Learning to Adapt to Group Distribution Shift

Anonymous Author(s)
Affiliation
Address
email

Abstract

A fundamental assumption of most machine learning algorithms is that the training and test data are drawn from the same underlying distribution. However, this assumption is violated in almost all practical applications: machine learning systems are regularly tested under *distribution shift*, due to changing temporal correlations, atypical end users, or other factors. In this work, we consider the problem setting of group distribution shift, where the training data are structured into groups and there may be multiple test time shifts, corresponding to new groups or group distributions. Most prior methods aim to learn a single robust model or invariant feature space that performs well on all groups. In contrast, we aim to learn models that *adapt* at test time to group shift using unlabeled test points. Our primary contribution is to introduce the framework of adaptive risk minimization (ARM), in which models are directly optimized for effective adaptation to shift by learning to adapt on the training groups. Compared to prior methods for robustness, invariance, and adaptation, ARM methods provide consistent gains of 1-4% test accuracy on a number of image classification problems exhibiting group shift.

1 Introduction

The standard assumption in empirical risk minimization (ERM) is that the data distribution at test time will match the training distribution. When this assumption does not hold, i.e., when there is *distribution shift*, the performance of standard ERM methods can deteriorate significantly [50, 34]. As an example which we will use throughout the paper, consider a handwriting classification model that, after training on data from past users, is deployed to new end users. Each new user represents a new test distribution that differs from the training distribution. Thus, each test setting involves dealing with shift, and users with atypical handwriting styles induce large shifts that can deteriorate model performance. How can we move past ERM in order to better handle distribution shifts such as this?

Any framework that aims to address this question must use additional assumptions beyond the ERM setting, and many such frameworks have been proposed [50]. One commonly used assumption within several frameworks, such as domain generalization [5, 20], is that the training data are provided in *groups* and distributions at test time will represent new groups (or new group distributions). The example above neatly fits this description if we equate users with groups – we would be assuming that the training data are organized by users and that the model will be tested separately on new users, and these are reasonable assumptions. Constructing training groups in practice is generally accomplished by using meta-data, which exists for many commonly used datasets. Thus, this group assumption is applicable for a wide range of realistic distribution shift problems (see, e.g., Koh et al. [32]).

However, prior benchmarks for this group based setting center around *invariances* – i.e., in these benchmarks, there is a consistent input-output relationship across all groups, and the goal is to learn this relationship while ignoring the spurious correlations within the groups (see, e.g., Gulrajani and Lopez-Paz [20]). Thus, prior methods aim for generalization to shifts by discovering this relationship, through techniques such as robust optimization and learning an invariant feature space [37, 3, 56].

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

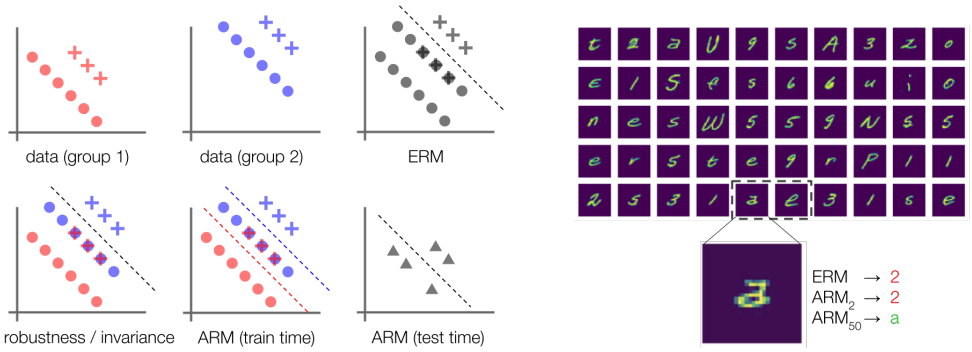

Figure 1: Best viewed in color. Left: a toy problem for which adaptation is crucial to good predictive performance, discussed in detail in Section 2. Non adaptive models are unable to deal with the ambiguous data points, whereas adaptive models trained via ARM methods can leverage group information and unlabeled test batches to succeed. Right: an example of ambiguous data points in handwriting classification, evaluated quantitatively in Section 6. A model trained via an ARM method, when given a batch of 50 unlabeled test points, is able to adapt to output the correct label "a" on the ambiguous example (shown enlarged) whereas other models incorrectly output "2".

These methods are appealing in that they make minimal assumptions about the information provided at test time – in particular, they do not require test labels, and the learned model can be immediately applied to predict on a single point. Nevertheless, these methods also have limitations, such as in dealing with problems where the input-output relationship varies across groups. We provide illustrative examples of such problems and the limitations of these methods in Section 2.

In this paper, we instead focus on methods that aim to *adapt* at test time to group shift. To do so, we study problems in which it is both feasible and helpful (and perhaps even necessary) to assume access to a batch or stream of inputs at test time. Leveraging this test assumption does not require labels for any test data and is feasible in many practical setups. For example, for handwriting classification, we do not access only single handwritten characters from an end user, but rather collections of characters such as sentences or paragraphs. Unlabeled adaptation has been shown empirically to be useful for distribution shift problems [63, 59, 68], such as for dealing with image corruptions [22]. Taking inspiration from these findings and from the examples in Section 2, we propose and evaluate on a number of group shift problems for which adaptation is beneficial, which we detail in Section 6.

The main contribution of this paper is to introduce the framework of adaptive risk minimization (ARM), which proposes the following objective: optimize the model such that it can maximally leverage the unlabeled adaptation phase to handle group distribution shift. We optimize this objective by introducing a general *meta-learning* algorithm and instantiating a set of methods that, given a set of training groups, meta-learns a model that is adaptable to these groups. Our experiments in Section 6 test on several image classification problems including benchmarks for federated learning [7] and image classifier robustness [22]. These testbeds are also a contribution of our work, as we believe these problems can supplement existing benchmarks which, almost exclusively, are designed with invariance in mind [3, 49, 20]. Empirically, we demonstrate that the proposed ARM methods, by leveraging the meta-training phase and test time adaptation, are able to consistently outperform prior methods based on robustness, invariance, and adaptation by 1-4% test accuracy.

2 Illustrative examples

Consider the toy 2D binary classification problem depicted in Figure 1 (left), where the label is indicated by a circle or cross. The training examples come from two groups (color coded red and blue), and the challenge in this problem is that some data points are ambiguous: identical points can have different labels depending on the group. ERM is depicted in the top row third panel: it does not use knowledge of the groups, thus it sees all points as "gray", and it learns a decision boundary that misclassifies half of the ambiguous training points. We may intuitively expect better performance if we use knowledge of the groups during training. However, as shown in the bottom row first panel, any group based learning method that learns a non adaptive model, i.e., a model which represents a single

decision boundary, will still misclassify half of the ambiguous training points *at best*. For example, robustness methods that optimize for worst case group performance would sacrifice correctness on group 1 in order to succeed on the majority points in group 2. Invariance methods would fare no better, as there is no feature space that can separate the ambiguous points and make them classifiable.

As shown in the last two panels in the bottom row, and as we will discuss in Section 5, ARM methods can solve this problem by learning adaptive models, which can represent different decision rules for different groups. Unlike prior methods for test time adaptation, ARM methods leverage the training groups to specifically train the model to adapt to group shift (middle panel). At test time, the model adapts by observing a batch or stream of unlabeled points (shown as triangles in the last panel). In the case of ambiguous data points, this batch or stream allows for (potentially implicit) inference of the underlying test group – and thus better performance – that is not possible from a single point.

It is natural to wonder whether this toy example is actually representative of real world problems. In Figure 1 (right), we highlight a similar example for handwriting classification, a problem setting that we evaluate quantitatively in Section 6. We visualize a batch of 50 examples from a test user, and we highlight an ambiguous example which may be either a "2" or an "a" depending on the user's handwriting. An ERM trained model and an ARM trained model, when only given two unlabeled examples for adaptation, incorrectly classify this example as "2". However, when given access to a batch of 50 images, which contain other examples of class "2" and "a" from this user, the ARM trained model successfully adapts its prediction to "a", which is the correct label. Beyond problems involving ambiguous data points, we also observe the benefits of ARM methods in a problem from the DomainBed benchmark suite [20], discussed in Appendix C, and additional problems in Section 6.

3 Related work

A number of prior works have studied distribution shift in various forms [50]. In this section, we review prior work in domain generalization, group robustness, meta-learning, and adaptation.

Invariance and robustness to groups. As discussed above, a number of frameworks leverage training groups to address test time shift. The terminology in prior work is scattered and, depending on the application, includes terms such as "domains", "datasets", "subpopulations", and "users"; in this work, we adopt the term "groups" which we believe is an appropriate unifying term. A number of testbeds for this problem setting have been proposed for image classification, including generalizing to new datasets [14], new image types [35, 49], and underrepresented demographics [56].

Prior benchmarks typically assume the existence of a consistent input-output relationship across groups that is learnable by the specified model, thus motivating methods such as learning an invariant feature space [37, 41, 3] or optimizing for worst case group performance [27, 56]. Gulrajani and Lopez-Paz [20] provide a comprehensive survey of many of these benchmarks and find that, surprisingly, ERM is competitive with the state of the art across all the benchmarks considered. In Appendix C, we discuss this finding as well as the performance of an ARM method on this benchmark suite. In Section 6, we identify different problems for which adaptation is helpful, and we find that ARM methods consistently outperform ERM and other non adaptive group based learning methods.

Meta-learning. Meta-learning [58, 4, 64, 24] has been most extensively studied in the context of few shot *labeled* adaptation [57, 67, 51, 15, 61]. Our aim is not to address few-shot recognition problems, nor to propose a novel meta-learning algorithm, but rather to extend meta-learning paradigms to problems requiring unlabeled adaptation, with the primary goal of tackling distribution shift. This aim differs from previous work in meta-learning for domain generalization [36, 12], which seek to meta-train models for non adaptive generalization performance. We discuss in Section 5 how paradigms such as contextual meta-learning [17, 53] can be readily extended using the ARM framework.

Some other meta-learning methods adapt using both labeled and unlabeled data, either in the semi supervised learning setting [52, 72, 39] or the transductive learning setting [47, 43, 2, 26]. These works all assume access to labeled data for adaptation, whereas we propose methods and problems for purely unlabeled adaptation. Prior works in meta-learning for unlabeled adaptation include Yu et al. [71], who adapt a policy to imitate human demonstrations in the context of robotic learning; Metz et al. [45], who meta-learn an update rule for unsupervised representation learning, though they still require labels to learn a predictive model; and Alet et al. [1], who meta-learn adaptive models based on task specific unsupervised objectives. Unlike these prior works, we propose a general framework

for tackling distribution shift problems by meta-learning unsupervised adaptation strategies. This framework simplifies the extension of meta-learning paradigms to these problems, encapsulates previous approaches such as the gradient based meta-learning approach of Yu et al. [71], and sheds light on how to improve existing strategies such as adaptation via batch normalization [40].

Adaptation to shift. Unlabeled adaptation has primarily been studied separately from meta-learning. Domain adaptation is a prominent framework that assumes access to test examples at *training* time [10, 69], similar to transductive learning [66]. As such, most domain adaptation methods consider the problem setting where there is a single test distribution [60, 11, 19, 16, 65, 8], and some of these methods are difficult to apply to problems where there are multiple test distributions. Certain domain adaptation methods have also been applied in the domain generalization setting, such as methods for learning invariant features [16, 37], and we compare to these methods in Section 6.

Blanchard et al. [5] and Blanchard et al. [6] provide a theoretic study of domain generalization and establish favorable generalization bounds for models that can adapt to group shift at test time. In comparison, our work establishes a framework that makes explicit the connection between adaptation to group shift and meta-learning, allowing us to devise new methods in a straightforward and principled manner. These methods are amenable to expressive models such as deep neural networks, which enables us to propose and evaluate on problems with raw image observations.

Test time adaptation has also been studied for dealing with label shift [55, 42, 62] and crafting favorable inductive biases for the domain of interest. For image classification, techniques such as normalizing via the test inputs [40] and optimizing self-supervised surrogate losses [63] have proven effective for adapting to image corruptions [22]. We compare to these prior methods in Section 6 and empirically demonstrate the advantage of using training groups to learn how to adapt.

4 Preliminaries and notation

In this section, we discuss the group distribution shift problem setting and formally describe adaptive models. In Section 5, we discuss how adaptive models can be meta-trained via the ARM objective and approach, and we instantiate ARM methods which we empirically evaluate in Section 6.

Let $\mathbf{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$ represent the input and output, respectively. We can formalize the group based setting using the following data generation process [6]: first, a joint data distribution $p_{\mathbf{x}y}$ is sampled from a set of distributions $\mathcal{P}_{\mathbf{x}y}$, and then some data points are sampled from $p_{\mathbf{x}y}$.[1] We refer to each $p_{\mathbf{x}y}$ as a group, e.g., a particular dataset or user, thus $\mathcal{P}_{\mathbf{x}y}$ represents the set of all possible groups. We assume that the training dataset is composed of data from S runs of this generative process, organized by group. An equivalent characterization which we will use for clarity is that, within the training set, there are S groups, and each data point $(\mathbf{x}^{(i)}, y^{(i)})$ is annotated with a group label $z^{(i)}$. Each $z^{(i)}$ is an integer that takes on a value between 1 and S, indicating which $p_{\mathbf{x}y}$ generated the i-th training point (though, of course, we cannot access $p_{\mathbf{x}y}$ itself). At test time, there may be multiple evaluation settings, where each setting is considered separately and contains only *unlabeled* data sampled via a new run of the same generative process. This data may represent, e.g., a new dataset or user, and the test groups are likely to be distinct from the training groups when $|\mathcal{P}_{\mathbf{x}y}|$ is large or infinite.

Our formal goal is to optimize for expected performance, e.g., classification accuracy, at test time. To do so, let us first consider predictive models of the form $f : \mathcal{X} \times \mathcal{P}_{\mathbf{x}} \to \mathcal{Y}$, where the model f takes in not just an input $\mathbf{x}$ but also the marginal input distribution $p_{\mathbf{x}} \in \mathcal{P}_{\mathbf{x}}$ that $\mathbf{x}$ was sampled from. We refer to f as an *adaptive* model, as it has the opportunity to use $p_{\mathbf{x}}$ to adapt its predictions on $\mathbf{x}$. The underlying assumption is that $p_{\mathbf{x}}$ provides information about $p_{y|\mathbf{x}}$, i.e., $p_{\mathbf{x}}$ is used as a surrogate input in place of $p_{\mathbf{x}y}$. In the worst case, if $p_{\mathbf{x}}$ and $p_{y|\mathbf{x}}$ are sampled independently, then the model does not benefit at all from knowing $p_{\mathbf{x}}$. In many problems, however, we expect knowledge about $p_{\mathbf{x}}$ to be useful, e.g., for resolving ambiguity as illustrated by the example problems in Section 2.

Theoretically, when $p_{\mathbf{x}}$ provides information about $p_{y|\mathbf{x}}$, and when training and test groups are drawn from the same distribution over $\mathcal{P}_{\mathbf{x}y}$, we can establish favorable generalization bounds for the expected performance of f in adapting to group shift at test time (see Theorem 6 and Lemma 9 in Blanchard et al. [6] for formal presentations of such results). Practically, we do not know and cannot input $p_{\mathbf{x}}$ to f in most cases. Instead, we instantiate f such that it takes in a batch of inputs $\mathbf{x}_1, \ldots, \mathbf{x}_K$, all from the same group, where K can vary. f makes predictions on the whole batch,

[1] Formally, the number of points sampled is another random variable with support over the positive integers.

which also serves as an empirical approximation (i.e., a histogram) $\hat{p}_{\mathbf{x}}$ of $p_{\mathbf{x}}$ [6]. In our exposition, we will assume that a batch of unlabeled points is available at test time for adaptation. However, we also experiment in Section 6 with the *streaming* setting where the test inputs are observed one at a time and adaptation occurs incrementally. Thus, the assumption of test batches is not crucial.

Notice that, if we instead passed in an approximation $\hat{p}_{\mathbf{x}y}$ of $p_{\mathbf{x}y}$ to the model, such as a batch of *labeled* data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_K, y_K)$, then this setup would resemble the standard few shot meta-learning problem [67]. Formally, a meta-learning model takes in both an input $\mathbf{x}$ and $\hat{p}_{\mathbf{x}y}$, which approximates the distribution that $\mathbf{x}$ was sampled from and thus can be used to adapt the prediction on $\mathbf{x}$. Compared to our problem setting, the meta-learning formalism can tackle a wider range of problems but also requires more restrictive assumptions, specifically, labels at test time via $\hat{p}_{\mathbf{x}y}$. *Transductive meta-learning* methods further assume that, in addition to $\hat{p}_{\mathbf{x}y}$, a full batch of inputs $\mathbf{x}_1, \ldots, \mathbf{x}_K$ is passed into the model, which allows for better estimation of the input distribution $p_{\mathbf{x}}$ [47, 43, 26]. The model then makes predictions on this entire batch. In meta-learning terminology, $\hat{p}_{\mathbf{x}y}$ and $\mathbf{x}_1, \ldots, \mathbf{x}_K$ are often referred to as the *support* and *query*, respectively. Therefore, another interpretation of the adaptive models that we study in this work is that they resemble transductive meta-learning models, but they are given only the unlabeled query and not the labeled support set.

In the next section, we expand on this connection to develop the ARM framework, which then allows us to bring forward tools from meta-learning to tackle group distribution shift problems.

5 Adaptive risk minimization

In this section, we formally describe the ARM framework, which defines an objective for training adaptive models to tackle group shift. Furthermore, we propose a general meta-learning algorithm as well as specific methods for optimizing the ARM objective. In Section 6, we test these ARM methods on problems for which unlabeled adaptation can be leveraged for better test performance.

5.1 Devising the ARM objective

We wish to learn an adaptive model $f : \mathcal{X}^K \to \mathcal{Y}^K$ to tackle group distribution shift. As noted, meta-learning methods for labeled adaptation study a similar form of model, and a common approach in many of these methods is to define f such that it is composed of two parts: first, a *learner* which ingests the data and produces parameters, and second, a *prediction model* which uses these parameters to make predictions [67, 15]. We will follow a similar strategy which, as we will discuss in subsection 5.2, allows us to easily extend and design meta-learning methods towards our goal.

In particular, we will decompose the model f into two modules: a standard *prediction model* $g(\cdot\,;\theta) : \mathcal{X} \to \mathcal{Y}$, that is parameterized by $\theta \in \Theta$ and predicts y given $\mathbf{x}$, and an *adaptation model* $h(\cdot\,,\cdot\,;\phi) : \Theta \times \mathcal{X}^K \to \Theta$, which is parameterized by ϕ. h takes in the prediction model parameters θ and K unlabeled data points and uses the K points to produce adapted parameters θ'. This is analogous to the learner in meta-learning, however, h adapts the model parameters using only unlabeled data.[2] h can be initialized as a variety of different adaptation procedures, and we defer this discussion to subsection 5.2. Our goal is to optimize ϕ and θ such that h can adapt g using unlabeled data sampled according to a particular group z. This motivates the ARM objective, given by

$$\min_{\theta,\phi} \mathbb{E}_{p_z}\left[\mathbb{E}_{p_{\mathbf{x}y|z}}\left[\frac{1}{K}\sum_{k=1}^{K}\ell(g(\mathbf{x}_k;\theta'), y_k)\right]\right], \text{ where } \theta' = h(\theta, \mathbf{x}_1, \ldots, \mathbf{x}_K; \phi)\,. \tag{1}$$

Mimicking the generative process from Section 4 that we assume generated the training data, p_z is a categorical distribution over $\{1, \ldots, S\}$ which places uniform probability mass on each training group, and $p_{\mathbf{x}y|z}$ assigns uniform probability to only the training points within a particular group. Theoretically, we expect the trained models to perform well at test time if the test groups are sampled independently and identically – i.e., from the same distribution over $\mathcal{P}_{\mathbf{x}y}$ – as the training groups. In practice, similar to how meta-learned few shot classification models are typically evaluated on new and unseen meta-test classes [67, 15], we empirically show in Section 6 that the trained models can generalize to test groups that are not sampled identically to the training groups.

[2] For some meta-learning methods, the learner does not take as input the unadapted model parameters [67], and we also devise some methods of this form. In the formalism above, these methods simply ignore the input θ.

5.2 Optimizing the ARM objective

Algorithm 1 presents a general meta-learning approach for optimizing the ARM objective. As described above, h outputs updated parameters θ' using an unlabeled batch of data (line 5). This mimics the adaptation procedure at test time, where we do not assume access to labels (lines 7-8). However, the training update itself does rely on the labels (line 6). We assume that h is differentiable with respect to its input θ and ϕ, thus we use gradient updates on both θ and ϕ to optimize for *post adaptation* performance on a mini batch of data sampled according to a particular group z. In practice, we also sample mini batches of groups, rather than just one group (as written in line 3), to provide a better gradient signal for optimizing ϕ and θ.

Algorithm 1 Meta-Learning for ARM

`// Training procedure`

Require: # training steps T, batch size K, learning rate η

1: **Initialize:** θ, ϕ
2: **for** $t = 1, \ldots, T$ **do**
3: Sample z uniformly from training groups
4: Sample $(\mathbf{x}_k, y_k) \sim p(\cdot, \cdot|z)$ for $k = 1, \ldots, K$
5: $\theta' \leftarrow h(\theta, \mathbf{x}_1, \ldots, \mathbf{x}_K; \phi)$
6: $(\theta, \phi) \leftarrow (\theta, \phi) - \eta \nabla_{(\theta,\phi)} \sum_{k=1}^{K} \ell(g(\mathbf{x}_k; \theta'), y_k)$

`// Test time adaptation procedure`

Require: θ, ϕ, test batch $\mathbf{x}_1, \ldots, \mathbf{x}_K$

7: $\theta' \leftarrow h(\theta, \mathbf{x}_1, \ldots, \mathbf{x}_K; \phi)$
8: $\hat{y}_k \leftarrow g(\mathbf{x}_k; \theta')$ for $k = 1, \ldots, K$

Together, Equation 1 and Algorithm 1 shed light on a number of ways to devise methods for solving the ARM problem. First, we can extend meta-learning paradigms to the ARM problem setting, and any paradigm in which the adaptation model h can be augmented to operate on unlabeled data is readily applicable. As an example, we propose the ARM-CML method, which is inspired by recent works in contextual meta-learning (CML) [17, 53]. Second, we can enhance prior unlabeled adaptation methods by incorporating a meta-training phase that allows the model to better leverage the adaptation. To this end, we propose the ARM-BN method, based on the general approach of adapting using batch normalization (BN) statistics of the test inputs [40, 59, 30, 46]. Third, we can incorporate existing methods for meta-learning unlabeled adaptation to solve group distribution shift problems. We demonstrate this by proposing the ARM-LL method, which is based on the robotic imitation learning method from Yu et al. [71] which adapts via a learned loss (LL). We summarize the methods here and refer the reader to Appendix B for complete details of each ARM method.

ARM-CML. In ARM-CML, the parameters ϕ of h define the weights of a *context network*, which processes each example $\mathbf{x}_k$ separately to produce vectors $\mathbf{c}_k \in \mathbb{R}^D$ for $k = 1, \ldots, K$, where D is a hyperparameter. This is different from prior contextual meta-learning approaches, in which the context network processes both inputs and outputs to produce each $\mathbf{c}_k$. The average vector $\mathbf{c} \triangleq \frac{1}{K} \sum_{k=1}^{K} \mathbf{c}_k$ is then passed to g as an additional input. In this setup, h can learn to provide useful information about the entire batch of K unlabeled points to g for predicting the correct outputs.

ARM-BN. ARM-BN is a particularly simple method that is applicable for any model g that has BN layers [28]. Practically, training g via ARM-BN follows the same protocol as Ioffe and Szegedy [28] except for two key differences: first, the training batches are sampled from a single group, rather than from the entire dataset, and second, the normalization statistics are recomputed at test time rather than using a training running average. As noted, this second difference has been explored by several works as a method for test time adaptation, but the first difference is novel to ARM-BN. Following Algorithm 1, ARM-BN defines a meta-training procedure in which g learns to adapt – i.e., compute normalization statistics – using batches of training points sampled from the same group. We empirically show in Section 6 that, for problems where BN adaptation already has a favorable inductive bias, such as for image classification, further introducing meta-training boosts its performance. We believe that other test time adaptation methods, such as those based on optimizing surrogate losses [63, 68], may similarly benefit from their corresponding meta-training procedures.

ARM-LL. ARM-LL follows the gradient based meta-learning paradigm [15] and learns parameters θ that are amenable to gradient updates on a loss function in order to quickly adapt to a new problem. In other words, h produces $\theta' = \theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathbf{x}_1, \ldots, \mathbf{x}_K; \phi)$, where α is a hyperparameter. Note that the loss function $\mathcal{L}$ used in the gradient updates is different from the original supervised loss function ℓ, in that it operates on only the inputs $\mathbf{x}$, rather than the input output pairs that ℓ receives. We follow the general implementation of Yu et al. [71], in that the adaptation model parameters ϕ define the weights of a *loss network*, which takes in the output of g and produces a scalar, and $\mathcal{L}$ is defined as the ℓ_2-norm of these scalar outputs across the batch of inputs.

6 Experiments

Our experiments are designed to answer the following:

1. Do ARM methods learn models that can leverage unlabeled adaptation to tackle group shift?
2. How do ARM methods compare to prior methods for robustness, invariance, and adaptation?
3. Can models trained via ARM methods adapt successfully in the streaming test setting?

6.1 Evaluation domains and protocol

We evaluate on four image classification problems, which we present below and describe in full detail in Appendix D. We believe that the problems we present in this paper can supplement existing benchmarks for group shift, which, as discussed above, are designed to test invariances. A key characteristic of the problems presented here is the potential for adaptation to improve test performance, and this differs from prior benchmarks such as the problems compiled by DomainBed [20]. In Appendix C, we compare our testbeds to DomainBed and group robustness benchmarks, and we briefly discuss the results in Gulrajani and Lopez-Paz [20], which also evaluate the ARM-CML method.

Rotated MNIST. We study a modified version of MNIST where images are rotated in 10 degree increments, from 0 to 130 degrees. We treat each rotation as a separate group, i.e., a different value of z. We use only 108 training data points for each of the 2 smallest groups (120 and 130 degrees), and 324 points each for rotations 90 to 110, whereas the overall training set contains 32292 points. In this setting, we hypothesize that adaptation can specialize the model to specific groups, in particular the *rare groups* in the training set. For each test evaluation, we generate images from the MNIST test set with a certain rotation. We measure both worst case and average accuracy across groups.

Federated Extended MNIST (FEMNIST). The extended MNIST (EMNIST) dataset consists of images of handwritten uppercase and lowercase letters, in addition to digits [9]. FEMNIST is the same dataset, but it also provides the meta-data of which user generated each data point [7]. We treat each user as a group. We measure each method's worst case and average accuracy across 35 test users, which are held out and thus *disjoint* from the training users. As discussed in Section 2, adaptation may help for this problem for specializing the model and resolving ambiguous data points.

Corrupted image datasets. CIFAR-10-C and Tiny ImageNet-C [22] augment the CIFAR-10 [33] and Tiny ImageNet test sets with common image corruptions that vary in type and severity. The original goal of these augmented test sets was to benchmark how well methods could handle these corruptions without access to *any* corruptions during training [22]. Thus, successful methods for these problems typically have relied on domain knowledge and heuristics designed specifically for image classification. For example, prior work has shown that carefully designed test time adaptation procedures are effective for these problems [63, 59, 68]. One possible reason for this phenomenon is that convolutional networks are biased toward texture [18], which is distorted by corruptions, thus adaptation can help the model recover its performance for each corruption type.

We study whether meta-training for adaptation performance can improve upon these results. To do so, we modify the protocol from Hendrycks and Dietterich [22] to fit into the ARM problem setting by applying a set of 56 corruptions to the training data, and we define each corruption to be a group. We use a *disjoint* set of 22 corruptions for the test data, which are mostly of different types from the training corruptions (thus, not sampled identically), and we measure worst case and average accuracy across the test corruptions. This modification allows us to study, for both ARM and prior methods, whether seeing corruptions at training time can help the model deal with new corruptions at test time.

6.2 Comparisons and ablations

We compare the ARM methods against several prior methods designed for robustness, invariance, and adaptation. We describe the comparisons here and provide additional details in Appendix D.

Group robustness and invariance. Sagawa et al. [56] recently proposed a state-of-the-art method for group robustness, and we refer to this approach as distributionally robust neural networks (DRNN). Their work also evaluates a strong upweighting (UW) baseline that samples uniformly from each group, and so we also evaluate this approach in our experiments. Additionally, we compare to domain adversarial neural networks (DANN) [16] and maximum mean discrepancy (MMD) feature learning [37], two state-of-the-art methods for adversarial learning of invariant predictive features.

Table 1: Worst case (WC) and average (Avg) top 1 accuracy on all testbeds, where means and standard errors are reported across three separate runs of each method. Horizontal lines separate methods that make use of (from top to bottom): neither, training groups, test batches, or both. ARM methods consistently achieve greater robustness, measured by WC, and Avg performance compared to prior methods. *UW is identical to ERM for CIFAR-10-C and Tiny ImageNet-C.

	MNIST		FEMNIST		CIFAR-10-C		Tiny ImageNet-C	
Method	**WC**	**Avg**	**WC**	**Avg**	**WC**	**Avg**	**WC**	**Avg**
ERM	74.3 ± 1.7	93.6 ± 0.4	62.9 ± 1.9	80.1 ± 0.9	49.6 ± 0.1	69.8 ± 0.4	19.3 ± 0.5	41.4 ± 0.2
UW*	80.2 ± 0.1	94.8 ± 0.2	61.8 ± 0.9	80.1 ± 0.3	—	—	—	—
DRNN	79.3 ± 1.1	94.8 ± 0.1	58.1 ± 0.7	74.4 ± 0.8	44.5 ± 0.5	70.7 ± 0.6	19.9 ± 0.3	41.6 ± 0.2
DANN	80.3 ± 1.4	95.1 ± 0.1	66.5 ± 0.4	82.3 ± 0.4	42.4 ± 0.2	69.1 ± 0.4	20.5 ± 0.1	41.9 ± 0.2
MMD	82.8 ± 2.0	95.6 ± 0.3	65.3 ± 1.6	81.6 ± 0.6	43.2 ± 0.4	70.4 ± 0.3	20.2 ± 0.0	39.9 ± 0.5
BN adaptation	75.1 ± 0.2	93.9 ± 0.1	66.9 ± 0.8	81.1 ± 0.3	62.5 ± 0.2	79.4 ± 0.3	23.9 ± 0.2	42.8 ± 0.2
TTT	81.1 ± 0.3	95.4 ± 0.1	64.1 ± 0.2	83.4 ± 0.1	66.6 ± 0.6	75.6 ± 0.8	19.7 ± 0.4	41.4 ± 0.3
CML ablation	78.2 ± 0.6	94.2 ± 0.1	64.4 ± 0.7	81.5 ± 0.7	47.8 ± 0.1	68.2 ± 0.1	19.6 ± 0.4	42.3 ± 0.2
LL ablation	82.4 ± 0.3	94.8 ± 0.2	61.9 ± 0.2	79.3 ± 0.6	61.5 ± 0.2	68.3 ± 0.5	25.8 ± 0.4	41.7 ± 0.1
ARM-CML	$\mathbf{88.7 \pm 0.6}$	$\mathbf{96.7 \pm 0.1}$	67.8 ± 1.3	$\mathbf{85.7 \pm 0.3}$	67.7 ± 0.5	79.2 ± 0.3	21.4 ± 0.2	43.3 ± 0.4
ARM-BN	82.8 ± 0.4	95.3 ± 0.1	$\mathbf{72.6 \pm 0.3}$	$\mathbf{85.7 \pm 0.1}$	$\mathbf{71.1 \pm 0.1}$	$\mathbf{80.9 \pm 0.2}$	$\mathbf{27.7 \pm 0.2}$	$\mathbf{44.9 \pm 0.2}$
ARM-LL	87.2 ± 0.5	96.3 ± 0.2	69.6 ± 2.1	$\mathbf{85.6 \pm 0.5}$	66.9 ± 0.2	75.7 ± 0.3	27.1 ± 0.3	44.2 ± 0.4

Test time adaptation. We evaluate the general approach of using test batches to compute BN statistics [40, 59, 30, 46], which we term BN adaptation. We also compare to test time training (TTT) [63], which adapts the model at test time using a self-supervised rotation prediction loss. These methods have previously achieved strong results for image classification, likely because they constitute favorable inductive biases for improving on the true classification task [63].

Robustness and invariance methods assume access to training groups but not test batches, whereas adaptation methods assume the opposite. Thus, at a high level, we can view the comparisons to these methods as evaluating the importance of each of these assumptions for the specified problems.

Ablations. We also include ablations of the ARM-CML and ARM-LL methods, which sample training batches of unlabeled examples uniformly from the entire training set, rather than sampling from a single group.[3] These "context ablation" and "learned loss ablation" are similar to test time adaptation methods in that they do not require training groups. However, these methods lack the inductive biases of BN adaptation and TTT, as they instead use learned context and loss networks.

6.3 Quantitative evaluation and comparisons

The results are presented in Table 1. From these results, we highlight several key takeaways:

ARM methods improve robustness and performance. Across all of our experiments, ARM methods consistently increase both worst case and average accuracy compared to all other methods. ARM-BN in particular achieves the best performance on most domains, demonstrating the effectiveness of training the model to make even greater use of an already strong adaptation procedure. The exception to this is the rotated MNIST experiment, where ARM-CML and ARM-LL outperform ARM-BN. This testbed represents a less realistic shift compared to the user and corruption shifts in the other experiments, thus we also see that BN adaptation itself provides less benefit. ARM-CML and ARM-LL use learned adaptation procedures and therefore are able to generally improve upon the other methods for almost all metrics. We suspect that these more expressive methods could perform better than ARM-BN for larger datasets or other modalities such as video sequences [71].

Prior group based learning methods perform worse on these testbeds. In our experiments, DRNN generally results in worse average case and, surprisingly, worst case performance. We see that DANN and MMD mostly outperform robustness methods, though the overall performance of these methods for learning invariant features across groups is still worse than ARM methods. As discussed, these results may be due to the differences between our problems and prior group based settings, and we provide additional details of these differences in Appendix C.

Prior test time adaptation methods generally perform well. In our evaluation, BN adaptation and TTT are the strongest prior methods, though they typically still lag behind ARM methods and

[3] Note that the corresponding ablation of ARM-BN is simply the BN adaptation method.

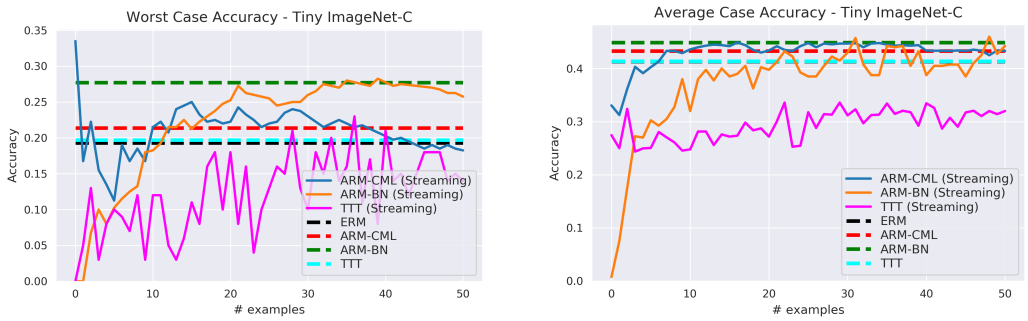

Figure 2: In the streaming setting, ARM methods reach strong performance on Tiny ImageNet-C after fewer than 25 data points, despite using training batch sizes of 50. This highlights that the trained models are able to adapt with small test batches and can operate successfully in the standard streaming evaluation setting. The streaming version of TTT is slower to adapt to test groups, likely because the model is not meta-trained to adapt quickly, and, consistent with Table 1, generally performs worse.

ARM-BN in particular. As discussed above, we believe that these adaptation procedures constitute inductive biases that are generally well suited for image classification. However, ARM methods are comparatively less reliant on favorable inductive biases and consistently attain better results.

In summary, in our experiments, we observe poor performance from robustness methods, varying performance from invariance methods, strong performance from adaptation methods, and the strongest performance from ARM methods. As ARM methods also make the strongest assumptions, we next present some findings on how the test batch assumption may be loosened.

6.4 Loosening the test batch assumption

When we cannot access a batch of test points all at once, and instead the points are observed in a streaming fashion, we can augment the proposed ARM methods to perform sequential model updates. For example, ARM-CML and ARM-BN can update their average context and normalization statistics, respectively, after observing each new test point. In Figure 2, we study this test setting for the Tiny ImageNet-C problem. We see that both models trained with ARM-CML and ARM-BN are able to achieve near their original worst case and average accuracy within observing 25 data points, well before the training batch size of 50. This result demonstrates that ARM methods are applicable for problems where test points must be observed one at a time, provided that the model is permitted to adapt using each point. We describe in detail how each ARM method can be applied to the streaming setting in Appendix B, and we provide streaming results on rotated MNIST in Appendix E.

7 Discussion and future work

We presented adaptive risk minimization (ARM), a framework and problem formulation for learning models that can adapt in the face of group distribution shift at test time using only a batch of unlabeled test examples. We devised an algorithm and instantiated a set of methods for optimizing the ARM objective that meta-learns models that are adaptable to different groups of training data. Empirically, we observed that ARM methods consistently improve performance in terms of both average and worst case metrics, as compared to a number of prior approaches for handling group shift.

Though we provided contextual meta-learning as a concrete example, a number of other meta-learning paradigms would also be interesting to extend to the ARM setting. For example, few shot generative modeling objectives would be a natural fit for unlabeled adaptation [13, 23, 70]. Another exciting direction for future work is to explore the problem setting where groups are not provided at training time. As discussed in Appendix E, in this setting, we can instead construct groups via unsupervised learning techniques. Similar to Hsu et al. [25], one promising approach is to generate a diverse set of groupings in order to learn generally effective adaptation strategies. Robustness and invariance methods cannot be used easily with multiple different groupings, learned or otherwise, as techniques such as group weighted loss functions [56] and group classifiers [16] are not immediately extendable to this setup. Thus, ARM methods may be uniquely suited to be paired with group learning.

References

[1] F. Alet, K. Kawaguchi, M. Bauza, N. Kuru, T. Lozano-Pérez, and L. Kaelbling. Tailoring: Encoding inductive biases by optimizing unsupervised objectives at prediction time. *arXiv preprint arXiv:2009.10623*, 2020.

[2] A. Antoniou and A. Storkey. Learning to learn via self-critique. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[3] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[4] S. Bengio, Y. Bengio, J. Cloutier, and J. Gecsei. On the optimization of a synaptic learning rule. In *Optimality in Artificial and Biological Neural Networks*, 1992.

[5] G. Blanchard, G. Lee, and C. Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[6] G. Blanchard, A. Deshmukh, U. Dogan, G. Lee, and C. Scott. Domain generalization by marginal transfer learning. *Journal of Machine Learning Research (JMLR)*, 2021.

[7] S. Caldas, S. Duddu, P. Wu, T. Li, J. Konečný, H. McMahan, V. Smith, and A. Talwalkar. LEAF: A benchmark for federated settings. In *Workshop on Federated Learning for Data Privacy and Confidentiality*, 2019.

[8] F. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. Rota Bulò. AutoDIAL: Automatic domaIn alignment layers. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[9] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik. EMNIST: An extension of MNIST to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017.

[10] G. Csurka. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*, 2017.

[11] H. Daumé III. Frustratingly easy domain adaptation. In *Conference of the Association for Computational Linguistics (ACL)*, 2007.

[12] Q. Dou, D. Castro, K. Kamnitsas, and B. Glocker. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[13] H. Edwards and A. Storkey. Towards a neural statistician. In *International Conference on Learning Representations (ICLR)*, 2017.

[14] C. Fang, Y. Xu, and D. Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[15] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

[16] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning (ICML)*, 2015.

[17] M. Garnelo, D. Rosenbaum, C. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. Teh, D. Rezende, and S. Eslami. Conditional neural processes. In *International Conference on Machine Learning (ICML)*, 2018.

[18] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations (ICLR)*, 2019.

[19] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations (ICLR)*, 2021.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[22] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019.

[23] L. Hewitt, M. Nye, A. Gane, T. Jaakkola, and J. Tenenbaum. The variational homoencoder: Learning to learn high capacity generative models from few examples. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018.

[24] S. Hochreiter, A. Younger, and P. Conwell. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks (ICANN)*, 2001.

[25] K. Hsu, S. Levine, and C. Finn. Unsupervised learning via meta-learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[26] S. Hu, P. Moreno, Y. Xiao, X. Shen, G. Obozinski, N. Lawrence, and A. Damianou. Empirical Bayes transductive meta-learning with synthetic gradients. In *International Conference on Learning Representations (ICLR)*, 2020.

[27] W. Hu, G. Niu, I. Sato, and M. Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning (ICML)*, 2018.

[28] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015.

[29] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with Gumbel-softmax. In *International Conference on Learning Representations (ICLR)*, 2017.

[30] A. Kaku, S. Mohan, A. Parnandi, H. Schambra, and C. Fernandez-Granda. Be like water: Robustness to extraneous variables via adaptive feature normalization. *arXiv preprint arXiv:2002.04019*, 2020.

[31] D. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

[32] P. Koh, S. Sagawa, H. Marklund, S. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. Phillips, I. Gao, T. Lee, E. David, I. Stavness, W. Guo, B. Earnshaw, I. Haque, S. Beery, J. Leskovec, A. Kundaje, E. Pierson, S. Levine, C. Finn, and P. Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning (ICML)*, 2021.

[33] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[34] D. Lazer, R. Kennedy, G. King, and A. Vespignani. The parable of Google flu: Traps in big data analysis. *Science*, 2014.

[35] D. Li, Y. Yang, Y. Song, and T. Hospedales. Deeper, broader and artier domain generalization. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[36] D. Li, Y. Yang, Y. Song, and T. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[37] H. Li, S. Pan, S. Wang, and A. Kot. Domain generalization with adversarial feature learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[38] T. Li, M. Sanjabi, A. Beirami, and V. Smith. Fair resource allocation in federated learning. In *International Conference on Learning Representations (ICLR)*, 2020.

[39] X. Li, Q. Sun, Y. Liu, Q. Zhou, S. Zheng, T. Chua, and B. Schiele. Learning to self-train for semi-supervised few-shot classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[40] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou. Revisiting batch normalization for practical domain adaptation. In *International Conference on Learning Representations Workshop (ICLRW)*, 2017.

[41] Y. Li, X. Tian, M. Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao. Deep domain generalization via conditional invariant adversarial networks. In *European Conference on Computer Vision (ECCV)*, 2018.

[42] Z. Lipton, Y. Wang, and A. Smola. Detecting and correcting for label shift with black box predictors. In *International Conference on Machine Learning (ICML)*, 2018.

[43] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. Hwang, and Y. Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[44] C. Maddison, A. Mnih, and Y. Teh. The Concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR)*, 2017.

[45] L. Metz, N. Maheswaranathan, B. Cheung, and J. Sohl-Dickstein. Meta-learning update rules for unsupervised representation learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[46] Z. Nado, S. Padhy, D. Sculley, A. D'Amour, B. Lakshminarayanan, and J. Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*, 2020.

[47] A. Nichol, J. Achiam, and J. Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[48] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[49] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang. Moment matching for multi-source domain adaptation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[50] J. Quiñonero Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[51] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2017.

[52] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. Tenenbaum, H. Larochelle, and R. Zemel. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations (ICLR)*, 2018.

[53] J. Requeima, J. Gordon, J. Bronskill, S. Nowozin, and R. Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[54] D. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning (ICML)*, 2014.

[55] A. Royer and C. Lampert. Classifier adaptation at prediction time. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[56] S. Sagawa, P. Koh, T. Hashimoto, and P. Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations (ICLR)*, 2020.

[57] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning (ICML)*, 2016.

[58] J. Schmidhuber. Evolutionary principles in self-referential learning. *Diploma thesis, Institut f. Informatik, Tech. Univ. Munich*, 1987.

[59] S. Schneider, E. Rusak, L. Eck, O. Bringmann, W. Brendel, and M. Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[60] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference (JSPI)*, 2000.

[61] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[62] M. Sulc and J. Matas. Improving CNN classifiers by estimating test-time priors. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[63] Y. Sun, X. Wang, Z. Liu, J. Miller, A. Efros, and M. Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning (ICML)*, 2020.

[64] S. Thrun and L. Pratt. *Learning to Learn*. Springer Science & Business Media, 1998.

[65] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[66] V. Vapnik. *Statistical Learning Theory*. Wiley New York, 1998.

[67] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[68] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 2021.

[69] G. Wilson and D. Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2020.

[70] M. Wu, K. Choi, N. Goodman, and S. Ermon. Meta-amortized variational inference and learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[71] T. Yu, C. Finn, A. Xie, S. Dasari, P. Abbeel, and S. Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. In *Robotics: Science and Systems (RSS)*, 2018.

[72] R. Zhang, T. Che, Z. Ghahramani, Y. Bengio, and Y. Song. MetaGAN: An adversarial approach to few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 7 and Appendix A.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix A.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] This paper is empirically focused, though we reference relevant theoretical results from Blanchard et al. [6] in Section 4.
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See code attached in supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix D.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Table 1.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Briefly, see Appendix D.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See references throughout.
   (b) Did you mention the license of the assets? [No] These license are easily accessible from the references.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See code attached in supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] Consent for using and modifying existing datasets was permitted by their licenses.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] The data we use do not.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]