Model Agnostic Sample Reweighting for Out-of-Distribution Learning

Xiao Zhou [*1] **Yong Lin** [*1] **Renjie Pi** [*1] **Weizhong Zhang** [1] **Renzhe Xu** [2] **Peng Cui** [2] **Tong Zhang** [1 3]

Abstract

Distributionally robust optimization (DRO) and invariant risk minimization (IRM) are two popular methods proposed to improve out-of-distribution (OOD) generalization performance of machine learning models. While effective for small models, it has been observed that these methods can be vulnerable to overfitting with large overparameterized models. This work proposes a principled method, **M**odel **A**gnostic sam**PL**e r**E**weighting (**MAPLE**), to effectively address OOD problem, especially in overparameterized scenarios. Our key idea is to find an effective reweighting of the training samples so that the standard empirical risk minimization training of a large model on the weighted training data leads to superior OOD generalization performance. The overfitting issue is addressed by considering a bilevel formulation to search for the sample reweighting, in which the generalization complexity depends on the search space of sample weights instead of the model size. We present theoretical analysis in linear case to prove the insensitivity of MAPLE to model size, and empirically verify its superiority in surpassing state-of-the-art methods by a large margin. Code is available at https://github.com/x-zho14/MAPLE.

1. Introduction

Despite the unprecedented success of deep learning in recent decades, machine learning methods are vulnerable to even slight distributional shift (Goyal et al., 2019; Sagawa et al., 2020; Gulrajani & Lopez-Paz, 2020). Actually, the common independent and identical distribution (IID) assumption in machine learning can be easily violated due to data selection biases or unobserved confounders that widely exist in real data (Liu et al., 2021b). Arjovsky et al. (2019) suggests that models trained by empirical risk minimization (ERM) can fail to learn causal factors but instead exploit the easier-to-fit spurious correlations, which are prone to distributional shift from training to testing domains (Gulrajani & Lopez-Paz, 2020). A typical example is that deep neural networks (DNN) can rely on the background (spurious features: sand or grassland) to distinguish between caw and camel (core features) (Beery et al., 2018). Such model can fail dramatically in recognizing a cow in desert. How to enable the deep models to generalize well under distributional shifts is an important long-standing problem.

In an effort to prevent DNN from exploiting the undesired spurious correlation, a popular research direction targets on regularizing DNN during training, including distributionally robust optimization (DRO) (Ben-Tal et al., 2013; Duchi et al., 2019; 2021; Sagawa et al., 2020) and invariant risk minimization (IRM) (Arjovsky et al., 2019; Krueger et al., 2021a; Xie et al., 2020). We refer them as **regularization-based methods** in this paper. DRO aims to optimize the worst case performance in a set of distributions within a certain distance to the original training distribution while IRM tries to learn an invariant representation that discards the spurious features. DRO and IRM have gained their popularity owed to promising performance on small models and datasets (Arjovsky et al., 2019; Duchi et al., 2019) and simplicity to perform training in an end-to-end manner. However, they are reported to be less effective when applied to DNNs in recent studies (Sagawa et al., 2019; Cherepanova et al., 2021; Yong Lin, 2021). Overparamterized DNN can easily reduce the regularization term of DRO or IRM to zero during training while still relying on the spurious features.

Another line of research is based on reweighting including importance sampling (Kanamori et al., 2009; Ben-Tal et al., 2013; Fang et al., 2020) and stable learning (Kuang et al., 2020; Shen et al., 2020; Xu et al., 2021). We refer to them as **reweighting-based methods**. They generally perform a two-stage pipeline: 1) reweight the data distribution by some heuristics; 2) perform ERM training on the reweighted distribution. In the first stage, they assign a weight to each sample: importance sampling upweights the rare group inversely to its group size and stable learning tries to find a weight that makes each feature orthogonal. With the weights found in the first stage, the second stage of weighted ERM training becomes resistant to spurious features. Since the

[*]Equal contribution [1]The Hong Kong University of Science and Technology [2]Tsinghua University [3]Google Research. Correspondence to: Tong Zhang <tongzhang@tongzhang-ml.org>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

first stage is agnostic to the model size of DNN, it does not suffer from the vulnerability of overfitting caused by model overparameterization as in DRO and IRM. However, the heuristics in those reweighting based methods require more strict prior knowledge like group annoatations to perform well, which makes them less competitive in practice compared with regularization-based counterparts.

In this paper, to resolve the above limitations, we propose a model agnostic sample reweighting method integrating the benefits of two lines of previous works. In short, we solve the overfitting problem of regularization-based methods by taking the weighted ERM training pattern and transform the search space of model parameters into that of sample weights. On the other hand, we avoid the strict requirements of reweighting-based methods by learning sample weights automatically. To achieve this, we formulate the learning of sample reweighting into a bilevel optimization problem. In the inner loop, we train the DNN on the weighted training samples. In the outer loop, we ultilize the OOD criterion evaluated on validation set as the outer objective to guide the learning of the sample weights. We alternatively perform the inner loop and outer loop and finally obtain a set of weights $\boldsymbol{w}$ with the such appealing property: with only learnt sample weights and training samples, we are able to perform weighted ERM training to obtain superior OOD performance, without any regularization term or strict prior knowledge on training samples. We use the term *model agnostic* in MAPLE to stress its ability to avoid overfitting regardless of the model size. In addition, the learned sample weights do not have strong dependence on the model used during the searching phase, e.g. the sample weights learned through ResNet-18 can be successfully applied to weighted ERM training on ResNet-50 on the same task (Table 6).

The general bilevel framework is presented below:

- **Outer loop.** Evaluate the model $\boldsymbol{\theta}$ by the OOD criterions to measure the model's reliance to spurious features and optimize $\boldsymbol{w}$ to minimize the criterion.
- **Inner loop.** Perform ERM training on the dataset weighted by $\boldsymbol{w}$ to obtain learned model $\boldsymbol{\theta}$.

An appealing feature of this formulation is that the inner loop can be viewed as a mapping from the sample weight space into the DNN parameter space, and the outer loop performs the optimization on weights. Our bilevel optimization framework is less prone to overfitting because it only searches for the weight candidate: the space of weight is much smaller than that of neural networks. For example, CIFAR-10 only contains 50K training data while ResNet-18 exhibits 11.4 million parameters. We empirically demonstrate the effectiveness of MAPLE on various OOD tasks and show that MAPLE surpasses the state-of-the-art methods by a large margin. Remarkably, we achieve even higher worst-group accuracy in Waterbirds without group labels in training samples compared with GroupDRO previously recognized as the Oracle upperbound (Table 2).

Our contributions are summarized as follows:

- We propose a model agnostic sample reweighting method based on bilevel optimization for OOD learning, which enjoys the following benefits:
  - MAPLE learns sample weights automatically through bilevel optimization avoiding the pathology of conventional reweighting-based methods' reliance on strong prior knowledge on data.
  - MAPLE transforms the optimization problem from DNN's parameter space to sample weight space, which in turn solves the overfitting problem suffered by regularization-based methods.
- We provide theoretical analysis in linear case to prove the existence of ideal sample weight under suitable conditions and insensitivity of the generalization performance to the model capacity of DNN, which is consistent with our empirical results.
- We empirically demonstrate the superior performance of MAPLE to state-of-the-art domain generalization methods on various tasks and models.

2. Related Work

Invariant Risk Minimization. IRM aims to learn a feature representation which elicits a classifier that is simultaneously optimal in various environments (Peters et al., 2016; Arjovsky et al., 2019). Several works try to improve IRM by proposing different variants: (Krueger et al., 2021b; Xie et al., 2020) suggest to penalize the variance of the risks among different environments and (Chang et al., 2020; Xu & Jaakkola, 2021) try to estimate the invariance violation by training neural networks. (Arjovsky et al., 2019; Rosenfeld et al., 2020; Chen et al., 2021b) provide theoretical guarantees for IRM on linear models with sufficient training environments. However, IRM is found to be less effective when applied to overparameterized neural networks (Gulrajani & Lopez-Paz, 2020; Lin et al., 2021). (Lin et al., 2022a) shows that this can be largely attributed to the overfitting problem.

Distributionally Robust Optimization. DRO optimizes the worst-case loss in an uncertainty set (Ben-Tal et al., 2013; Sagawa et al., 2019; Duchi et al., 2019; Oren et al., 2019; Duchi et al., 2021; Zhang et al., 2022). When the uncertainty set is properly chosen, Duchi & Namkoong (2019; 2021) shows that DRO can improve the robustness of the learned model by imposing regularization. Unfortunately, similar to IRM, DRO is also shown to be less effective on overparameterized neural networks (Sagawa et al., 2019), which may be largely attributed to the deep model's ability to overfit all the training data. In an effort to enhance DRO

in this case, Sagawa et al. (2019) suggests to impose large ℓ_2 regularization or early stopping on the DNN to alleviate the catastrophic overfitting. Liu et al. (2021a) proposes a two-stage method that firstly performs ERM with early stopping and then conduct weighted ERM training by upweighting misclassified samples from the model obtained in the first stage.

Reweighting. Sample reweighting is a classic method to deal with distribution shifts. Traditional sample reweighting methods, e.g., importance sampling, assume the prior knowledge of testing distributions are known and they can estimate the density ratio between training and testing distributions directly (Shimodaira, 2000; Huang et al., 2006; Sugiyama et al., 2007; 2008; Kanamori et al., 2009; Fang et al., 2020). As a result, ERM training on the reweighted distribution is unbiased in the testing distribution (Fang et al., 2020). Recent works consider a much more challenging setting where the testing distribution is unknown (Shen et al., 2021). In this direction, stable learning proposes to learn sample weights that make features statistically independent in the reweighted distribution (Kuang et al., 2020; Shen et al., 2020; Zhang et al., 2021b; Wang et al., 2022; Xu et al., 2020). Xu et al. (2021) further theoretically analyze the effectiveness of such algorithms by explaining them as processes of feature selection. However, stable learning is still limited in the sense that the features need to be provided generally. A recent work aiming at addressing learning with label noise also relies on optimizing sample reweighting using a bilevel framework (Ren et al., 2018), where a validation set with the same distribution as the test set is needed to ensure good performance. However, in OOD tasks, the training and validation sets are from the same distribution, which is different from the test distribution, rendering these methods inapplicable.

Causality. The topics covered in this work is closely related to causality. Peters et al. (2016) proposes Invariant Causal Prediction (ICP) to utilize the invariance property to identify the direct cause of the target. IRM then extends this idea to DNN by incorporating feature learning (Arjovsky et al., 2019). Both ICP and IRM need train data to be split into distinct environments, whereas, environments partition is frequently not available in real application. It is of great interest to learn invariance without explicit environment indexes. (Lin et al., 2022b) proposes a framework called ZIN that can provably learn both invariance and environment partition based on the carefully chosen auxiliary information. DRO is also intrinsically related to causality by noting that causal model optimizes the worst case loss w.r.t. infinite intervention on the causal graph. (Rothenhäusler et al., 2021) explicitly build the connection between distributional robustness with causality. We believe our method is also a potential technique to make causal models compatible with large neural networks.

3. Preliminaries

Notations. Given a dataset $\mathcal{D} := \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ with samples $(\mathbf{x}_i, \mathbf{y}_i)$ drawn from $\mathcal{X} \times \mathcal{Y}$, we denote weighted empirical loss as $\mathcal{L}(\mathcal{D}, \boldsymbol{\theta}; \boldsymbol{w}) := \frac{1}{n}\sum_{i=1}^n w_i \ell(f(\mathbf{x}_i; \boldsymbol{\theta}), \mathbf{y}_i)$, where $f(\cdot; \boldsymbol{\theta})$ is a network parameterized by $\boldsymbol{\theta}$, $\ell(\cdot, \cdot)$ is the loss function, e.g., cross entropy and least square loss, and $w_i \in \mathbb{R}^+$ is the non-negative weight. We denote $\mathcal{L}(\mathcal{D}, \boldsymbol{\theta})$ to be the unweighted loss $\mathcal{L}(\mathcal{D}, \boldsymbol{\theta}; \mathbf{1})$ for abbreviation. Let $\boldsymbol{z}_c \in \mathcal{Z}_c$ and $\boldsymbol{z}_s \in \mathcal{Z}_s$ be the *core* and *spurious* features. The core feature is safe to rely on and the reliance on the spurious feature is unstable and unwanted. We assume the observed feature space is generated by an unknown/known mapping from the core and spurious feature spaces, i.e., $\mathcal{K}(\cdot, \cdot) : \mathcal{Z}_c \times \mathcal{Z}_s \to \mathcal{X}$.

IRM and DRO aim to learn a *good* predictor $f : \mathcal{X} \to \mathcal{Y}$, in a sense that f does not rely on the spurious feature $\mathcal{Z}_s$. They formulate it into a minimization problem of different objective functions (referred as *OOD Risk*) based on different settings in practice. The details are presented below.

3.1. IRM

IRM assumes that we have multiple environments $\mathcal{E} := \{e_1, e_2, \ldots, e_E\}$ in the sample space $\mathcal{X} \times \mathcal{Y}$ with different joint distributions, and the correlation between the spurious features and labels is unstable among different environments. IRM formulates the predictor $f(\cdot; \boldsymbol{\theta})$ as a composite function of representaion $\phi(\cdot; \Phi)$ and classifier $h(\cdot; \boldsymbol{v})$, i.e., $f(\cdot; \boldsymbol{\theta}) = h(\phi(\cdot; \Phi); \boldsymbol{v})$, where $\boldsymbol{\theta} = \{\boldsymbol{v}, \Phi\}$ are the trainable parameters. Its idea is that if a predictor $f(\cdot; \boldsymbol{\theta})$ works well on all the environments, then it can be expected that the correlation between the spurious features and the labels are not fitted as it is unstable. Therefore, it formulates the task as to minimize a certain OOD risk to find such good predictor. Two popular risks are

$$\mathcal{R}^{\text{IRMv1}}(\mathcal{D}, \boldsymbol{\theta}) := \sum_e \mathcal{L}(\mathcal{D}^e, \boldsymbol{\theta}) + \lambda \|\nabla_v \mathcal{L}(\mathcal{D}^e, \boldsymbol{\theta})\|_2^2 \quad (1)$$

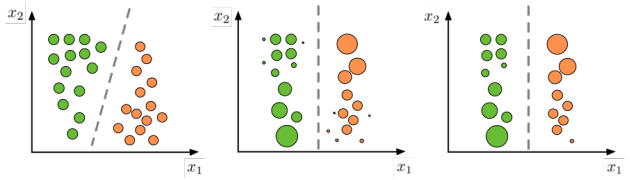
$$\mathcal{R}^{\text{REx}}(\mathcal{D}, \boldsymbol{\theta}) := \sum_e \mathcal{L}(\mathcal{D}^e, \boldsymbol{\theta}) + \lambda \mathbb{V}_e[\mathcal{L}(\mathcal{D}^e, \boldsymbol{\theta})], \quad (2)$$

where $\mathcal{D} = \cup_e \mathcal{D}^e$ with $\mathcal{D}^e$ being the data drawn from environment e and $\mathbb{V}_e[\mathcal{L}(\mathcal{D}^e, \boldsymbol{\theta})]$ is the variance of the loss across different environments.

3.2. DRO

DRO aims to optimize the worst case performance in a set of distributions within a certain distance to the original training distribution.

When a set of distributions with different group annotations g, i.e., $\mathcal{D} = \bigcup_g \mathcal{D}^g$, is available, a popular method named GroupDRO (Sagawa et al., 2019) learns a robust predictor by minimizing the following risk, which is actually the



(a) unweighted (b) weighted (c) weighted+sparse

Figure 1. An illustrative example of removing the reliance on spurious feature via sample reweighting and sparsity constraint on sample size. Circles with larger radius means more weight paid to this training sample. Different colors indicate different labels, i.e., {0,1}. Here, $x_1 = z_c$ and $x_2 = z_s$ are the core and spurious features, respectively.

worst-group loss over $\{\mathcal{D}^g\}_g$, i.e.,:

$$\mathcal{R}_{\text{Group-DRO}}(\mathcal{D}, \theta) := \max_g \mathcal{L}(\mathcal{D}^g, \theta). \tag{3}$$

When such set of distributions is not available, a typical method, conditional value at risk (CVaR) (Rockafellar et al., 2000), constructs distributions near the original training distributions by reweighting on the training samples and minimizes a risk defined as the supreme loss over these distributions, i.e.,

$$\mathcal{R}_{\text{CVaR-DRO}}(\mathcal{D}, \theta) := \sup_{\boldsymbol{w} \in \mathcal{C}(\alpha)} \mathcal{L}(\mathcal{D}, \theta; \boldsymbol{w}), \tag{4}$$

where $\mathcal{C}(\alpha) = \{\boldsymbol{w} : \boldsymbol{w} \succeq 0, \|\boldsymbol{w}\|_\infty \leq \frac{1}{\alpha n}, \|\boldsymbol{w}\|_1 = 1\}$.

4. Model Agnostic Sample Reweighting

In this section, we will first present the bilevel formulation of our proposed MAPLE and provide some theoretical analysis about its generalization ability. Then we will introduce sparsity into MAPLE to enhance its generalization ability.

4.1. Bilevel Formulation of MAPLE

We illustrate our key idea using the example in Figure 1, which is to remove the reliance of the learned predictor f on the spurious features by sample reweighting. To be precise, in this example, we assume x_1 and x_2 are the core and spurious features, and we aim to learn a classifier on these training data. If without reweighting, it is clear that with conventional loss functions, the optimal classifier is the dashed slant line in Figure 1.(a), which depends on x_2. If we assign larger weights to the samples in the left-bottem and right-up areas, then the optimal classifier would rotate to be vertical shown in Figure 1.(b). We can see the vertical classifier does not depend on the spurious feature x_2, as for fixed x_1 and any value of x_2, the output of the classifier never changes. Therefore, it shows that we can remove the reliance on the spurious features by sample reweighting. Thus, the problem comes to how to automatically learn appropriate weights for training samples.

Consider a training dataset $\mathcal{D}_{tr} := \{(\mathbf{x}_i^{tr}, \mathbf{y}_i^{tr})\}_{i=1}^{n_{tr}}$ and a validation dataset $\mathcal{D}_v := \{(\mathbf{x}_i^{v}, \mathbf{y}_i^{v})\}_{i=1}^{n_v}$ randomly partitioned from dataset $\mathcal{D}$. We formulate the task of learning sample weights to remove the reliance on the spurious features as the following bilevel optimization problem:

$$\min_{\boldsymbol{w} \in \mathcal{C}} \mathcal{R}(\mathcal{D}_v, \boldsymbol{\theta}^*(\boldsymbol{w})), \tag{5}$$

$$s.t.\ \boldsymbol{\theta}^*(\boldsymbol{w}) \in \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{D}_{tr}, \boldsymbol{\theta}; \boldsymbol{w}), \tag{6}$$

where $\boldsymbol{w}$ is a sample weight vector with length n_{tr} indicating the importance of training samples, $\mathcal{C} = \{\boldsymbol{w} : \boldsymbol{w} \succeq 0\}$. Any OOD Risk $\mathcal{R}(\mathcal{D}, \boldsymbol{\theta})$ described in the Section 3 can be used as the outer objective here. In the inner loop, we minimize the weighted ERM loss on training samples, in order to obtain a model $\boldsymbol{\theta}^*(\boldsymbol{w})$, and in the outer loop, we evaluate the learned model's reliance on spurious features through OOD Risk and optimize sample weights. By alternatively performing inner loop and outer loop, the sample weights gradually evolve to the state of being able to produce satisfactory OOD performance with simply ERM training.

Moreover, our formulation has the following advantages:

- In our framework, we essentially define an implicit mapping from the sample weight space to the model parameter space, which enables us to learn in the sample weight space. As the sample weight space is always significantly smaller than model parameter spaces, we can avoid the pathology of overfitting caused by over-parameterization.
- Compared with existing regularization-based methods, MAPLE adopts validation dataset in the outer loop to alleviate the problem of overfitting to training dataset.

These advantages are consistent with our theoretical analysis (Section 4.2) and empirical observations (Section 5).

[Optimization by Truncated Back-propagation and Projected Gradient Descent]. The above bilevel optimization can be solved by performing projected gradient descent to $\boldsymbol{w}$. The gradient of $\boldsymbol{w}$ can be calculated by:

$$\begin{aligned}
&\nabla_{\boldsymbol{w}} \mathcal{R} \\
&= \nabla_{\boldsymbol{\theta}} \mathcal{R}|_{\theta^*} \nabla_{\boldsymbol{w}} \boldsymbol{\theta}^* && (7)\\
&\approx \nabla_{\boldsymbol{\theta}} \mathcal{R}|_{\theta_T} \nabla_{\boldsymbol{w}} \boldsymbol{\theta}_T && (8)\\
&= \nabla_{\boldsymbol{\theta}} \mathcal{R}|_{\theta_T} \sum_{j \leq L} \left[\prod_{k<j} I - \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\intercal}\Big|_{\boldsymbol{\theta}_{T-k-1}} \right] \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta} \partial \boldsymbol{w}^\intercal}\Big|_{\boldsymbol{\theta}_{T-j-1}} \\
&\approx \nabla_{\boldsymbol{\theta}} \mathcal{R}|_{\theta_T} \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta} \partial \boldsymbol{w}^\intercal}\Big|_{\boldsymbol{\theta}_{T-1}}, && (9)
\end{aligned}$$

where Eqn. (7) follows chain rule, Eqn. (8) approximates θ^* by θ_T obtained from T steps of inner loop gradient descent and Eqn. (9) performs 1-step truncated backpropagation

(Shaban et al., 2019). Then MAPLE updates $\boldsymbol{w}$ by projected gradient descent:

$$\boldsymbol{w} \leftarrow \text{proj}_{\mathcal{C}}\left(\boldsymbol{w} - \eta\, \nabla_{\boldsymbol{\theta}}\mathcal{R}|_{\theta_T}\, \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta}\partial \boldsymbol{w}^\intercal}\Big|_{\boldsymbol{\theta}_{T-1}}\right), \quad (10)$$

where η is the learning rate.

4.2. Theoretical Analysis on Linear Case

In this section, we analyze the performance of our method in the linear case where we consider Problem (5) with linear predictor $f(\boldsymbol{x};\boldsymbol{\theta}) = \boldsymbol{x}^\intercal\boldsymbol{\theta}, \boldsymbol{\theta} \in \mathbb{R}^d$ and least square loss $\ell(f(\boldsymbol{x}), \boldsymbol{y}) = \|f(\boldsymbol{x}) - \boldsymbol{y}\|_2^2$. We further consider $\boldsymbol{x} \in \mathbb{R}^d$ to be the generated from core features $\boldsymbol{z}_c \in \mathbb{R}^{d_c}$ and spurious features $\boldsymbol{z}_s \in \mathbb{R}^{d_s}$ by a transformation matrix $\boldsymbol{S} \in \mathbb{R}^{d\times(d_c+d_s)}$, i.e., $\boldsymbol{x} = \boldsymbol{S}[\boldsymbol{z}_c; \boldsymbol{z}_s]$. We assume $d_c + d_s = d$ for simplicity and assume the feature transformation $\boldsymbol{S}$ is invertible by some matrix $\boldsymbol{T} \in \mathbb{R}^{(d_c+d_s)\times d}$ such that $\boldsymbol{TS}([\boldsymbol{z}_c; \boldsymbol{z}_s]) = [\boldsymbol{z}_c; \boldsymbol{z}_s]$. Our goal is to learn a function f that predicts $\boldsymbol{y}$ based on $\boldsymbol{x}$ without reliance on $\boldsymbol{z}_s$. Let $\mathbb{P}(\boldsymbol{x}, \boldsymbol{y})$ denote the distribution on the training and validation sets as defined in Section 4.1. We further use $\mathbb{E}$ to denote the expectation w.r.t. $\mathbb{P}(\boldsymbol{x}, \boldsymbol{y})$.

In Section 4.2.1, we consider the population level property, i.e., when infinite samples are available. In Section 4.2.2, we consider the case with finite samples.

4.2.1. Population Level Properties

At first, we need to extend the weight and loss of problem (13) into the population level as follows.

Definition 4.1. We define the set of weight functions as

$$\mathcal{W} = \{w : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+ | \mathbb{E}[w(\boldsymbol{x}, \boldsymbol{y})] = 1\}.$$

Given any $w \in \mathcal{W}$, the populated unweighted and weighted loss can be defined as

$$\mathcal{L}(\boldsymbol{\theta}) = \int (y - \boldsymbol{x}^\intercal\boldsymbol{\theta})^2 \mathbb{P}(\boldsymbol{x}, y) d\boldsymbol{x} dy \quad (11)$$

$$\mathcal{L}(\boldsymbol{\theta}; w) = \int (y - \boldsymbol{x}^\intercal\boldsymbol{\theta})^2 \mathbb{P}_w(\boldsymbol{x}, y) d\boldsymbol{x} dy, \quad (12)$$

where $\mathbb{P}_w(\boldsymbol{x}, \boldsymbol{y}) = w(\boldsymbol{x}, \boldsymbol{y})\mathbb{P}(\boldsymbol{x}, \boldsymbol{y})$ is the weighted distribution.

The populated version of problem (5) takes the form of

$$\min_{\boldsymbol{w}\in\mathcal{C}} \mathcal{R}(\boldsymbol{\theta}^*(w)), \quad (13)$$

$$s.t.\ \boldsymbol{\theta}^*(w) \in \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; w), \quad (14)$$

here $\mathcal{R}(\boldsymbol{\theta})$ is the populated OOD risk obtained by replacing the empirical loss with the populated one in Eqn (1)-(4). We assume the solution in the inner loop is unique. We define the optimal linear classifier as the one that minimizes the expected loss without using any spurious features:

Definition 4.2. We define the optimal debiased predictor as

$$\bar{\boldsymbol{\theta}} := \boldsymbol{T}^\intercal[\bar{\boldsymbol{\theta}}_c; \boldsymbol{0}],$$

where $\bar{\boldsymbol{\theta}}_c := \arg\min_{\boldsymbol{\theta}_c} \mathbb{E}[\|y - \boldsymbol{z}_c^\intercal\boldsymbol{\theta}_c\|^2]$.

We now make further assumptions as follows:

Assumption 4.3 (Strictly positive density)**.** $\forall \mathbf{y} \in \mathcal{Y}, \mathbf{z}_c \in \mathcal{Z}_c, \mathbf{z}_s \in \mathcal{Z}_s, P(\boldsymbol{z}_c = \mathbf{z}_c, \boldsymbol{z}_s = \mathbf{z}_s, \boldsymbol{y} = \mathbf{y}) > 0$.

Assumption 4.4. The optimal debiased predictor $\bar{\boldsymbol{\theta}}$ is identifiable by the populated OOD Risk $\mathcal{R}$, i.e.,

$$\mathcal{R}(\bar{\boldsymbol{\theta}}) < \mathcal{R}(\boldsymbol{\theta}), \forall \boldsymbol{\theta} \in \mathbb{R}^d, \boldsymbol{\theta} \neq \bar{\boldsymbol{\theta}}.$$

Assumption 4.3 is common in existing works because there always exists uncertainty in the data (Pearl, 1988; Strobl & Visweswaran, 2016; Xu et al., 2021). Assumption 4.4 is a natural condition, making it possible to provably identify $\bar{\boldsymbol{\theta}}$ by using $\mathcal{R}$. For example, it has been demonstrated that the metrics of IRM can satisfy this condition with sufficient number of environments (Arjovsky et al., 2019; Rosenfeld et al., 2020).

Theorem 4.5 (Identifiability on population level)**.** *When Assumption 4.3 holds, there exists a weight function $w \in \mathcal{W}$, such that the optimum solution of Eq. (14) satisfies that*

$$\boldsymbol{\theta}^*(w) = \bar{\boldsymbol{\theta}}.$$

Further, when Assumption 4.4 holds, the populated MAPLE, i.e., Eqn.(13)-(14), can uniquely identify $\bar{\boldsymbol{\theta}}$.

The theorem above shows MAPLE can provably find the sample weight to removes reliance of model on the spurious features. This verify the main idea illustrated in Figure 1.

4.2.2. Finite Sample

Now we turn to analyze the finite sample case. By extending the weight vector $\boldsymbol{w}$ into the functional form $w(\boldsymbol{x}, y)$ in Definition 4.1, we rewrite problem (5) into:

$$\min_{\boldsymbol{w}\in\mathcal{C}} \mathcal{R}(\mathcal{D}_v, \hat{\boldsymbol{\theta}}^*(w)), \quad (15)$$

$$s.t.\ \hat{\boldsymbol{\theta}}^*(w) = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{D}_{tr}, \boldsymbol{\theta}; w),$$

Then given a weight function w, $\hat{\boldsymbol{\theta}}^*(w)$ is a deterministic mapping from $\mathcal{W}$ to the parameter space. Suppose we can find a $\hat{w}$ that is a $\epsilon-$approximate solution of minimizing $\mathcal{R}(\mathcal{D}_v, \hat{\boldsymbol{\theta}}^*(w))$, i.e.,

$$\mathcal{R}(\mathcal{D}_v, \hat{\boldsymbol{\theta}}^*(\hat{w})) \leq \inf_{w\in\mathcal{W}} \mathcal{R}(\mathcal{D}_v, \hat{\boldsymbol{\theta}}^*(w)) + \epsilon. \quad (16)$$

Observing that $\hat{\boldsymbol{\theta}}^*(\cdot)$ only depends on $\mathcal{D}_{tr}$, we can obtain the following generalization bound with standard uniform convergence analysis on $\mathcal{D}_v$:

Theorem 4.6 (Finite Samples). *Suppose $|\mathcal{D}_v| = n$. Let $\mathcal{D}_v^{-1}$ denote the dataset generated by replacing one sample in $\mathcal{D}_v$ with another arbitrary sample. Assume there exists a constant $M > 0$ such that $\forall \boldsymbol{\theta}, |\mathcal{R}(\mathcal{D}_v, \boldsymbol{\theta}) - \mathcal{R}(\mathcal{D}_v^{-1}, \boldsymbol{\theta})| \leq M/n$, where $\mathcal{R}(\mathcal{D}, \boldsymbol{\theta})$ denotes the OOD risk on the dataset $\mathcal{D}$. Further assume $\mathcal{W}$ contains $|\mathcal{W}|$ discrete choices. With probability at least $1 - \delta$, MAPLE outputs a solution $\hat{w}$ satisfies*

$$\mathcal{R}(\hat{\boldsymbol{\theta}}^*(\hat{w})) \leq \inf_{w \in \mathcal{W}} \mathcal{R}(\hat{\boldsymbol{\theta}}^*(w)) + \epsilon + M\sqrt{\frac{2\ln(2|\mathcal{W}|/\delta)}{n}}, \tag{17}$$

where $\mathcal{R}(\boldsymbol{\theta})$ is the populated OOD risk achieved by $\boldsymbol{\theta}$, $\hat{\boldsymbol{\theta}}$ and ϵ are defined in Eqn. (15) and Eqn. (16), respectively.

Theorem 4.6 shows that the generalization performance depends on the complexity of $\mathcal{W}$ and the size of validation dataset. As our weight space $\mathcal{W}$ is usually significantly smaller than the parameter space, MAPLE could have better generalization performance compared with training OOD risk directly on DNN. Further, the RHS of Eqn. (17) does not involve the complexity of neural networks, indicating MAPLE is insensitive to the model size. Extensive experimental results in Section 5 verify this result, showing MAPLE can achieve significant better performance than existing methods, especially on large models. We'd like to point out that Theorem 4.6 still holds when $\hat{\boldsymbol{\theta}}^*(\cdot)$ is a general non-linear function because the theorem is a direct application of the standard uniform convergence analysis which doesn't require $\hat{\boldsymbol{\theta}}^*(\cdot)$ to be linear.

4.3. Enhance MAPLE by sparsity

As shown in Figure 1.(c), we further impose a sparsity constraint on the training sample size, i.e., $\mathcal{C}$ becomes $\{\boldsymbol{w} : \boldsymbol{w} \succeq 0, \|\boldsymbol{w}\|_0 \leq K\}$ in order to save the computational cost in the inner loop. We will verify the benefit in our experiment. Intuitively, sparsity can be seen as forcing several sample weights to be zero. In this way, noisy data samples are removed. Inspired by previous works on L_0 regularization optimization (Louizos et al., 2018; Zhou et al., 2021a;b; Zou et al., 2019), we relax the original formulation to be continuous:

$$\min_{(\boldsymbol{w},\boldsymbol{s}) \in \mathcal{C}'} \Phi(\boldsymbol{w}, \boldsymbol{s}) = \mathbb{E}_{p(\boldsymbol{m}|\boldsymbol{s})}\, \mathcal{R}(\mathcal{D}_v, \boldsymbol{\theta}^*(\boldsymbol{w}, \boldsymbol{m})), \tag{18}$$
$$s.t.\ \boldsymbol{\theta}^*(\boldsymbol{w}, \boldsymbol{m}) \in \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{D}_{tr}, \boldsymbol{\theta}; \boldsymbol{w} \circ \boldsymbol{m})$$

where $\mathcal{C}' = \{(\boldsymbol{w}, \boldsymbol{s}) : \boldsymbol{w} \succeq 0, 0 \preceq \boldsymbol{s} \preceq 1, \|\boldsymbol{s}\|_1 \leq K\}$ is the feasible set, m_i is viewed as a Bernoulli random variable with probability s_i to be 1 and $1 - s_i$ to be 0. Assuming the variables m_i are independent, we can get $p(\boldsymbol{m}|\boldsymbol{s}) = \Pi_{i=1}^n (s_i)^{m_i}(1-s_i)^{(1-m_i)}$. The discrete constraint $\|\boldsymbol{w}\|_0 \leq K$ in problem (5) can be relaxed into $\|\boldsymbol{s}\|_1 \leq K$.

We calculate the gradient to $\boldsymbol{w}$ and $\boldsymbol{s}$ by Straight-through Gumbel-softmax (Paulus et al., 2021):

$$\nabla_{\boldsymbol{w},\boldsymbol{s}} \Phi \approx \nabla_{\boldsymbol{w},\boldsymbol{s}} \mathcal{R}(\boldsymbol{\theta}^*(\boldsymbol{w}, \mathbb{1}(\log(\frac{\boldsymbol{s}}{\mathbf{1}-\boldsymbol{s}}) + \boldsymbol{g_1} - \boldsymbol{g_0} \geq 0))),$$

where $\boldsymbol{g_0}$ and $\boldsymbol{g_1}$ are two random variables with each element IID sampled from $\mathrm{Gumbel}(0, 1)$ and the following calculations are similar to those of Eqn. 9. Then MAPLE updates $\boldsymbol{w}$ and $\boldsymbol{s}$ by projected gradient descent:

$$(\boldsymbol{w}, \boldsymbol{s}) \leftarrow \mathrm{proj}_{\mathcal{C}'}(\boldsymbol{w} - \eta \nabla_{\boldsymbol{w}} \Phi, \boldsymbol{s} - \eta \nabla_{\boldsymbol{s}} \Phi), \tag{19}$$

where η is the learning rate.

5. Experiment

In this section, we conduct a series of experiments to justify the superiority of our MAPLE in IRM and DRO. Detailed dataset descriptions and experimental configurations are placed in appendix due to space limitation.

5.1. Datasets and Baselines

[Datasets]. For IRM experiments, ColoredMNIST is the most widely used benchmark in IRM and ColoredObject, CIFARMNIST are adopted to showcase the superior performance of MAPLE on more challenging largescale settings (Arjovsky et al., 2019; Krueger et al., 2021b; Ahuja et al., 2020; Zhang et al., 2021a). We adopt two popular vision datasets, Waterbirds and CelebA, to validate the effectiveness of MAPLE on DRO problems (Wah et al., 2011; Sagawa et al., 2019; Liu et al., 2015; Sagawa et al., 2019; Liu et al., 2021a; Lin et al., 2021). Waterbirds and CelebA are real-world datasets and we adopt them to demonstrate the generalizability of MAPLE to real-world scenarios. We follow the challenging setting of Liu et al. (2021a) where no group annotation is provided in the training dataset.

[Baselines]. To demonstrate the superiority of our MAPLE on IRM, we compare with standard empirical risk minimization (ERM), two popular foundational invariant risk minimization methods IRMv1 (Arjovsky et al., 2019) and REx (Krueger et al., 2021b) and the latest competitive method MRM (Zhang et al., 2021a) and SparseIRM (Zhou et al., 2022b) which boost IRM via imposing sparsity. SparseIRM imposes sparsity during training while MRM imposes sparsity after training. We also compare with BayesianIRM (Lin et al., 2022a) which introduces Bayesian Inference into IRM to estimate a distribution of classifiers. We include ERM trained on datasets without spurious features to serve as an upper bound (Oracle). To showcase the effectiveness of MAPLE on DRO, we compare with standard empirical risk minimization (ERM), three widely-used DRO methods without group annotations on the training samples: CVaR DRO (Levy et al., 2020) which is described in Eqn. (4), Learn from failure (LfF) (Nam et al., 2020), Just Train Twice

Algorithm 1 Model Agnostic Sample Reweighting (MAPLE)

Input: a network $\boldsymbol{\theta}$, remaining training sample size K, training set $\mathcal{D}_{tr}$ and validation set $\mathcal{D}_v$.
1: Initialize sample weights $\boldsymbol{w} = \mathbf{1}$ and probabilities $\boldsymbol{s} = \frac{K}{|\mathcal{D}_{tr}|}\mathbf{1}$.
2: **for** training iteration $i = 1, 2 \dots I$ **do**
3: Sample mask $\boldsymbol{m}$ according to the probability distribution $p(\boldsymbol{m}|\boldsymbol{s}) = \Pi_{i=1}^{n}(s_i)^{m_i}(1-s_i)^{(1-m_i)}$.
4: Train the inner loop to converge: $\boldsymbol{\theta}^*(\boldsymbol{w}, \boldsymbol{m}) \leftarrow \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{D}_{tr}, \boldsymbol{\theta}; \boldsymbol{w}, \boldsymbol{m})$ started from randomly initialized $\boldsymbol{\theta}$.
5: Estimate $\nabla_{\boldsymbol{s}}\Phi(\boldsymbol{w}, \boldsymbol{s})$ and $\nabla_{\boldsymbol{w}}\Phi(\boldsymbol{w}, \boldsymbol{s})$ by Straight-through Gumbel-softmax and 1-step truncated backpropagation.
6: Perform projected gradient descent: $(\boldsymbol{w}, \boldsymbol{s}) \leftarrow \text{proj}_{\mathcal{C}'}(\boldsymbol{w} - \eta\nabla_{\boldsymbol{w}}\Phi(\boldsymbol{w}, \boldsymbol{s}), \boldsymbol{s} - \eta\nabla_{\boldsymbol{s}}\Phi(\boldsymbol{w}, \boldsymbol{s}))$
7: **end for**
output The weighted set $\{(\mathbf{x}_i, \mathbf{y}_i, w_i) : m_i \neq 0, (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}_{tr}\}$ with $\boldsymbol{m}$ sampled from $p(\boldsymbol{m}|\boldsymbol{s})$

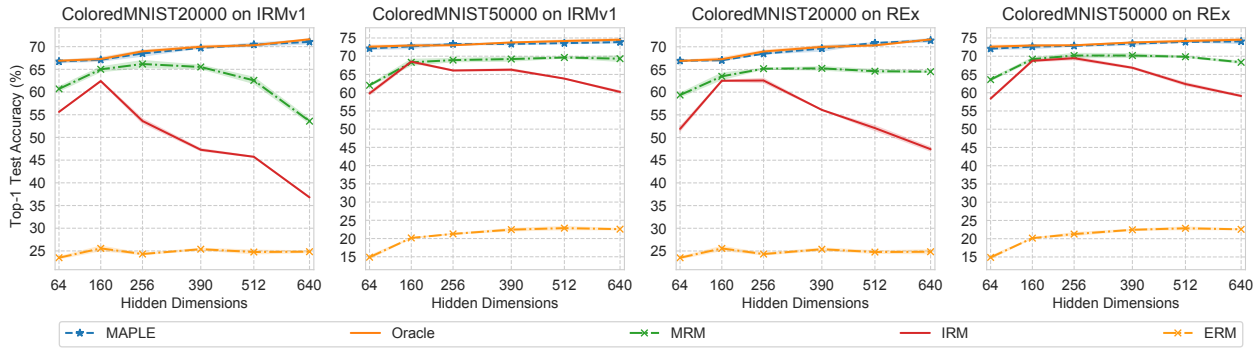


Figure 2. Comparing MAPLE with Oracle, IRM, MRM (Zhang et al., 2021a) and ERM on MLP on ColoredMNIST with varying hidden dimensions and dataset sizes. Oracle is the method done with ERM training with no spurious features and serves as an upper bound. The left (right) two figures demonstrate the comparison of MAPLE with IRMv1 (REx) where MAPLE adopts the same IRMv1 (REx) loss as the outer objective. MAPLE achieves comparable generalization performance with Oracle in all settings.

(JTT) (Liu et al., 2021a) and two DRO methods demanding group annotations on the training samples: UpWeighting (Cui et al., 2019; Cao et al., 2019), GroupDRO (Sagawa et al., 2019).

5.2. MAPLE on IRM

[IRM on ColoredMNIST]. For different vanilla IRM method (IMRv1 or REx) to be compared, we ultilize the same IRM loss as the outer objective in MAPLE. We vary the number of training sample size and model parameters to demonstrate the general applicability to various scales in practice. We add a number to the end of the dataset name to indicate the training set size. We split 10% training data as the validation dataset.

From Figure 2, vanilla IRM methods still lags behind the Oracle performance by a large margin. The gap becomes more prominent when the model is more overparameterized. MRM further boosts the generalization performance while its performance is still limited by regularization-based IRM training paradigm. MAPLE transforms the search space of model parameters into that of sample weights and searches for the optimal sample weights on training dataset, and further guides the optimization by evaluating the criterion on the learned model. MAPLE beats these latest competitive baselines by a large margin and even approaches the performance of Oracle.

Table 1. Comparison of Top-1 Test Accuracy on ResNet-18 on ColoredObject and CIFARMNIST.

Dataset		ColoredObject	CIFARMNIST
Oracle		87.9 ± 0.3	83.7 ± 1.5
ERM		51.6 ± 0.5	39.5 ± 0.4
BayesianIRM		78.1 ± 0.6	59.3 ± 0.8
IRMv1b	IRM	72.5 ± 2.1	51.3 ± 3.0
	MRM	58.4 ± 0.9	56.7 ± 2.3
	SparseIRM	87.4 ± 0.6	63.9 ± 0.4
	MAPLE	$\mathbf{87.4 \pm 0.5}$	$\mathbf{82.9 \pm 0.4}$
REx	IRM	73.8 ± 1.3	50.1 ± 2.2
	MRM	55.7 ± 2.9	52.6 ± 1.5
	SparseIRM	80.3 ± 1.1	62.7 ± 0.6
	MAPLE	$\mathbf{86.9 \pm 1.0}$	$\mathbf{82.5 \pm 0.7}$

[IRM on ColoredObject and CIFRAMNIST]. In this section, we evaluate the performance of MAPLE on ColoredOb-

Table 2. Comparison of MAPLE and state-of-the-art DRO methods in Waterbirds and CelebA. MAPLE surpasses previous methods without group annotations by a large margin and even achieves comparable or even better performance than GroupDRO and Upweighting, which utilize the group annotation for training samples.

Method	Group annotations for training samples?	Waterbirds		CelebA	
		Average	Worst-group	Average	Worst-group
Upweighting (Cui et al., 2019)	Yes	92.2	87.4	89.3	83.3
GroupDRO (Sagawa et al., 2019)	Yes	93.5	91.4	92.9	88.9
ERM	No	97.3	72.6	95.6	47.2
CVaR DRO (Levy et al., 2020)	No	96.0	75.9	82.4	64.4
LfF (Nam et al., 2020)	No	91.2	78.0	86.0	70.6
JTT (Liu et al., 2021a)	No	93.3	86.7	88.0	81.1
MAPLE	No	**92.9**	**91.7**	**89.0**	**88.0**

ject and CIFARMNIST with large-sized model ResNet-18 in Table 1. We split 10% training data as the validation dataset. MAPLE consistently beats the baselines by a large margin and achieves performance approaching Oracle. These results validate the effectivenss of MAPLE on more modern ResNet architecture and diverse tasks. Notably MAPLE surpasses vanilla IRM method by over 30% percent in the CIFARMNIST dataset. It shows that in more challenging scenarios MAPLE can outperform IRM by a larger margin.

5.3. DRO on Waterbirds and CelebA

In this section, we further validate the effectiveness of MAPLE when applied to DRO. The worst-group accuracy is taken as the core criterion to evaluate the effectiveness of DRO methods. In this experiment, we adopt the CVaR DRO objective. To be noted, our bilevel formulation doesn't rely on the group annotations on training samples. We set the α to be 20% and we find it serves as a good threshold without hyperparameter search.

From the Table 2, we find that MAPLE beats previous state-of-the-art method JTT without group annotations on training samples by 5% in Waterbirds and 6.9% in CelebA. This can be expected as JTT upweights mis-classified training samples by a mannually-searched magnitude, by evaluating a checkpoint obtained from ERM training at a manually-searched epoch. This inevitably leads to suboptimal performance due to its cumbersome criterion of just upweighting the misclassified training samples at a specific epoch rather than considering more globally is imperfect. To be totally contrary, MAPLE ultilizes the CVaR DRO criterion evaluated on validation set to consider the problem more reasonably by gradually optimizing the sample weights through evaluating the model learned from current sample weights step by step. Upweighting simply upweights the rare groups inversely to its portion in the whole dataset and ignores the importance differed from sample to sample in the same group. GroupDRO makes further improvement to Upweighting by regularziation term and is generally considered as a upperbound by previous works (Liu et al., 2021a). MAPLE surpasses Upweighting by 4.3% and GroupDRO by 0.3% in Waterbirds and surpasses Upweighting by 4.7% in CelebA demonstrating the effectiveness of MAPLE in more complex DRO setting.

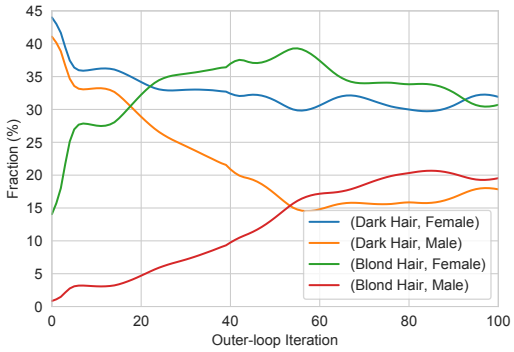


Figure 3. Training dynamics of each group weight fraction for ResNet-50 on CelebA. The weight fraction of (Blond Hair, Male) and (Dark Hair, Male) changes to 20%. The weight fraction of (Dark Hair, Female) and (Blond Hair, Female) changes to 30%. This indicates that MAPLE can automatically adjust the weight fraction of different groups and the weight fraction of four groups need not be the same.

5.4. Further Analysis

[Training dynamics of the weights of different groups] We plot the training dynamics of sample weight fraction in CelebA experiment in Figure 3. Initially all the weights of different samples are initilized as 1. As there are scarce training samples in group (Blond Hair, Male), its weight fraction is initially only 0.085%. After 100 iterations of updates, the weight fraction of (Blond Hair, Male) gradually comes up to approximately 20%. Concurrently, the weight fraction of group (Dark Hair, Male) goes down to approxi-

mately 20% and the weight fraction of (Dark Hair, Female) and (Blond Hair, Female) both come to approximately 30%. This demonstrates that we need not upweight each group to the same importance level, which indicates one reason why Upweighting fails behind MAPLE.

[Weight Distributions of Four Groups] We further plot the histogram of samples weights in Figure 4 for four groups in CelebA experiment at the end of training. It indicates that the weights of group (Blond Hair, Female) flattens to around 30. This is consistent with our primal goal to upweight the group with few training samples, and MAPLE sucessfully achieve this without any training group annotations. We also discovers that the sample weight assigned to different training samples need not be the same. This demonstrates another reason why MAPLE beats JTT and Upweighting by a large margin.

6. Conclusion

In this work, we present a model agnostic sample reweighing method named MAPLE for out-of-domain learning. We propose a novel bilevel optimization framework to learn sample weights to address the out-of-domain learning problem effectively. We further enhance MAPLE with sparsity to improve training speed. We present theorectical analysis in linear case and demonstrate its superior performance various tasks and models.

Acknowledgements

XZ, YL, RP, WZ and TZ acknowledge the funding supported by GRF 16201320. RX and PC acknowledge the funding supported by National Key R&D Program of China (No. 2018AAA0102004), National Natural Science Foundation of China (No. 62141607, U1936219).

References

Ahmed, F., Bengio, Y., van Seijen, H., and Courville, A. Systematic generalisation with group invariant predictions. In *International Conference on Learning Representations*, 2020.

Ahuja, K., Shanmugam, K., Varshney, K., and Dhurandhar, A. Invariant risk minimization games. In *International Conference on Machine Learning*, pp. 145–155. PMLR, 2020.

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

Bai, H., Sun, R., Hong, L., Zhou, F., Ye, N., Ye, H.-J., Chan, S.-H. G., and Li, Z. Decaug: Out-of-distribution generalization via decomposed feature representation and semantic augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6705–6713, 2021a.

Bai, H., Zhou, F., Hong, L., Ye, N., Chan, S.-H. G., and Li, Z. Nas-ood: Neural architecture search for out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8320–8329, 2021b.

Beery, S., Van Horn, G., and Perona, P. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 456–473, 2018.

Ben-Tal, A., Den Hertog, D., De Waegenaere, A., Melenberg, B., and Rennen, G. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013.

Borsos, Z., Mutnỳ, M., and Krause, A. Coresets via bilevel optimization for continual learning and streaming. *arXiv preprint arXiv:2006.03875*, 2020.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. Learning imbalanced datasets with label-distribution-aware margin loss, 2019.

Chang, S., Zhang, Y., Yu, M., and Jaakkola, T. Invariant rationalization. In *International Conference on Machine Learning*, pp. 1448–1458. PMLR, 2020.

Chen, K., Hong, L., Xu, H., Li, Z., and Yeung, D.-Y. Multisiam: Self-supervised multi-instance siamese representation learning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7546–7554, 2021a.

Chen, Y., Rosenfeld, E., Sellke, M., Ma, T., and Risteski, A. Iterative feature matching: Toward provable domain generalization with logarithmic environments. *arXiv preprint arXiv:2106.09913*, 2021b.

Cherepanova, V., Nanda, V., Goldblum, M., Dickerson, J. P., and Goldstein, T. Technical challenges for training fair neural networks. *arXiv preprint arXiv:2102.06764*, 2021.

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. Class-balanced loss based on effective number of samples, 2019.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Diao, S., Bai, J., Song, Y., Zhang, T., and Wang, Y. Zen: Pre-training chinese text encoder enhanced by n-gram representations. *arXiv preprint arXiv:1911.00720*, 2019.

Diao, S., Xu, R., Su, H., Jiang, Y., Song, Y., and Zhang, T. Taming pre-trained language models with n-gram representations for low-resource domain adaptation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3336–3349, 2021.

Duchi, J. and Namkoong, H. Variance-based regularization with convex objectives. *The Journal of Machine Learning Research*, 20(1):2450–2504, 2019.

Duchi, J. C. and Namkoong, H. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378–1406, 2021.

Duchi, J. C., Hashimoto, T., and Namkoong, H. Distributionally robust losses against mixture covariate shifts. *Under review*, 2019.

Duchi, J. C., Glynn, P. W., and Namkoong, H. Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research*, 2021.

Fang, T., Lu, N., Niu, G., and Sugiyama, M. Rethinking importance weighting for deep learning under distribution shift. *Advances in Neural Information Processing Systems*, 33, 2020.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135. PMLR, 2017.

Gao, J., Xu, H., Ren, X., Yu, P. L., Liang, X., Jiang, X., Li, Z., et al. Autobert-zero: Evolving bert backbone from scratch. *arXiv preprint arXiv:2107.07445*, 2021.

Gao, J., Zhou, Y., Yu, P. L., Joty, S., and Gu, J. Unison: Unpaired cross-lingual image captioning. 2022.

Goyal, A., Lamb, A., Hoffmann, J., Sodhani, S., Levine, S., Bengio, Y., and Schölkopf, B. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893*, 2019.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., and others. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv:2006.07733*, 2020.

Gu, J., Cai, J., Joty, S. R., Niu, L., and Wang, G. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7181–7189, 2018.

Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., and Smola, A. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19:601–608, 2006.

Huang, M., Huang, Z., Li, C., Chen, X., Xu, H., Li, Z., and Liang, X. Arch-graph: Acyclic architecture relation predictor for task-transferable neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11881–11891, 2022.

Kanamori, T., Hido, S., and Sugiyama, M. A least-squares approach to direct importance estimation. *The Journal of Machine Learning Research*, 10:1391–1445, 2009.

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., and Courville, A. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pp. 5815–5826. PMLR, 2021a.

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., and Courville, A. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pp. 5815–5826. PMLR, 2021b.

Kuang, K., Xiong, R., Cui, P., Athey, S., and Li, B. Stable prediction with model misspecification and agnostic distribution shift. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4485–4492, 2020.

Levy, D., Carmon, Y., Duchi, J. C., and Sidford, A. Large-scale methods for distributionally robust optimization, 2020.

Lin, Y., Lian, Q., and Zhang, T. An empirical study of invariant risk minimization on deep models. *ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning*, 2021.

Lin, Y., Dong, H., Wang, H., and Zhang, T. Bayesian invariant risk minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16021–16030, 2022a.

Lin, Y., Zhu, S., and Cui, P. Zin: When and how to learn invariance by environment inference? *arXiv preprint arXiv:2203.05818*, 2022b.

Liu, E. Z., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., Liang, P., and Finn, C. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pp. 6781–6792. PMLR, 2021a.

Liu, H., Simonyan, K., and Yang, Y. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

Liu, J., Hu, Z., Cui, P., Li, B., and Shen, Z. Heterogeneous risk minimization. *arXiv preprint arXiv:2105.03818*, 2021b.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild, 2015.

Liu, Z., Han, J., Chen, K., Hong, L., Xu, H., Xu, C., and Li, Z. Task-customized self-supervised pre-training with scalable dynamic routing. In *AAAI*, 2022.

Lorraine, J., Vicol, P., and Duvenaud, D. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, pp. 1540–1552. PMLR, 2020.

Louizos, C., Welling, M., and Kingma, D. P. Learning sparse neural networks through l0 regularization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1Y8hhg0b.

Luo, P., Wang, X., Shao, W., and Peng, Z. Towards understanding regularization in batch normalization. *arXiv preprint arXiv:1809.00846*, 2018.

MacKay, M., Vicol, P., Lorraine, J., Duvenaud, D., and Grosse, R. Self-tuning networks: Bilevel optimization of hyperparameters using structured best-response functions. *arXiv preprint arXiv:1903.03088*, 2019.

Maclaurin, D., Duvenaud, D., and Adams, R. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pp. 2113–2122. PMLR, 2015.

Nam, J., Cha, H., Ahn, S., Lee, J., and Shin, J. Learning from failure: Training debiased classifier from biased classifier, 2020.

Nichol, A. and Schulman, J. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

Oren, Y., Sagawa, S., Hashimoto, T. B., and Liang, P. Distributionally robust language modeling. *arXiv preprint arXiv:1909.02060*, 2019.

Paulus, M. B., Maddison, C. J., and Krause, A. Rao-blackwellizing the straight-through gumbel-softmax gradient estimator. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Mk6PZtgAgfq.

Pearl, J. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

Pedregosa, F. Hyperparameter optimization with approximate gradient. In *International conference on machine learning*, pp. 737–746. PMLR, 2016.

Peters, J., Bühlmann, P., and Meinshausen, N. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pp. 947–1012, 2016.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pp. 4334–4343. PMLR, 2018.

Rockafellar, R. T., Uryasev, S., et al. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

Rosenfeld, E., Ravikumar, P., and Risteski, A. The risks of invariant risk minimization. *arXiv preprint arXiv:2010.05761*, 2020.

Rothenhäusler, D., Meinshausen, N., Bühlmann, P., and Peters, J. Anchor regression: Heterogeneous data meet causality. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 83(2):215–246, 2021.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Sagawa, S., Raghunathan, A., Koh, P. W., and Liang, P. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pp. 8346–8356. PMLR, 2020.

Shaban, A., Cheng, C.-A., Hatch, N., and Boots, B. Truncated back-propagation for bilevel optimization, 2019.

Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. The pitfalls of simplicity bias in neural networks. *arXiv preprint arXiv:2006.07710*, 2020.

Shao, W., Meng, T., Li, J., Zhang, R., Li, Y., Wang, X., and Luo, P. Ssn: Learning sparse switchable normalization via sparsestmax. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Shen, Z., Cui, P., Zhang, T., and Kunag, K. Stable learning via sample reweighting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5692–5699, 2020.

Shen, Z., Liu, J., He, Y., Zhang, X., Xu, R., Yu, H., and Cui, P. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.

Shi, H., Pi, R., Xu, H., Li, Z., Kwok, J., and Zhang, T. Bridging the gap between sample-based and one-shot neural architecture search with bonas. *Advances in Neural Information Processing Systems*, 33:1808–1819, 2020.

Shi, H., Gao, J., Ren, X., Xu, H., Liang, X., Li, Z., and Kwok, J. T.-Y. Sparsebert: Rethinking the importance analysis in self-attention. In *International Conference on Machine Learning*, pp. 9547–9557. PMLR, 2021.

Shimodaira, H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., and Meng, D. Meta-weight-net: Learning an explicit mapping for sample weighting. *arXiv preprint arXiv:1902.07379*, 2019.

Sinha, A., Malo, P., and Deb, K. A review on bilevel optimization: from classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, 22(2):276–295, 2017.

Strobl, E. V. and Visweswaran, S. Markov boundary discovery with ridge regularized linear models. *Journal of Causal inference*, 4(1):31–48, 2016.

Sugiyama, M., Krauledat, M., and Müller, K.-R. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5), 2007.

Sugiyama, M., Suzuki, T., Nakajima, S., Kashima, H., von Bünau, P., and Kawanabe, M. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Wang, H., Wu, Z., and He, J. Training fair deep neural networks by balancing influence. *arXiv preprint arXiv:2201.05759*, 2022.

Xie, C., Chen, F., Liu, Y., and Li, Z. Risk variance penalization: From distributional robustness to causality. *arXiv e-prints*, pp. arXiv–2006, 2020.

Xu, R., Cui, P., Kuang, K., Li, B., Zhou, L., Shen, Z., and Cui, W. Algorithmic decision making with conditional fairness. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2125–2135, 2020.

Xu, R., Cui, P., Shen, Z., Zhang, X., and Zhang, T. Why stable learning works? a theory of covariate shift generalization. *arXiv preprint arXiv:2111.02355*, 2021.

Xu, Y. and Jaakkola, T. Learning representations that support robust transfer of predictors. *arXiv preprint arXiv:2110.09940*, 2021.

Xu, Y., Xie, L., Zhang, X., Chen, X., Qi, G.-J., Tian, Q., and Xiong, H. Pc-darts: Partial channel connections for memory-efficient architecture search. *arXiv preprint arXiv:1907.05737*, 2019.

Yao, L., Pi, R., Xu, H., Zhang, W., Li, Z., and Zhang, T. G-detkd: Towards general distillation framework for object detectors via contrastive and semantic-guided feature imitation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3591–3600, 2021a.

Yao, L., Pi, R., Xu, H., Zhang, W., Li, Z., and Zhang, T. Joint-detnas: Upgrade your detector with nas, pruning and dynamic distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10175–10184, 2021b.

Ye, N., Li, K., Bai, H., Yu, R., Hong, L., Zhou, F., Li, Z., and Zhu, J. Ood-bench: Quantifying and understanding two dimensions of out-of-distribution generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7947–7958, 2022.

Yong Lin, Qing Lian, T. Z. An empirical study of invariant risk minimization on deep models. *preprints*, 2021.

Zhang, D., Ahuja, K., Xu, Y., Wang, Y., and Courville, A. Can subnetwork structure be the key to out-of-distribution generalization? *arXiv preprint arXiv:2106.02890*, 2021a.

Zhang, X., Cui, P., Xu, R., Zhou, L., He, Y., and Shen, Z. Deep stable learning for out-of-distribution generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5372–5382, 2021b.

Zhang, X., Xu, Z., Xu, R., Liu, J., Cui, P., Wan, W., Sun, C., and Li, C. Towards domain generalization in object detection. *arXiv preprint arXiv:2203.14387*, 2022.

Zhou, W., Zeng, Y., Diao, S., and Zhang, X. Vlue: A multi-task benchmark for evaluating vision-language models, 2022a. URL https://arxiv.org/abs/2205.15237.

Zhou, X., Zhang, W., Chen, Z., Diao, S., and Zhang, T. Efficient neural network training via forward and backward propagation sparsification. *Advances in Neural Information Processing Systems*, 34:15216–15229, 2021a.

Zhou, X., Zhang, W., Xu, H., and Zhang, T. Effective sparsification of neural networks with global sparsity constraint. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3599–3608, 2021b.

Zhou, X., Lin, Y., Zhang, W., and Zhang, T. Sparse invariant risk minimization. In *International Conference on Machine Learning*. PMLR, 2022b.

Zhou, X., Pi, R., Zhang, W., Lin, Y., and Zhang, T. Probabilistic bilevel coreset selection. In *International Conference on Machine Learning*. PMLR, 2022c.

Zou, F., Shen, L., Jie, Z., Zhang, W., and Liu, W. A sufficient condition for convergences of adam and rmsprop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11127–11135, 2019.

Supplementary Materials:
Model Agnostic Sample Reweighting for Out-of-Distribution Learning

This appendix can be divided into the following parts:

1. Section A gives the details of datasets in IRM and DRO.
2. Section B presents experimental configurations of this work.
3. Section C presents experiments on weight distributions of different groups to show the ability of MAPLE to identify weights for each training samples.
4. Section D presents experiments on the effectivenss of improving training speed via sparsity constraint on training sample size.
5. Section E presents experiments on validation of transferability of sample weights.
6. Section F presents proof of Theorem 4.5
7. Section G presents proof of Theorem 4.6
8. Section H introduces related works on bi-level optimization.
9. Section I presents discussions on future works.

A. Dataset Details

ColoredMNIST is the most widely used benchmark in IRM and ColoredObject, CIFARMNIST are adopted to showcase the superior performance of MAPLE on more challenging largescale settings. The labels for IRM datasets are generated from the core features. The spurious features have strong correlations with the labels in the training set but the correlation reverses in the testing set. In each dataset there exist two training environments and one testing environment with different correlations. We combine the correlations of two training environments and one testing environment into a correlation tuple. Label noise is added to the datasets to make the task more challenging (Arjovsky et al., 2019; Zhang et al., 2021a).

Waterbirds and CelebA are real-world datasets and we adopt them to demonstrate the generalizability of MAPLE to real-world scenarios. Waterbirds and CelebA are both binary prediction tasks. In each dataset, there exists a binary spurious feature highly correlated with the label. We follow the challenging setting of Liu et al. (2021a) that no group annotation is provided in the training dataset and group annotations are provided in the small validation set.

ColoredMNIST (Arjovsky et al., 2019). It contains images from MNIST and the images are labeled as 0 or 1. Each image is attached with a color as the spurious feature. Correlation tuple is $(0.9, 0.8, 0.1)$. Noise ratio is 25%.

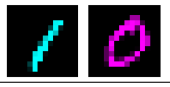
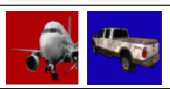
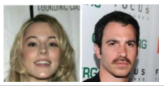
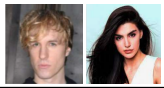
ColoredObject (Ahmed et al., 2020; Zhang et al., 2021a). It is constructed by extracting 8 classes of objects from MSCOCO and put them onto colored backgrounds. Correlation tuple is $(0.999, 0.7, 0.1)$. Noise ratio is 5%.

CIFARMNIST (Shah et al., 2020; Lin et al., 2021). It is constructed by concatenating images of CIFAR10 with MNIST. The CIFAR images are the invariant features and the MNIST images are the spurious features. Correlation tuple is $(0.999, 0.7, 0.1)$. Noise ratio is 10%.

Waterbirds (Wah et al., 2011; Sagawa et al., 2019). The Waterbirds dataset contains two group of birds, i.e., {waterbird, landbird}. There are two kinds of background, i.e., {water background, land background}. The background type is spuriously correlated with the bird type. No background annotation is provided in the training dataset.

CelebA (Liu et al., 2015; Sagawa et al., 2019). In the CelebA dataset, the task is to predict hair color, {blond, dark}, based on the image input. The attribute gender, {male, female}, is spuriously correlated with the hair color.

Table 3. Illustration of each dataset. Core and Spurious stand for the core and spurious features, respectively. Spurious features are highly correlated with the label. However, the correlations are reversed in the testing samples to simulate the distributional shift.

Dataset	Core	Spurious	Training	Testing
ColoredMNIST	Digit	Color		
ColoredObject	Object	Background		
CIFARMNIST	CIFAR	MNIST		
Waterbirds	Bird	Background		
CelebA	Hair Color	Gender		

B. Experimental Configurations

Table 4. Experimental Configurations of MAPLE. The hyperparameters of sample weight and probability optimization are obtained via grid search on validation set on ColoredMNIST and applied directly to other scenarios. The demonstrates the robustness of MAPLE to different settings. We directly takes the regular training recipe for ERM training as the hyperparameters of inner loop model parameter optimization. We early stop in the inner loop as we find that training for such schedule is enough to obtain approximately best performance in validation set.

Dataset	ColoredMNIST	CIFARMNIST	ColoredObject	Waterbirds	CelebA
GPUs	1	1	1	1	8
Batch Size	50000	1000	1000	128	1024
Outer Iterations	100	100	100	50	100
Inner Training Schedule	100 iterations	100 iterations	100 iterations	3 epochs	1 epoch
Sample Weight Optimizer	Adam	Adam	Adam	Adam	Adam
Sample Weight Learning Rate	0.25	0.25	0.25	0.25	0.25
Sample Probability Optimizer	Adam	Adam	Adam	Adam	Adam
Sample Probability Learning Rate	5e-2	5e-2	5e-2	5e-2	5e-2
Model Parameter Optimizer	SGD	SGD	SGD	SGD	SGD
Model Parameter Learning Rate	1e-1	1e-2	1e-2	1e-4	1e-4
Model Parameter Weight Decay	1e-1	1e-2	1e-2	1e-1	1e-2

C. Weight Distributions of Different Groups

We further plot the histogram of samples weights in Figure 4 for four groups in CelebA experiment at the end of training. It indicates that the weights of group (Blond Hair, Female) flattens to around 30, while the weights of other groups still

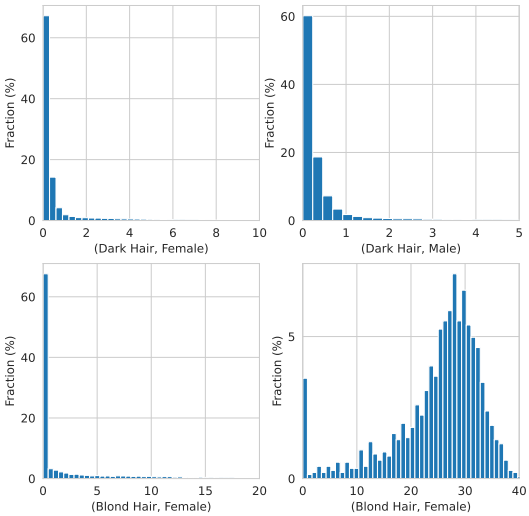


Figure 4. Histogram of weights for four groups in CelebA. MAPLE automatically upweights the weights of group (Blond Hair, Female) and the histograms demonstrate that the weight assigned to different groups need not be the same.

remainly lies around 1. This is consistent with our primal goal to upweight the group with few training samples, and MAPLE sucessfully achieve this without any training group annotations. We also discovers that the sample weight assigned to different training samples need not be the same. This demonstrates another reason why MAPLE beats JTT and Upweighting by a large margin.

D. Effectiveness of Sparsity in Promoting Training Speed

Table 5 demonstrates the comparison of training speed between MAPLE with no sparsity constraint on sample sizes and MAPLE. MAPLE saves a lot of inner loop computation time.

Table 5. Comparing computational time of inner loop of different methods on Waterbirds. MAPLE(NS) indicates MAPLE with no sparsity constraint.

Method	MAPLE(NS)	MAPLE
GPU Hours	8.43	**6.74**

E. Validation of Transferability of Sample Weights

We transfer the sample weights searched via ResNet-18 and directly apply it to train the weighted training samples on ResNet-50. Table 6 demonstrates that the searched sample weights on ResNet-18 can be successfully applied to perform weighted ERM training on ResNet-50, even with slight performance boost.

Table 6. Validating transferability of sample weights on Waterbirds, from ResNet-18 on seaching phase and ResNet-50 on downstream weighted training phase.

Sample Weights Searched on ResNet-18	Weighted ERM Training on ResNet-18	Weighted ERM Training on ResNet-50
Worst-group Acc	91.2%	**91.6%**

F. Proof of Theorem 4.5

By Assumption 4.3, $\mathbb{P}(\boldsymbol{y}, \boldsymbol{z}_c, \boldsymbol{z}_s) > 0$. Then we can define the following weighting function

$$w(\boldsymbol{y}, \boldsymbol{x}) := \frac{\mathbb{P}(\boldsymbol{y}, \boldsymbol{z}_c)\mathbb{P}(\boldsymbol{z}_s)}{\mathbb{P}(\boldsymbol{y}, \boldsymbol{x})} \tag{20}$$

Below, we will show that $w(\boldsymbol{x}, y)$ is the desired weight function, and the solution of this ordinary least square re-weighted by $w(\boldsymbol{x}, y)$ is the optimal debiased predictor $\bar{\boldsymbol{\theta}}$. Specifically,

$$\mathcal{L}(\boldsymbol{\theta}; w) = \int (y - \boldsymbol{x}^\intercal \boldsymbol{\theta})^2 \mathbb{P}_w(\boldsymbol{x}, y) d\boldsymbol{x} dy, \tag{21}$$

It is easy to know that the minimizer of Eqn. (21).

$$\begin{aligned}
\boldsymbol{\theta}^*(w) &= \left(\int \boldsymbol{x}\boldsymbol{x}^\top \mathbb{P}_w(\boldsymbol{x}, y) d\boldsymbol{x} dy \right)^{-1} \int \boldsymbol{x} y \mathbb{P}_w(\boldsymbol{x}, y) d\boldsymbol{x} dy \\
&= \left(\int \boldsymbol{S}\boldsymbol{z}\boldsymbol{z}^\top \boldsymbol{S}^\top \mathbb{P}_w(\boldsymbol{x}, y) d\boldsymbol{x} dy \right)^{-1} \int \boldsymbol{S} \boldsymbol{z} y \mathbb{P}_w(\boldsymbol{z}, y) d\boldsymbol{x} dy \\
&= (\boldsymbol{S}^\top)^{-1} \left(\int \boldsymbol{z}\boldsymbol{z}^\top \mathbb{P}_w(\boldsymbol{z}, y) d\boldsymbol{z} dy \right)^{-1} \int \boldsymbol{z} y \mathbb{P}_w(\boldsymbol{z}, y) d\boldsymbol{z} dy \\
&= (\boldsymbol{T}^\top) \left(\int \boldsymbol{z}\boldsymbol{z}^\top \mathbb{P}_w(\boldsymbol{z}, y) d\boldsymbol{z} dy \right)^{-1} \int \boldsymbol{z} y \mathbb{P}_w(\boldsymbol{z}, y) d\boldsymbol{z} dy.
\end{aligned}$$

At last, we are going to show $\left(\int \boldsymbol{z}\boldsymbol{z}^\top \mathbb{P}_w(\boldsymbol{z}, y) d\boldsymbol{z} dy \right)^{-1} \int \boldsymbol{z} y \mathbb{P}_w(\boldsymbol{z}, y) d\boldsymbol{z} dy$ will be equal to $[\bar{\boldsymbol{\theta}}_c; \boldsymbol{0}]$ as defined in Definition 4.2.

Proof. We denote $\Sigma^w = \int \boldsymbol{x}\boldsymbol{x}^\top \mathbb{P}_w(\boldsymbol{x}, y) d\boldsymbol{x} dy$, and turn to simplify $\theta^*(w)$ by computing Σ^w and Cov^w

It follows that

$$\mathbb{P}_w(\boldsymbol{y}, \boldsymbol{z}_c, \boldsymbol{z}_s) = \mathbb{P}_w(\boldsymbol{y}, \boldsymbol{x}) = w(\boldsymbol{y}, \boldsymbol{x})\mathbb{P}(\boldsymbol{y}, \boldsymbol{x}) = \mathbb{P}(\boldsymbol{y}, \boldsymbol{z}_c)\mathbb{P}(\boldsymbol{z}_s).$$

It is easy to see $\mathbb{P}_w(\boldsymbol{y}, \boldsymbol{z}_c) = \mathbb{P}(\boldsymbol{y}, \boldsymbol{z}_c)$ and $\mathbb{P}_w(\boldsymbol{z}_s) = \mathbb{P}(\boldsymbol{z}_s)$ because

$$\begin{aligned}
\mathbb{P}_w(\boldsymbol{y}, \boldsymbol{z}_c) &= \int_{\boldsymbol{z}_s} \mathbb{P}_w(\boldsymbol{y}, \boldsymbol{z}_c, \boldsymbol{z}_s) = \int_{\boldsymbol{z}_s} \mathbb{P}(\boldsymbol{y}, \boldsymbol{z}_c)\mathbb{P}(\boldsymbol{z}_s) = \mathbb{P}(\boldsymbol{y}, \boldsymbol{z}_c) \int_{\boldsymbol{z}_s} \mathbb{P}(\boldsymbol{z}_s) = \mathbb{P}(\boldsymbol{y}, \boldsymbol{z}_c) \\
\mathbb{P}_w(\boldsymbol{z}_s) &= \int_{\boldsymbol{y}, \boldsymbol{z}_c} \mathbb{P}_w(\boldsymbol{y}, \boldsymbol{z}_c, \boldsymbol{z}_s) = \int_{\boldsymbol{y}, \boldsymbol{z}_c} \mathbb{P}(\boldsymbol{y}, \boldsymbol{z}_c)\mathbb{P}(\boldsymbol{z}_s) = \mathbb{P}(\boldsymbol{z}_s) \int_{\boldsymbol{y}, \boldsymbol{z}_c} \mathbb{P}(\boldsymbol{y}, \boldsymbol{z}_c) = \mathbb{P}(\boldsymbol{z}_s)
\end{aligned}$$

So we further have

$$P_w(\boldsymbol{y}, \boldsymbol{z}_c, \boldsymbol{z}_s) = P(\boldsymbol{y}, \boldsymbol{z}_c)P(\boldsymbol{z}_s) = \mathbb{P}_w(\boldsymbol{y}, \boldsymbol{z}_c)\mathbb{P}_w(\boldsymbol{z}_s).$$

It also leads to

$$\begin{aligned}
P_w(\boldsymbol{z}_c, \boldsymbol{z}_s) &= \mathbb{P}_w(\boldsymbol{z}_c)\mathbb{P}_w(\boldsymbol{z}_s) \\
P_w(\boldsymbol{y}, \boldsymbol{z}_s) &= \mathbb{P}_w(\boldsymbol{y})\mathbb{P}_w(\boldsymbol{z}_s)
\end{aligned}$$

It follows that

$$\begin{aligned}
\Sigma_c^w &:= \mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}_c\boldsymbol{z}_c^\intercal] = \int \boldsymbol{z}_c\boldsymbol{z}_c^\intercal \mathbb{P}_w(\boldsymbol{x}, y) = \int \boldsymbol{z}_c\boldsymbol{z}_c^\intercal \mathbb{P}_w(\boldsymbol{z}, y) = \int \boldsymbol{z}_c\boldsymbol{z}_c^\intercal \mathbb{P}_w(\boldsymbol{z}_c) = \int \boldsymbol{z}_c\boldsymbol{z}_c^\intercal \mathbb{P}(\boldsymbol{z}_c) = \Sigma_c \\
\Sigma_b^w &:= \mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}_s\boldsymbol{z}_s^\intercal] = \int \boldsymbol{z}_s\boldsymbol{z}_s^\intercal \mathbb{P}_w(\boldsymbol{x}, y) = \int \boldsymbol{z}_s\boldsymbol{z}_s^\intercal \mathbb{P}_w(\boldsymbol{z}, y) = \int \boldsymbol{z}_s\boldsymbol{z}_s^\intercal \mathbb{P}_w(\boldsymbol{z}_s) = \int \boldsymbol{z}_s\boldsymbol{z}_s^\intercal \mathbb{P}(\boldsymbol{z}_s) = \Sigma_b
\end{aligned}$$

Furthermore,

$$\mathrm{Cov}^w(\boldsymbol{z}_c, \boldsymbol{z}_s)$$

$$
\begin{aligned}
&= \mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}_c^\intercal \boldsymbol{z}_s] - \mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}_c]^\intercal \mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}_s] \\
&= \int \mathbb{P}_w(\boldsymbol{z}_c, \boldsymbol{z}_s)\boldsymbol{z}_c^\intercal \boldsymbol{z}_s d\boldsymbol{z}_c d\boldsymbol{z}_s - \mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}_c]^\intercal \mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}_s] \\
&= \int \mathbb{P}_w(\boldsymbol{z}_c)\mathbb{P}_w(\boldsymbol{z}_s)\boldsymbol{z}_c^\intercal \boldsymbol{z}_s d\boldsymbol{z}_c d\boldsymbol{z}_s - \mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}_c]^\intercal \mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}_s] \\
&= \mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}_c]^\intercal \mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}_s] - \mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}_c]^\intercal \mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}_s] = \mathbf{0}.
\end{aligned}
$$

Similarly, we can obtain

$$
\mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}_c y] = \mathbb{E}[\boldsymbol{z}_c y], \quad \mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}_s y] = \mathbf{0}.
$$

Putting these together, we have

$$
\begin{aligned}
\Sigma^w &= \begin{bmatrix} \Sigma_c^w & \mathrm{Cov}^w(\boldsymbol{z}_c, \boldsymbol{z}_s) \\ \mathrm{Cov}^w(\boldsymbol{z}_s, \boldsymbol{z}_c) & \Sigma_s^w \end{bmatrix} = \begin{bmatrix} \Sigma_b & \mathbf{0} \\ \mathbf{0} & \Sigma_c \end{bmatrix}, \\
\mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}y] &= \begin{bmatrix} \mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}_c y] \\ \mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}_s y] \end{bmatrix} = \begin{bmatrix} \mathbb{E}[\boldsymbol{z}_c y] \\ 0 \end{bmatrix}.
\end{aligned}
$$

Then

$$
\boldsymbol{\theta}^*(w) = (\Sigma^w)^{-1}\mathbb{E}[w(\boldsymbol{x}, y)\boldsymbol{z}y] = \begin{bmatrix} \Sigma_c & \mathbf{0} \\ \mathbf{0} & \Sigma_b \end{bmatrix}^{-1} \begin{bmatrix} \mathbb{E}[\boldsymbol{z}_c y] \\ 0 \end{bmatrix} = \begin{bmatrix} \Sigma_c^{-1}\mathbb{E}[\boldsymbol{z}_c y] \\ 0 \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{\theta}}_c \\ 0 \end{bmatrix} = \bar{\boldsymbol{\theta}}.
$$

The second part proof is straightforward. By Assumption 4.4, for any $\theta \neq \theta^* = \theta_w$, we have

$$
\mathcal{R}(\theta) > \mathcal{R}(\bar{\boldsymbol{\theta}}) = \mathcal{R}(\boldsymbol{\theta}^*(w)). \tag{22}
$$

We already know that $\boldsymbol{\theta}^*(w)$ is in the feasible solution of MAR. Eq. (22) further shows that $\boldsymbol{\theta}^*(w)$ achieves the minimum loss of $\mathcal{R}$. Putting these together, we conclude that MAR uniquely identify $\bar{\theta}$. □

G. Proof of Theorem 4.6

By the bounded difference inequality (Corollary 2.21 of (Wainwright, 2019)), given any $\boldsymbol{w}$, we have with probability $1 - \delta/2$,

$$
\mathcal{R}(\hat{\boldsymbol{\theta}}^*(\boldsymbol{w}); \mathcal{D}_v) \leq \mathcal{R}(\hat{\boldsymbol{\theta}}^*(\boldsymbol{w})) + M\sqrt{\frac{\ln(2/\delta)}{2N}}, \tag{23}
$$

where $\mathcal{R}(\hat{\boldsymbol{\theta}}^*(\boldsymbol{w}); \mathcal{D}_v)$ is the OOD risk on the validation dataset $\mathcal{D}_v$ and $\mathcal{R}(\hat{\boldsymbol{\theta}}^*(\boldsymbol{w}))$ is the population OOD risk. Then we have with probability $1 - \delta$,

$$
\begin{aligned}
&\mathcal{R}(\hat{\boldsymbol{\theta}}^*(\hat{\boldsymbol{w}})) \\
\leq &\mathcal{R}(\hat{\boldsymbol{\theta}}^*(\hat{\boldsymbol{w}}); \mathcal{D}_v) + M\sqrt{\frac{2\ln(2|\mathcal{W}|/\delta)}{N}} \\
\leq &\mathcal{R}(\hat{\boldsymbol{\theta}}^*(\boldsymbol{w}); \mathcal{D}_v) + M\sqrt{\frac{\ln(2|\mathcal{W}|/\delta)}{2N}} + \epsilon \\
\leq &\mathcal{R}(\hat{\boldsymbol{\theta}}^*(\boldsymbol{w})) + M\sqrt{\frac{\ln(2/\delta)}{2N}} + M\sqrt{\frac{\ln(2|\mathcal{W}|/\delta)}{2N}} + \epsilon \\
\leq &\mathcal{R}(\hat{\boldsymbol{\theta}}^*(\boldsymbol{w})) + M\sqrt{\frac{2\ln(2|\mathcal{W}|/\delta)}{N}} + \epsilon,
\end{aligned}
$$

The first inequality because we require inequality (23) to hold uniformly for all $|\mathcal{W}|$ functions. The second inequality is because $\hat{\boldsymbol{w}}$ is the ϵ-approximated solution descrided in Eqn. (16). The third inequality is applying inequality (23). The forth inequality is because $|\mathcal{W}| > 1$. Taking infimum over $\boldsymbol{w}$ on the right hand side, we obtain the desired bound.

H. Related Works on Bilevel Optimization

Bilevel optimization (Sinha et al., 2017) has aroused much attention in recently due to its ability to handle hierarchical decision making processes. Previous works utilize bilevel optimization in multiple areas of research, such as hyper-paramter optimization (Lorraine et al., 2020; Maclaurin et al., 2015; Pedregosa, 2016; MacKay et al., 2019), meta learning (Finn et al., 2017; Nichol & Schulman, 2018), neural architecture search (Liu et al., 2018; Xu et al., 2019; Shi et al., 2020; Yao et al., 2021b; Gao et al., 2021; Yao et al., 2021a; Shi et al., 2021) and sample re-weighting (Ren et al., 2018; Shu et al., 2019), coreset selection (Zhou et al., 2022c; Borsos et al., 2020).

I. Future Directions

MAPLE stills needs to demonstrate its applicability to NLP tasks especially on today's large pretraining language models (Devlin et al., 2018; Radford et al., 2019; Liu et al., 2019; Diao et al., 2019; Brown et al., 2020), cross-modal tasks (Gu et al., 2018; Gao et al., 2022; Zhou et al., 2022a), domain adaptation tasks (Diao et al., 2021; Huang et al., 2022) and self-supervised learning tasks (He et al., 2020; Grill et al., 2020; Chen et al., 2021a; Liu et al., 2022). It is also interesting to explore how MAPLE interacts with other parallel domain generalization methods (Luo et al., 2018; Bai et al., 2021a;b) , how it interacts with other methods focusing on model sparsity (Shao et al., 2019; Zhou et al., 2022b; Shi et al., 2021) and how it performs on more challenging benchmarks (Ye et al., 2022).