\mu)}{1 + \frac{L}{2a}(y-\mu)^2} \tag{C.2.8}$$

$$= \underbrace{\Big(\lambda_x - \lambda_y\Big)}_{\tau_d} y + \underbrace{\Big(\lambda_y m - \lambda_x f(x)\Big)}_{\tau_\mu} + \frac{\overbrace{(2a+1)}^{A}\overbrace{\frac{L}{2a}}^{M}(y-\mu)}{1 + \frac{L}{2a}(y-\mu)^2} \tag{C.2.9}$$

$$= \tau_d y + \tau_\mu + \frac{AM(y-\mu)}{1 + M(y-\mu)^2} \tag{C.2.10}$$

Setting to zero, we have

$$\Big(\tau_d y + \tau_\mu\Big)\Big(1 + M(y-\mu)^2\Big) + AM(y-\mu) = 0 \tag{C.2.11}$$

$$\implies \Big(\tau_d y + \tau_\mu\Big)\Big(1 + My^2 + M\mu^2 - 2M\mu y\Big) + AM(y-\mu) = 0 \tag{C.2.12}$$

$$\implies \tau_d y + M\tau_d y^3 + M\mu^2\tau_d y - 2M\mu\tau_d y^2 + \tau_\mu + M\tau_\mu y^2 + M\tau_\mu\mu^2 - 2M\mu\tau_\mu y + AMy - AM\mu = 0 \tag{C.2.13}$$

$$\implies M\tau_d y^3 + (M\tau_\mu - 2M\mu\tau_d)y^2 + (\tau_d + M\mu^2\tau_d - 2M\mu\tau_\mu + AM)y + (\tau_\mu + M\tau_\mu\mu^2 - AM\mu) = 0 \tag{C.2.14}$$

which is the equation we shall solve for y. We use NumPys polynomial solver to find roots. A cubic equation either has one real and a pair of conjugate imaginary roots, or all real roots. We test the real solutions for a positive curvature (implying local minima), and pick the minima resulting in smallest value of the objective J.

C.2.2. Second derivative test for solutions

The second derivative of J is given by

$$\tau_d - \frac{2AM^2(y-\mu)^2}{(1+M(y-\mu)^2)^2} + \frac{AM}{1+M(y-\mu)^2} \tag{C.2.15}$$

Writing $y - \mu$ as D, we have

$$\tau_d + \frac{AM}{(1+MD^2)} - \frac{2AM^2D^2}{(1+MD^2)^2} = \tau_d + \frac{AM}{1+MD^2}\Big(1 - \frac{2MD^2}{1+MD^2}\Big) = \tau_d + \frac{AM(1-MD^2)}{(1+MD^2)^2} \tag{C.2.16}$$

When this expression is positive, we have a local minima.

For the first term to be positive, we require that $\tau_d > 0$, which has a straightforward intuitive interpretation: $\tau_x > \tau_y$, i.e. output precision should be higher than marginal-adjustment precision. This is a reasonable condition which we expect to be fulfilled, since we typically expect to rely more strongly on the underlying predictive model than simply the marginal.

In the second term, AM is always non-negative, for a positive pseudo-count. The denominator is always positive. Substituting in expressions for the values after the t-th update, we have

$$MD^2 = \frac{\frac{\kappa_t}{\kappa_t+1}(y-\mu_t)^2}{\sum_{\tau=0}^{t-1}\frac{\kappa_\tau}{\kappa_\tau+1}(\hat{y}_{\tau+1}-\mu_\tau)^2}. \tag{C.2.17}$$

When this term is ≤ 1, we are guaranteed positivity (strictly speaking, τ_d provides the second term with some room for negative values, but we ignore it for simplified reasoning). This condition implies

$$(y-\mu_t)^2 \leq \frac{\kappa_t+1}{\kappa_t}\sum_{\tau=0}^{t-1}\frac{\kappa_\tau}{\kappa_\tau+1}(\hat{y}_{\tau+1}-\mu_\tau)^2, \tag{C.2.18}$$

which then implies that the following range for y allows local minima

$$\mu_t - \sqrt{\frac{\kappa_t + 1}{\kappa_t} \sum_{\tau=0}^{t-1} \frac{\kappa_\tau}{\kappa_\tau + 1} (\hat{y}_{\tau+1} - \mu_\tau)^2} \leq y \leq \mu_t + \sqrt{\frac{\kappa_t + 1}{\kappa_t} \sum_{\tau=0}^{t-1} \frac{\kappa_\tau}{\kappa_\tau + 1} (\hat{y}_{\tau+1} - \mu_\tau)^2}. \quad \text{(C.2.19)}$$

An intuitive interpretation of this condition is that valid updates are allowed within an increasing range as a function of the total observed variances up to the t-th test example. In practice, we find that validation tends to pick values for $\tau_x > \tau_y$, and that the case for no-local-minima typically does not arise for the optimal hyper-parameters in our experiments.

C.2.3. Initializing priors

For initializing priors, we might endeavour to stay unbiased, since we assume that deployment locations can have significantly different target distributions than we might anticipate from the marginal over the training set. For classification, we built this in by using a uniform pseudo-count for all classes and sources. For regression, we simulate a pseudo-count of uniform samples from the output range.

If we start with a reference prior for the Normal-Gamma distribution with parameter settings

$$\mu = ., \kappa = 0, \alpha = -0.5, \beta = 0, \quad \text{(C.2.20)}$$

then after observing a N data-points $\{y_1, \cdots, y_N\}, y_i \sim \mathcal{U}[L, H]$ (the uniformly sampled points we will simulate), the resulting posterior is

$$\mu = \frac{1}{N} \sum_{i=1}^{N} y_i, \quad \text{(C.2.21)}$$

$$\kappa = N, \quad \text{(C.2.22)}$$

$$\alpha = \frac{N - 1}{2}, \quad \text{(C.2.23)}$$

$$\beta = \frac{1}{2} \sum_{i=1}^{N} (y_i - \mu)^2. \quad \text{(C.2.24)}$$

In this view, κ corresponds to the pseudo-count (as per the interpretation of the parameters of the Normal-Gamma conjugate prior as in (135)). α is defined in terms of κ. To improve stability, we will set μ to the middle of the output range rather than actually estimate the mean of our uniform pseudo-samples. Likewise, we will set β by estimating its value as a

function of κ and using the expression for variance of a uniform distribution,

$$\mathbb{E}[\beta] = \frac{1}{2}(\kappa - 1)\text{Var}(y_i) = (\kappa - 1)\frac{(H - L)^2}{24}. \tag{C.2.25}$$

C.3. Experimental details

C.3.1. Synthetic MNIST

The splitting of digits into two sets is performed by observing mis-classification matrices after 200 iterations of training a neural network averaged across a 100 runs – digits are put into opposing sets if they tend to be confused, while also trying to keep the set-sizes balanced.

The network architecture consists of 3 CONV layers with 64, 128 and 256 channels, each followed by MAXPOOL, BATCHNORM, and RELU. After the third layer, we spatially mean-pool activations and use a linear layer to map to the logits. A weight-decay of $5e-4$ is applied on all parameters. Training is conducted for 20 epochs with batches of size 256 where training accuracy saturates to 100%. An initial learning rate of 0.1 is used, which is cut by 5 at the 6-th, 12-th and 16-th epochs.

The datapoint-counts in the train/val/test environments are as follows.

		0	1	2	3	4	5	6	7	8	9
Train	red	4889	5614	4915	38	49	4664	57	59	41	4946
	cyan	43	64	53	5063	4810	42	4894	5116	4801	42
IID validation	red	989	1052	985	7	10	904	12	8	12	949
	cyan	2	12	5	1023	973	11	955	1082	997	12
OOD validation	cyan	687	714	689	313	304	635	310	315	301	664
	red	304	350	301	717	679	280	657	775	708	297
OOD Test	cyan	980	1135	1032	0	0	892	0	0	0	1009
	red	0	0	0	1010	982	0	958	1028	974	0

C.3.2. Synthetic Gaussian

The synthetic data for this experiment is generated with the following function

$$y(x) = 10\mathcal{N}(y \mid x; \mu = -2, \sigma = 0.5) + 3\mathcal{N}(y \mid x; \mu = 2, \sigma = 0.5) + 6\mathcal{N}(y \mid x; \mu = 0, \sigma = 1)$$

Training points: Training points are sampled from two regions on the x-axis, $x \sim \mathcal{N}(-2, 0.4)$ and $x \sim \mathcal{N}(2, 0.2)$, with 250 points each.

OOD validation points: OOD validation points are sampled from $\mathcal{N}(-3.5, 0.2)$ and $\mathcal{N}(1, 0.2)$, with 250 points each.

OOD test points: OOD test points are sampled from $\mathcal{N}(0, 0.2)$ and $\mathcal{N}(3, 0.2)$, with 250 points each.

For OOD sets, the different sampling distributions correspond to different locations. For different trials, we repeat the whole experiment from scratch, sampling new training, validation, and test sets, and performing validation every time.

The network architecture is a 3 layer MLP with 128 hidden units, with BATCHNORM and RELU after hidden activations. A weight decay of $1e-8$ is applied on all parameters. We train for a 100 epochs with batch-sizes of 100, with SGD + Momentum (0.9), starting with an initial learning rate of 0.01 and scaling it by 0.95 after every epoch.

We include the non-aggregated MSEs below to confirm that there are consistent improvements over every base model/data-sampling individually.

Seed	IID-Base	OOD-Base	OOD-Online
0	0.08	11.23	3.14
1	0.13	12.37	3.82
2	0.16	6.13	3.00
3	0.19	9.14	5.50
4	0.21	7.00	6.31

C.3.3. Synthetic Skewed-COCO-on-Places

We chose the following objects for this synthetic classification task: *bicycle*, *train*, *cat*, *chair*, *horse*, *motorcycle*, *bus*, *dog*, *couch*, and *zebra*; and the following scenes to simulate different sources.

Training: beach, canyon, building_facade, desert/sand, iceberg

OOD validation: oast_house, orchard, crevasse, ball_pit, viaduct

OOD test: water_tower, staircase, waterfall, bamboo_forest, zen_garden

When there are multiple instances of a class in an image, we pick the instance occupying largest area, such that only images with objects occupying at least 10K pixels are retained. All images are resized to 256×256.

Across the 5 sources, the number of examples for training, validation, and test sets are as follows.

Note that the pattern of label-shift is the same across validation and test subsets (albeit of a smaller size). This proof-of-concept experiment is intended as a middle-ground between the COLORED MNIST and WILDS-iWildCam experiments, in that the potential of learning hyper-parameters to account for conditional shift is tested while keeping label-shift pattern fixed).

We train for 400 epochs with SGD + Momentum (0.9), using batch sizes of 128, with an initial learning rate of 0.1 which is cut by 5 at the 240th, 320th, 360th epochs. An L2 weight decay regulariser is applied on all parameters with a coefficient of $5e{-}4$. We normalize images with the training set mean and standard deviation per channel, and apply data augmentation of random crops to 224×224 and random horizontal reflections.

C.4. Identity approximation for confusion matrix

Degenerate confusion matrices can arise when there are missing categories in the validation set used to compute it (leading to zero-rows), or if two or more rows are exactly the same (for example, when multiple rare categories both get categorized the same way). Two options

are to use a soft-confusion matrix, or a pseudo-inverse (126). Since the iWildCam dataset is significantly long-tailed, with a large number of classes not represented in the validation sets, we end up with a number of zero rows for the soft-confusion matrix. For such rows, we simply placed a 1 in the diagonal element.

In Table 4, we find these alternatives to result in degraded performance for iWildCam, generally much worse than our identity approximation. We hypothesize that part of the reason is to do with the fact that both our zero-confusion heuristic for dealing with missing classes for the soft-confusion matrix, as well as the same underlying effect being applied by the pseudo-inverse results in a misleading effect: rare classes, absent from validation sets, are in fact more likely to be confused than the frequent ones. This is one possibility for why the less presumptive identity approximation performs better. The inherent difficulty in estimating robust confusion matrices has been recognized in the literature, with the typical approach being to hold out significantly large validation sets in order to reliably estimate less noisy confusion matrices. In Table 5, we include numbers from an identity approximation in the synthetic datasets where the confusion matrices were invertible.

On the whole, we suggest to practitioners that in difficult, real-life situations, simpler approximations might continue to serve us well, while more sophisticated methods can pose specific requirements to be successful.

C.5. Hyperparameters, compute, and code and data licenses.

The hyper-parameters involved are the two calibration terms λ_u, λ_y and the pseudo-count term α_0 for classification, and $\lambda_x, \lambda_y, \kappa$ for the regression problems. These were picked via grid-search on the OOD validation sets, optimizing for OOD performance in all cases. For OGD methods, an additional hyper-parameter is the learning rate used for updating $\mathbf{p}$. This learning rate is searched over a range from $1e$-8 to 10 in steps of $\times 10$.

V100 GPUs were used to train base models (in cases where we trained our own models), and the online adjustment experiments were performed on an Apple Macbook Air with saved outputs from the models.

We reused code from https://github.com/p-lambda/wilds, released under the MIT License, and code from https://github.com/wrh14/online_adaption_to_label_

Table 1. Training set

	bicycle	train	cat	chair	horse	motorcycle	bus	dog	couch	zebra
beach	669	669	429	176	46	7	0	0	0	0
canyon	135	329	513	513	329	135	35	6	0	0
building_facade	5	34	132	322	503	503	322	132	34	5
desert/sand	0	0	6	35	135	329	513	513	329	135
iceberg	0	0	0	0	7	46	176	429	669	669

Table 2. Validation sets

	bicycle	train	cat	chair	horse	motorcycle	bus	dog	couch	zebra
beach	167	167	107	44	11	1	0	0	0	0
canyon	33	82	128	128	82	33	8	1	0	0
building_facade	1	8	33	80	125	125	80	33	8	1
desert/sand	0	0	1	8	33	82	128	128	82	33
iceberg	0	0	0	0	1	11	44	107	167	167

Table 3. Test sets

	bicycle	train	cat	chair	horse	motorcycle	bus	dog	couch	zebra
beach	401	401	257	105	27	4	0	0	0	0
canyon	81	197	308	308	197	81	21	3	0	0
building_facade	3	20	79	193	302	302	193	79	20	3
desert/sand	0	0	3	21	81	197	308	308	197	81
iceberg	0	0	0	0	4	27	105	257	401	401

Table 4. We compare use of a soft-confusion matrix and the pseudo-inverse with our approximation with an identity matrix for iWildCam. We find that FTH performance drops strongly, and for OGD, the optimal learning rate is most often zero, leading to no differences with base performance. For OGD, we find the optimal learning rate on the test-set for all choices of confusion matrix, reporting best-case performance.

Dataset	Method	Soft confusion matrix	Pseudo-Inverse	Identity
iWildCam (avg)	FTH (C-IID)	43.41 ± 21.80	37.23 ± 19.34	71.41 ± 4.91
	FTH (C-OOD)	34.56 ± 16.71	28.20 ± 13.74	71.41 ± 4.91
	OGD (C-IID)	73.10 ± 3.26	73.29 ± 3.04	73.16 ± 3.33
	OGD (C-OOD)	73.10 ± 3.26	73.10 ± 3.26	73.17 ± 3.18
iWildCam (macro-F1)	FTH (C-IID)	22.42 ± 4.33	11.33 ± 0.26	29.57 ± 0.93
	FTH (C-OOD)	23.73 ± 3.36	10.82 ± 4.64	29.57 ± 0.93
	OGD (C-IID)	32.71 ± 0.18	32.70 ± 0.16	32.75 ± 0.17
	OGD (C-OOD)	32.71 ± 0.14	32.70 ± 0.16	32.70 ± 0.16

TABLE 5. Identity approximation with S-MNIST and S-COCO-ON-PLACES, with test-time performance using the original confusion matrix C_f for reference. When using the identity approximation, OGD (IID) uses the IID validation set to estimate C_g and OGD (OOD) uses the OOD validation set.

Dataset	Method	Identity approximation	Original
S-MNIST	FTH	96.02 ± 1.07	96.04 ± 1.03
	OGD (IID)	89.47 ± 1.96	88.32 ± 2.06
	OGD (OOD)	95.70 ± 0.68	95.75 ± 0.70
S-COCO-ON-PLACES	FTH	59.27 ± 0.64	58.94 ± 0.63
	OGD (IID)	57.48 ± 0.52	57.37 ± 0.51
	OGD (OOD)	56.02 ± 0.35	57.75 ± 0.29

distribution_shift, publicly released by Wu et al. (186). We also used data from MS-COCO, released under the CREATIVE COMMONS ATTRIBUTION 4.0 LICENSE. WILDS-iWILDCAM is under COMMUNITY DATA LICENSE AGREEMENT – PERMISSIVE – V1.0, and the WILDS-POVERTYMAP data is U.S. PUBLIC DOMAIN (LANDSAT/DMSP/VIIRS).