Improving Generalization via Meta-Learning on Hard Samples

Nishant Jain Arun S. Suggala Pradeep Shenoy
Google Research India
{nishantjn,arunss,shenoypradeep}@google.com

Abstract

Learned reweighting (LRW) approaches to supervised learning use an optimization criterion to assign weights for training instances, in order to maximize performance on a representative validation dataset. We pose and formalize the problem of optimized selection of the validation set used in LRW training, to improve classifier generalization. In particular, we show that using hard-to-classify instances in the validation set has both a theoretical connection to, and strong empirical evidence of generalization. We provide an efficient algorithm for training this meta-optimized model, as well as a simple train-twice heuristic for careful comparative study. We demonstrate that LRW with easy validation data performs consistently worse than LRW with hard validation data, establishing the validity of our meta-optimization problem. Our proposed algorithm outperforms a wide range of baselines on a range of datasets and domain shift challenges (Imagenet-1K, CIFAR-100, Clothing-1M, CAMELYON, WILDS, etc.), with 1% gains using VIT-B on Imagenet. We also show that using naturally hard examples for validation (Imagenet-R / Imagenet-A) in LRW training for Imagenet improves performance on both clean and naturally hard test instances by 1-2%. Secondary analyses show that using hard validation data in an LRW framework improves margins on test data, hinting at the mechanism underlying our empirical gains. We believe this work opens up new research directions for the meta-optimization of meta-learning in a supervised learning context.

1. Introduction

Overparameterized models, common in supervised learning [27], carry the risk of overfitting to training data. Typically, model generalization is measured on a validation dataset separate from the training data, for purposes of hyperparameter selection. Increasingly, this validation dataset is itself used as part of the learning objective in nested formulations, e.g., for hyperparameter tuning via gradient descent on the validation loss [11]. In particular, learned reweighting (LRW) approaches *learn* importance weights associated with training instances [14, 16, 17, 31, 32, 40] or groups of training instances [25, 41] by optimizing a weighted training loss alongside an unweighted *meta-loss* on the validation data. This bilevel optimization aligns training loss with the validation data distribution via reweighting, a useful property in addressing covariate shift [34], and for group DRO [9, 41]. Even for in-domain test data, Ren et al. [31] show that using "clean" validation data with bilevel optimization can overcome significant amounts of label noise in training data.

Thus the *choice of validation set* in a LRW paradigm can greatly influence the quality and properties of the learned classifier. We therefore ask the question: can we *optimize* the choice of validation data in LRW so as to maximize generalization of the resulting classifier? We refer to this problem of validation set selection as *meta-optimization*, since it produces data that is the input of another optimization (the meta-learning approach underlying LRW). The closest related work to our proposal is [40] which constructs a validation dataset on the fly during training of an LRW classifier, with a criterion of choosing representative instances.

Our primary hypothesis is that we can improve classifier generalization by using a validation set consisting of hard-to-classify instances from the training distribution. We therefore pose and formalize the problem of **M**eta-**O**ptimized **LE**arned **RE**weighting (MOLERE) where the partitioning of data into train and validation splits, and the LRW classifier corresponding to that split, are jointly optimized. We design an efficient algorithm to tackle this meta-optimization, and make the following contributions:

- We formalize the problem statement of *validation set optimization* in learned reweighting (LRW) classifiers for improved generalization. We prove that asymptotically our optimization objective exactly achieves our stated goal of maximizing accuracy on the hardest samples.
- We simplify the nested optimization of our proposal into a tractable bi-level optimization with a min-max game between two auxiliary networks: a "splitter" that finds the hardest samples, and a "reweighter" that minimizes loss on those samples using LRW. We also provide a simple train-twice heuristic that can be used for careful analysis

of the choice of validation data in LRW.

- We show strict accuracy ordering of LRW models based on validation set: easy < random < hard, demonstrating the importance of optimizing LRW validation sets. We obtain reliable gains over ERM across datasets (e.g., 1% on Imagenet/VIT-B backbone), and in domain-generalization (e.g., 1.36% on iWildCam dataset [4, 18]) and noisy-label settings (e.g., 4.2% on Clothing1M dataset [37]). We outperform a range of baselines including reweighting, meta-learning among others. Analyses show improved margins on test set in MOLERE classifiers, suggesting an explanation of our gains.
- We extend our results to natural hard samples as validation (Imagenet with Imagenet-R / Imagenet-A), showing 1-3% gains on *both in-domain and out-of-domain* test sets. This shows the value of our ideas even in scenarios where we pay no additional cost for meta-optimization.

We hope that our work will be seen as an initial step in establishing the value of meta-optimization of meta-learning, with our findings providing a strong proof-of-concept for the general research direction.

2. Related Work

Importance weighting for robustness. There have been several works on learning robust representations via example re-weighting lately. Mostly, these work aim at avoiding noisy labels in the train set by analyzing the effect of decreasing loss on a given instance on a clean validation set. This line of work includes either learning a per-instance free parameter [31] or learning a simple MLP network [32] to predict importance of an instance based on its loss value. However, the requirement of a clean validation set limits their applicability to realistic scenarios. To deal with this, a meta-learning based re-weighting scheme (Fast Sample Reweighting [40]) was proposed based on generating some sort of pseudo-clean data as a proxy validation set. It also proposed certain approximations to make the training process more computation efficient. On a different note, a recent proposal called RHO-loss [24] proposed to select only *worthy* points for training which increased generalization property of the model, calculated as the difference of the training loss and a hold-out set loss.

A recent line of work [16, 17] proposed weighting based on context/relevance of an instance compared with the overall data distribution. These papers target slow temporal drift in longitudinal datasets, and the development of better uncertainty measures for selective classification, respectively. In particular, Jain et al. [17], suggest that using target domain data in the validation set can improve domain shift performance of classifiers.

Meta Learning. The sample re-weighting task using a validation set comes under the umbrella of meta-learning [15], which follows the *learning-to-learn* paradigm. It is conceptually similar to model agnostic meta learning (MAML) [10], which learns a single set of parameters that can easily be customized (few-shot) to multiple tasks. Learning these shared parameters involves a nested optimization similar to the one presented here, and significantly optimized for efficiency by recent work[28, 30, 42].

Probabilistic Margins. Recent work [22] showed that the probabilistic margin in multi-class problems can be used to improve how neural networks deal with adversarial examples. The margin is defined as the difference between the probability of the true label and the largest of the remaining label probabilities, indicating the difficulty of classifying that instance. Liu et al. [22] propose re-weighting training instances, in presence of an adversarial attack, inversely related to their probability margins.

Just Train Twice. A recent work [21] proposed an effective strategy to improve sensitivity of ERM models towards certain groups by training the ERM model in 2 stages. The first stage is a standard training procedure, whereas the second stage involves giving more importance to the incorrectly classified examples in the first stage by up-weighting their loss in the aggregate loss term for updating the parameters. This can be interpreted as similar to giving more importance to low margin examples (here essentially a hard separation between positive and negative).

3. Preliminaries: Learned ReWeighting

We work with learned reweighting (LRW) classifiers where training data is reweighted in order to optimize some specified metric on validation data. The basic LRW formulation [31] works with two datasets $S_{tr} = \{(x_i, y_i)\}_{i=1}^{N}$ and $S_{val} = \{(x_i, y_i)\}_{i=1}^{M}$ (training & validation, respectively). Given a desired loss function $\ell(y, \hat{y})$, LRW learns a classifier $f_\theta(\cdot)$, (with parameters θ) and an instance-wise weighting function $\phi(\cdot)$ that minimize following bi-level objective:

$$\begin{aligned} \theta^*(\phi) &= \arg\min_{\theta} \sum_{(x,y)\in S_{tr}} \phi(x)\ell(y, f_\theta(x)), \\ \texttt{s.t.} \quad \phi^* &= \arg\min_{\phi} \sum_{(x,y)\in S_{val}} \ell(y, f_{\theta^*(\phi)}(x)) \end{aligned} \tag{1}$$

Notice that the validation loss is unweighted, while the training objective is a weighted loss. Essentially, the above bilevel objective computes an *estimate* of classifier performance on the validation set, and optimizes it indirectly via reweighting of training data to influence the learned classifier. The weighted-loss-minimizing classifier $f_{\theta^*}(\cdot)$, and the weights $\phi^*(\cdot)$, are learned jointly, typically using alternating stochastic updates over minibatches to make the learning tractable [31]; this is in line with other meta-learning approaches such as MAML [10]. Following more recent work [16], we use an instance-dependent neural network $\phi(x)$ for learning the training instance weights, instead of free parameters [31].

Intuition: LRW has been used for overcoming training label noise [31, 32] and for handling covariate shift [16, 41]. In these cases, the validation set is assumed to be representative of test samples (i.e., clean labels, or covariate-shifted data, respectively). The intuition is that the learned reweighting of the training loss *aligns* it with the (possibly different) validation distribution in the process of indirectly minimizing the meta-loss (Eq. 1). In particular, Shu et al. [32] show analytically that "validation-like" instances are upweighted. Thus, the validation set in LRW can be thought of as a *target for generalization*, with the learned classifier being optimized for performance on that target set.

4. MOLERE: Optimizing LRW models

4.1. Hypothesis and formal objective

Our primary hypothesis is that we can improve generalization capabilities of supervised learning by combining two ingredients: a) a learned-reweighting classifier, as described in the previous section, and b) an *optimized validation set* that strongly encourages desired properties in the reweighting classifier. We refer to this idea as **M**eta-**O**ptimization of the **Le**arned **Re**weighting framework.

In particular, we propose learning an LRW classifier with hard samples as validation set, to improve accuracy and generalization of learned classifiers. Since LRW by definition maximizes classifier performance on a given validation set, we believe that this choice of validation set will maximize generalization. Thus, given a dataset for training a predictive model, we need to 1) select the "hard instances" from the dataset and separate it into a validation set, and 2) train a classifier on the remaining data using this validation set for LRW. Notice that hardness of instances is determined in terms of the learned model itself; thus, formalizing the above idea leads to a joint optimization problem of data partitioning (train, validation), and LRW training. We present the formal problem below.

Objective: Let $S = \{(x, y)\}_{i=1}^{N+M}$ be the available data, and let Θ be the splitting function that splits S into training and validation datasets; to be precise, we let $\Theta(S)$ be the validation set and $\Theta(S)^c$ be its complement. MOLERE aims to solve the following tri-level optimization problem

$$\begin{aligned}
&\Theta^* = \arg\max_{\Theta} \sum_{(x,y)\in\Theta(S)} \ell(y, f_{\theta^*(\phi^*(\Theta),\Theta)}(x)) \\
&\texttt{where } \phi^*(\Theta) = \arg\min_{\phi} \sum_{(x,y)\in\Theta(S)} \ell(y, f_{\theta^*(\phi,\Theta)}(x)) \\
&\texttt{s.t. } \theta^*(\phi, \Theta) = \arg\min_{\theta} \sum_{(x,y)\in\Theta(S)^c} \phi(x)\ell(y, f_\theta(x)).
\end{aligned} \tag{2}$$

In words: Find a data split, such that across all possible splits the LRW classifier learned on the split has maximal error on the chosen validation set. In practice, we impose an additional constraint on the validation set size: $|\Theta(S)|/(N + M) \leq \delta$, where δ is some predefined fractional constant.

4.2. MOLERE objective and generalization

We now study the asymptotic properties of our proposed meta-optimization as $N + M \to \infty$, as a means of gaining theoretical insights into its generalization capabilities in comparison to classical empirical risk minimization (ERM). This analysis assumes a weighting function $\phi(\cdot)$ dependent on both x and y. Interestingly, our experiments revealed similar performance between this formulation of ϕ and one relying solely on x. The following proposition shows that asymptotically MOLERE solves a *robust optimization* objective.

Theorem 1 (Asymptotics)**.** *Consider the tri-level optimization in Equation (2). Suppose the weighting function $\phi(\cdot)$, and splitting function $\Theta(\cdot)$ are dependent on both x and y. Let's suppose $N + M \to \infty$, and $\lim_{N,M\to\infty} \frac{M}{N+M} = \delta$. Moreover, suppose the domains of ϕ, θ, Θ are very large and contain the set of all measurable functions. Then the objective of* MOLERE *is equivalent to*

$$\max_{S:|S|=\delta(N+M)} \min_{\theta} \sum_{(x,y)\in S} \ell(y, f_\theta(x)). \tag{3}$$

Proof Sketch. The proof of the theorem relies on the observation that there exists a weighting function ϕ that can transform any probability distribution P to any another distribution Q (as long as the support of Q is a subset of P). Using this observation, one can show that the second, third level optimization problems in Equation (2) is equivalent to following problem: $\min_\theta \sum_{(x,y)\in\Theta(S)^c} \ell(y, f_\theta(x))$. □

A similar result holds when both $\Theta(\cdot)$, $\phi(\cdot)$ rely solely on x. Intuitively, the above theorem shows that in the limit of infinite samples, MOLERE identifies the hardest samples in the training data, and learns a classifier that *minimizes the error on those samples*. This is exactly the goal laid out in our hypothesis above.

Connections to DRO. Interestingly, the objective in Equation (3) is the dual of the following Distributionally Robust Optimization (DRO) objective [5]

$$\min_{\theta} \max_{S:|S|=\delta(N+M)} \sum_{(x,y)\in S} \ell(y, f_\theta(x)). \tag{4}$$

DRO is a well studied framework for learning robust models [8, 26]. However, to the best of our knowledge, dual DRO is not studied in the literature; we will explore this connection more deeply in future work.

4.3. Efficient algorithm for validation optimization

Designing a tractable algorithm for meta-optimization runs into two technical challenges: a) How can we learn to assign instances to train and validation sets? and b) How do we efficiently solve the tri-level objective proposed in Sec. 4.1? We address these two challenges in this section. First, we use a second auxiliary network for soft-assigning instances to train and validation datasets. Second, we collapse the outer two loops of the trilevel objective into a minimax formulation, and thereby reduce it to a bi-level optimization problem. We describe each of these in order below.

Soft data assignment: Inspired by a recent work [3], we use a partitioning network to predict probability for each instance to be included in the validation set. At any point in time, the "splitter" network outputs soft assignments $\mathbb{P}(z|x, y)$ with $z \in \{0, 1\}$ indicating whether the example (x, y) belongs to the *pseudotest* set ($z = 0$) or the *pseudotrain* set. The pseudotrain set is used to train a classifier using standard cross-entropy. The splitter is then updated to identify easy instances for the classifier and assign them to the pseudotrain partition in the next round; this is achieved by minimizing the cross-entropy between its soft-assignment and classifier accuracy:

$$\begin{aligned}\mathcal{L}_{split} &= CE(\mathbb{P}_{splitter}(z_i|x_i, y_i), \mathbb{I}_{y_i}(\hat{y})) \\ \texttt{where} \quad \hat{y} &= \arg\max \mathbb{P}_{predictor}(y|x_i)\end{aligned} \tag{5}$$

In order to maintain the label distribution and train-to-test ratio, two regularizers [3] are added to penalize shift from the prior distribution, and to push label margins in the training split and testing split close to the original label margin (see supplementary for details).

Meta-optimization with min-max objective: To simplify the tri-level optimization in Sec. 4.1, we propose a bi-level approximation where the outer loop combines the data splitting and instance reweighting objectives. Specifically, we propose a min-max game between the splitter (parametrized by Θ) and the meta-network (parametrized by ϕ), where the splitter has to maximize validation set error whereas instance weights are focused on minimizing it:

$$\begin{aligned}\Theta^*, \phi^* &= \arg\max_{\Theta}\min_{\phi} \sum_{(x,y)\in S_t^c} \left(\ell(y, f_{\theta^*(\phi,\Theta)}(x)) - \mathcal{L}_{split}\right) \\ where \quad \theta^*(\phi, \Theta) &= \arg\min_{\theta} \sum_{(x,y)\in S_t} \phi(x)\ell(y, f_\theta(x))\end{aligned} \tag{6}$$

where the set $S_t = \{(x, y) : (x, y) \in S, \mathbb{I}\{\Theta(x, y) > 0.5\}\}$ and $S_t^c = S \setminus S_t$, $\mathbb{I}$ denotes the indicator function. The overall algorithm updates the splitter and instance weighting network once every Q steps of classifier update, and also regularizes the splitter with $\Omega_{ratio} + \Omega_{ld}$ at set intervals (every R steps). The complete description of this method is provided in Algorithm 1 and analysis of the loss function at the outer level, derivation of update equations for all parameters (Θ, ϕ, θ) are provided in the supplementary. Both the instance weight network and Splitter are parameterized as neural networks. The meta-network for instance weights predict the weight for a training instance $(x_i,\ y_i)$ by using (x_i) as input, $w_i = g_\phi(x_i)$ and the Splitter predicts a probability of (x_i,y_i) being in the train set by taking both of them as inputs $z_i = g_\Theta(x_i, y_i)$. In experiments, we refer to this end-to-end optimization method as **LRWOpt**.

Algorithm 1 LRWOpt: The Overall One-Shot Algorithm.

Require: θ, Θ, ϕ, learning rates $(\beta_1, \beta_2, \beta_3)$, $\mathcal{S}$, N, M.
Ensure: Robustly trained classifier parameters θ.

Randomly initialize θ, Θ and ϕ;
initialize ge = 0; ▷ Difference b/w train and val error
for e=1 **to** MaxEpochs **do**
    $\mathcal{S}_{tr}, \mathcal{S}_{val}$ = GenerateSplit($\mathcal{D}, \Theta$)
    for $b = 1$ **to** M//m **do** ▷ m is the batch size
        $\{(x_i^v, y_i^v)\}_{i=1}^m$ = SampleMiniBatch($\mathcal{S}_{val}, m$);
        $\Theta \leftarrow \Theta - \beta_1 \nabla_\Theta \sum (\mathcal{L}_{split} - \ell(y_i^v, f_\theta(x_i^v)))$
        $\phi \leftarrow \phi - \beta_2 \nabla_\phi \sum (\ell(y_i^v, f_\theta(x_i^v)) - \mathcal{L}_{split})$
        for $j = 1$ **to** Q **do**
            $\{(x_i, y_i)\}_{i=1}^n$ SampleMiniBatch($\mathcal{D}_t, n$);
            $\theta \leftarrow \theta - \beta_3 \nabla_\theta \sum g_\phi(x_i)\ell(f_\theta(x_i), y_i)$;
        end for
    end for
    if $\sum \ell(y_i^v, f_\theta(x_i^v)) - \sum \ell(y_i, f_\theta(x_i))$ <ge **then**
        break;
    end if
    ge = $\sum \frac{1}{M}\ell(y_i^v, f_\theta(x_i^v)) - \frac{1}{N}\sum \ell(y_i, f_\theta(x_i))$
end for

4.4. A simple train-twice heuristic

We now describe a simple *train-twice* heuristic that can be used to establish the importance of validation set optimization in LRW. We first train an ERM classifier on the available training data, and use the *probabilistic margin* (PM) as a proxy for instance hardness:

$$PM(x, y, \theta) = \mathbf{p}_y(x, \theta) - \max_{j, j\neq y} \mathbf{p}_j(x, \theta) \tag{7}$$

Here θ denotes the ERM classifier's parameters. This criterion was used to manually reweight adversarial examples in recent work [22]. Although this proxy score is inexact[1], it nevertheless allows us to design interesting heuristic LRW variants based on its rank-ordering of training instances: (1) **LRW-Hard**, where we use the lowest margin instances as validation data, and the rest of the instances as training data, (2) **LRW-Easy**, in which the *highest margin* instances are

[1] For instance, it measures instance hardness under an ERM classifier, not the to-be-trained LRW model; further, an overtrained classifier may give incorrectly overconfident margins [21].

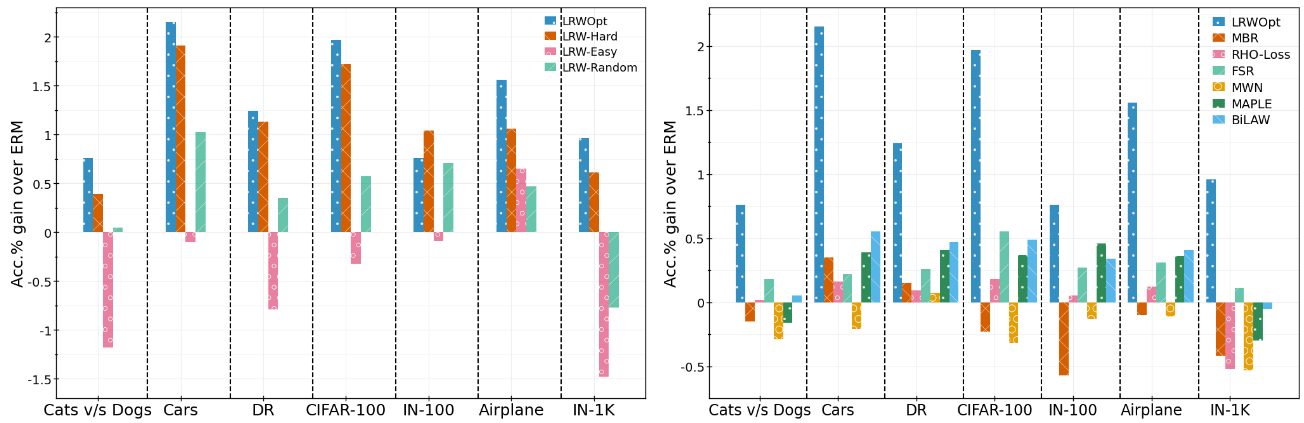

Figure 1. **Robustness analysis on benchmark datasets. Left:** Comparing different LRW variants, based on the choice of validation set (Easy, Random, Hard, corresponding to the rank-ordering of training data by probabilistic margin of an ERM classifier). y-axis shows accuracy gains over ERM for each dataset (x-axis). We see consistent ordering of performance, with LRW-Easy < LRW-Random < LRW-Hard, showing the importance of validation set optimization. **Right:** Comparing against other re-weighting methods. The figure shows that our proposal (LRW-Hard) outperforms the other reweighting techniques on average, with fast sample re-weighting (FSR) begin competitive in some datasets. In-1K corresponds to ImageNet-1K. For absolute accuracy values refer supplementary.

used as validation data, and (3) **LRW-Random**, a control which uses a randomly selected validation set that does not depend on the ERM margin.
We can use these variants to quantify the impact of perturbing the validation set in LRW, and to provide an *existence proof* of validation optimization techniques that materially improve learned classifiers. In particular, we expect to see a clear ordering of classifier test accuracy – LRW-Easy < LRW-Random < LRW-Hard.

5. Experiments

We perform extensive experimentation on multiple classification tasks including distribution shift benchmarks. For all datasets, if a train-validation split is already available, we use the training data for the ERM classifier in the train-twice heuristic before pooling, ranking, and repartitioning. For the end-to-end optimization, we start with pooled train-validation data and simultaneously learn the data splits and the corresponding LRWOpt model.

5.1. Datasets

We use popular classification benchmarks including CIFAR-100 [20], ImageNet-100 [35], ImageNet-1K [7], Aircraft [23], Stanford Cars [19], Oxford-IIIT Fine-grained classification (Cats v/s Dogs) [29] and Diabetic Retinopathy (DR) dataset. Furthermore, for OOD analysis we use the ImageNet-A [13], ImageNet-R [12] datasets for models trained on ImageNet-1K. We also use the Camelyon [2], iWildCam [4] dataset from the widely popular WILDS benchmark [18] and also the country shifted test set for the DR [1] dataset. We further analyze for robustness in presence of instance dependent noise on the noisy version of CIFAR-10 dataset (Inst.C-10) proposed by Xia et al. [36] and Clothing-1M [37] dataset. Please refer supplementary (supp.) for more details, setup for these datasets.

5.2. Baselines

Along with ERM, we also compare our method against various re-weighting based methods designed for improving robustness or handling noisy scenarios. These include learned re-weighting methods: MWN [32], FSR [40], L2R [31], MAPLE [41], BiLAW [14], GDW [6], StableNet [39] along with Margin Based Reweighting (MBR) [22] and Rho-Loss [24]. Please refer supp. for more details regarding them.

6. Results

6.1. In-Distribution Generalization

6.1.1 MOLERE **improves classification accuracy**

Figure 1 (left) shows the LRW advantage (gains over an ERM baseline) for a range of datasets. Shown are 4 alternatives–LRW-Easy, LRW-Random, and LRW-Hard, corresponding to easy, random, and hard validation sets per the train-twice heuristic (Sec. 4.4), and LRWOpt, which is the end-to-end optimization of LRW classifier and train-validation split (Sec. 4.3). Refer supplementary for absolute accuracy values.

- Gains vs ERM are strictly ordered in nearly all datasets–LRW-Easy < LRW-Random < LRW-Hard–robustly confirming our hypothesis that validation sets in LRW classi-

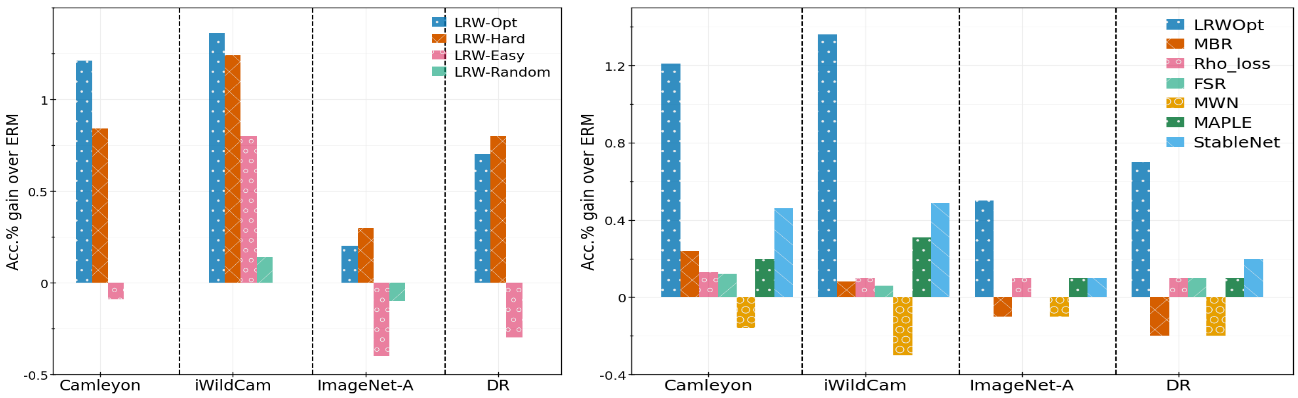

Figure 2. **OOD generalization. Left:** Comparison of LRW variants on domain shift benchmarks. The ordering between the validation selection methods is reconfirmed on domain shift benchmark datasets as well, suggesting that earlier gains are not via overfitting to training distribution. **Right:** Comparing against other re-weighting methods. The figure shows that our proposal (LRW-Hard) outperforms the other reweighting techniques on average, with fast sample re-weighting (FSR) begin competitive in some datasets. For absolute values refer supp.

fiers need to be optimized. The difference between LRW-Hard and LRW-Easy is over 2.5% relative gain for Imagenet & CIFAR-100, and 1.6% relative for Clothing-1M.

- LRW-Random shows modest gains over ERM in most datasets[2]. LRW-Hard shows significant gains, underscoring the benefit of combining LRW with validation set optimization – 0.8% relative gain on Imagenet, 1.3% on Clothing-1M, 2.15% relative on CIFAR-100.
- The end-to-end LRWOpt matches or exceeds LRW-Hard, showcasing its effectiveness without the need for training twice. On a minority of datasets, LRW-Hard is nominally better; we believe this is due to the small-sample nature of real-world datasets, and attendant estimation noise.

Although the presented accuracy is on unseen test sets, a concern may be that LRW-Hard overfits to the training data distribution and results in brittle classifiers. To address this, we performed a number of experiments on datasets with matched out-of-domain test sets (Diabetic retinopathy [1, 33], Camelyon [2], WildCam [4]) in next subsection.

6.1.2 MOLERE **outperforms existing re-weighting baselines**

We now compare our LRWOpt method against other reweighting methods– FSR [40]) , MBR [22], MAPLE [41], MWN [32], StableNet [39], BiLAW [14])–see Fig. 1 (right) for more details. While FSR, RHO-Loss, MBR, MAPLE, StableNet and BiLAW are proposals in the literature for learned reweighting of the training data (via meta-learning) in order to address various clean- and noisy-label scenarios, MBR is an extrapolation of a proposal for ad-hoc upweighting of *adversarial examples* alone [22], for increased robustness against those specific adversarial attacks. In similar spirit, the proposal "just train twice" [21] also suggests an ad-hoc upweighting of poor performing instances in a second round of training. RHO-loss [24], on the other hand, follows the harder version as against all these methods and selects only some points from a batch which minimize the holdout set loss for efficient training.

On the datasets tested, we see that LRWOpt clearly outperforms all these baseline methods consistently across all the datasets. It is followed by MAPLE, StableNet which are designed keeping group/OOD robustness in mind. FSR and Rho-Loss perform similar to ERM only. Interestingly, MBR and MWN are among the worst performers, showing that it is insufficient to simply upweight instances based on a first-round estimate of margin and the commonly followed bi-level optimization procedure by using free-parameters/loss based re-weighting, even though capable of providing unbiasedness, is not sufficient and requires another level of optimization for finding the appropriate validation set.

6.2. Out-Of-Distribution (OOD) generalization

We further evaluate the robustness of our learned representations on the standard out-of-distribution datasets from the Wilds benchmark namely the camelyon [2] and the iwildcam [4] datasets along with the Diabetic Retinopathy dataset with a country shifted test set (APTOS test dataset) [1]. For the first two, 10% examples from the train set, sampled randomly, are used as val set. For the last one, a separate validation set, with same domain as the train set, is provided. Refer supplementary for more details regarding

[2] Previous results [31] had shown neutral or slightly negative gains over ERM, focusing chiefly on training with noisy training labels + clean validation data. The improvement in our results are driven by the use of the meta-network.

the dataset details and our evaluation setup. Furthermore, we also analyze the performance on the ImageNet-A dataset by using both training and validation data from ImageNet-1K. Our primary goal here is to check whether the gains we saw above are primarily driven by in-domain learning, or a broader improvement in generalization capacity over the existing learned reweighting methods –as a result, we restrict ourselves to comparison among ERM, the LRW variants along with LRWOpt and the re-weighting baselines, rather than the substantial literature on domain shift. Fig. 2 (left) confirms that our learned classifiers generalize better to domain shift data as compared to ERM classifiers; further, the ordering between the different validation datasets is largely preserved. Fig. 2 (right) provides evaluation of our methods and other re-weighting baselines. These include the standard bi-level optimization based ones like MWN, FSR along with ones designed for OOD generalization/robustness like MAPLE, StableNet, Rho-Loss and MBR. It can be observed that our method surpasses all of these baselines thereby showing the importance of validation set optimization problem for the bi-level optimization based re-weighting methods, introduced in the paper. StableNet comes out to be the closest competitor of our method which has been designed explicitly for OOD learning. Rest all the re-weighting methods perform similar to ERM with minor gains/losses. Absolute accuracy values for this analysis are provided in the supplementary.

6.3. Practical Label Noise Settings

We now study the noisy label setting with a focus on real-world label noise using datasets like Clothing1M [37], and noisy CIFAR-10 (Inst. C-10 [36]), which contain instance conditioned noise as against randomly flipping labels or adding uniform noise. This noise can be treated as a measure of instance hardness rather than labeling error (refer supplementary for more details). Table 3 compares LRWOpt method against bi-level optimization approaches like MWN, FSR, L2R, GDW which although specifically designed for noisy labels have been primarily tested on label flipping, uniform noise data. We also compare against MAPLE, designed for general robustness. On Inst. C-10, our method clearly surpasses all baselines, with 1.53% accuracy gain. On Clothing-1M, we show about 0.85% gain over the best performing GDW baseline. For other noisy settings like random label flips, prior work on outlier robust DRO [38] has shown good results using loss clipping (e.g. exclude k% of highest loss as label noise), which could be co-opted into our work.

6.4. Skewed Labels

We now study the skewed label setting, comparing against instance based reweighting schemes proposed with this setting in mind (Meta-Weight-Net, FSR and another recently proposed re-weighting method GDW [6]). Table 2 shows the results for this analysis on the CIFAR-100 datasets with various skew levels ranging from 1 to 200. It can be observed that our LRWOpt significantly outperform the existing loss based reweighting methods at all of the skew levels showing accuracy gains upto 2.22%.

Method	Easy	Hard	Random	LRWopt	MWN	ERM
Acc.	83.17	**85.09**	83.87	84.94	84.02	84.11

Table 1. ImageNet-1k dataset with a ViT-B/16 pretrained backbone. We compare various versions of our method with the ERM, Meta-Weight and l2s baselines.

class skew	200	50	10	1
MWN [32]	40.11 ± 0.9	48.67 ± 0.7	61.32 ± 0.6	74.23 ± 0.3
FSR [40]	38.04 ± 0.8	45.12 ± 0.9	58.38 ± 0.6	74.68 ± 0.2
GDW [6]	40.36 ± 1.0	48.89 ± 0.8	61.67 ± 0.5	74.41 ± 0.4
LRWOpt	$\mathbf{42.33} \pm 0.8$	$\mathbf{50.77} \pm 0.7$	$\mathbf{63.28} \pm 0.8$	$\mathbf{75.12} \pm 0.3$

Table 2. Comparison, on CIFAR-100 dataset, of our LRWOpt method with existing meta-learning based reweighting methods in a label skew setup for which these methods were defined.

	MWN	FSR	L2R	MAPLE	GDW	Ours
Inst. C-10	65.89	67.12	70.21	70.34	69.12	**71.87**
Clothing-1M	72.79	72.07	72.22	71.67	73.12	**73.97**

Table 3. **Instance-dependent noise**. ImageNet-1k dataset with a ViT-B/16 pretrained backbone. We compare various versions of our method with the ERM, Meta-Weight and l2s baselines.

6.5. MOLERE scales to large pretrained models

We further analyze the LRW-Hard, Easy and Random methods along with the ERM baseline on ViT-B/16 pretrained backbone trained using these techniques and evaluated on the ImageNet-1K dataset. Table 1 shows the results from this experiments. Again LRW-Hard emerges the winner and improves significantly (around 1%) w.r.t. other methods and the baseline advocating robustness using hard examples. Furthermore, there is 1.92% difference in accuracy between LRW-Easy and LRW-Hard, even though we warm-started all the techniques with a pretrained backbone, showing the sensitivity of techniques to choice of validation set.

6.6. Leveraging OOD val set: A heuristic solution

We now turn to the use of known hard instance, or OOD, datasets when available, in the MOLERE context. For the well-studied Imagenet-1K dataset, two datasets are frequently used to gauge generalization properties of learned classifiers – IN-A (“natural adversarial instances”, examples known to be misclassified by a trained Resnet-50

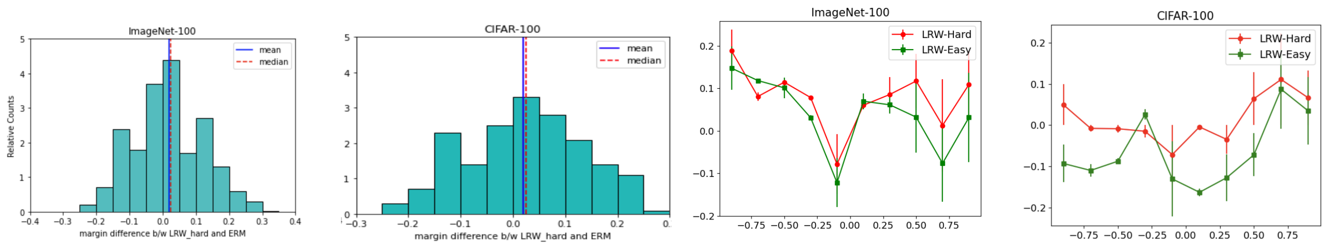

Figure 3. MOLERE improves margins of learned classifiers. (a,b): paired margin deltas between LRWOpt and ERM are moderately right-skewed with mean/median greater than zero. (c,d): As a function of ERM margin, clear separation seen between LRW-Hard (better) and LRW-Easy (worse) in terms of margin gain over ERM (errorbars are SEM). All results on unseen test data; Imagenet-100, CIFAR100 shown for brevity, with similar results for other datasets in appendix.

model), and IN-R (renditions of objects, such as drawings, paintings, sculptures, etc). For each of these, we used a portion of the dataset as LRW validation set, and retained another portion for testing. Table 4 shows the results of this study. We report the following two surprising results: (1) our approach outperforms the baseline ERM classifier on not just the OOD test set, but *also on the in-domain test set* (1.7-3% and 0.5-0.7% gains respectively), and (2) *simply adding the OOD data to the train set* only moderately improves OOD test set performance while *degrading* ID accuracy. These findings were replicated for both IN-A and IN-R as LRW validation sets, underscoring the real-world applicability of our findings, on large-scale datasets.

	IN-R val (ResNet50)		IN-A val (ResNet152)	
	IN1K Test	INR Test	IN1k Test	INA Test
LRWOpt + INR val	**76.14**	**49.1**	**78.12**	**7.9**
ERM (IN1K)	75.65	46.1	77.31	6.2
ERM (IN1K+INR)	74.89	47.4	77.08	6.6

Table 4. Natural hard examples as validation. LRW classifiers on Imagenet (IN1k) data and OOD validation sets (IN-A & IN-R respectively) outperform ERM on *both in-domain and OOD* test sets. Simply augmenting training data of ERM baseline does not match this (see text for details).

6.7. Margin maximization via meta-learning

We present empirical evidence showing that MOLERE has a *margin maximization effect*, i.e., the learned classifiers have wider margins than ERM classifiers. Since our validation data is selected to be low-margin instances, our LRW classifier upweights, and improves performance on, training instances most similar to the low-margin validation set. 3 shows two views of margin differences between MOLERE and ERM on the test set, to confirm this expectation[3]. Panels (a,b) shows a histogram of paired margin differences between ERM and LRWOpt, indicating a modest right-skew with mean & median to the right of 0. Panels (c,d) shows LRW-Hard and LRW-Easy deltas w.r.t. ERM, averaged over ERM margin buckets (mean and S.E.M. errorbars)– across the board (i.e., for most values of ERM margin), LRW-Hard contrasts with ERM better than LRW-Easy.

Further details on Algorithm 1, proof of Theorem 1, the time complexity of the proposed scheme compared to ERM, the training setup, ablation study over the validation set selection/loss, and discussion regarding the early-stage performance of the proposed scheme are provided in the supplementary.

7. Discussion & conclusion

We proposed the novel idea of *optimizing the choice of validation data* in a learned-reweighting setting, and showed that it gives significant gains over ERM on a range of datasets and domain generalization benchmarks. In particular, in most experiments on clean data, we saw a clear ordering between choosing easy, random, and hard samples as validation data in an LRW setup, with the latter performing best and delivering consistent gains over ERM. This ordering provides broad support to our primary hypothesis: meta-optimization of the metalearning workflow in LRW is an important area of research with potential for substantial impact. Our specific heuristic of choosing low-margin points is a simple, straightforward instantiation of what we believe is a family of optimization algorithms that can be brought to bear on the general problem of optimizing meta-learned classifiers. Indeed, our heuristic is not competitive under very high label noise scenarios, suggesting the need for follow-on work that explores more formal, optimization-driven approaches towards this problem. We are also excited about elucidating the theoretical basis of observed gains in the MOLERE framework.

[3]We show Imagenet-100 & CIFAR100 for brevity; findings consistent across all datasets (see Appendix).

References

[1] Kaggle. diabetic retinopathy detection challenge, 2015, 2015. 5, 6

[2] Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE transactions on medical imaging*, 38(2):550–560, 2018. 5, 6

[3] Yujia Bao and Regina Barzilay. Learning to split for automatic bias detection. *arXiv preprint arXiv:2204.13749*, 2022. 4

[4] Sara Beery, Elijah Cole, and Arvi Gjoka. The iwildcam 2020 competition dataset. *arXiv preprint arXiv:2004.10340*, 2020. 2, 5, 6

[5] Aharon Ben-Tal, Dick Den Hertog, Anja De Waegenaere, Bertrand Melenberg, and Gijs Rennen. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357, 2013. 3

[6] Can Chen, Shuhao Zheng, Xi Chen, Erqun Dong, Xue Steve Liu, Hao Liu, and Dejing Dou. Generalized dataweighting via class-level gradient manipulation. *Advances in Neural Information Processing Systems*, 34:14097–14109, 2021. 5, 7

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[8] John Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *arXiv preprint arXiv:1810.08750*, 2018. 3

[9] Matthew Faw, Rajat Sen, Karthikeyan Shanmugam, Constantine Caramanis, and Sanjay Shakkottai. Mix and match: an optimistic tree-search approach for learning models from mixture distributions. *Advances in Neural Information Processing Systems*, 33:11010–11021, 2020. 1

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017. 2

[11] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimilano Pontil. Bilevel programming for hyperparameter optimization and meta-learning, 2018. 1

[12] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021. 5

[13] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021. 5

[14] Chester Holtz, Tsui-Wei Weng, and Gal Mishne. Learning sample reweighting for accuracy and adversarial robustness. *arXiv preprint arXiv:2210.11513*, 2022. 1, 5, 6

[15] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020. 2

[16] Nishant Jain and Pradeep Shenoy. Instance-conditional timescales of decay for non-stationary learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12773–12781, 2024. 1, 2, 3

[17] Nishant Jain, Karthikeyan Shanmugam, and Pradeep Shenoy. Learning model uncertainty as variance-minimizing instance weights. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 2

[18] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021. 2, 5

[19] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 5

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[21] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021. 2, 4, 6

[22] Feng Liu, Bo Han, Tongliang Liu, Chen Gong, Gang Niu, Mingyuan Zhou, Masashi Sugiyama, et al. Probabilistic margins for instance reweighting in adversarial training. *Advances in Neural Information Processing Systems*, 34: 23258–23269, 2021. 2, 4, 5, 6

[23] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 5

[24] Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, et al. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, pages 15630–15649. PMLR, 2022. 2, 5, 6

[25] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pages 4615–4625. PMLR, 2019. 1

[26] Hongseok Namkoong and John C Duchi. Variance-based regularization with convex objectives. *Advances in neural information processing systems*, 30, 2017. 3

[27] Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*, 2018. 1

[28] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018. 2

[29] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 5

[30] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019. 2

[31] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343. PMLR, 2018. 1, 2, 3, 5, 6

[32] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32, 2019. 1, 2, 3, 5, 6, 7

[33] Asia Pacific Tele-Ophthalmology Society. Aptos 2019 blindness detection dataset, 2019. 6

[34] Masashi Sugiyama, Taiji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul Von Bünau, and Motoaki Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60:699–746, 2008. 1

[35] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European conference on computer vision*, pages 776–794. Springer, 2020. 5

[36] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems*, 33:7597–7610, 2020. 5, 7

[37] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699, 2015. 2, 5, 7

[38] Runtian Zhai, Chen Dan, Zico Kolter, and Pradeep Ravikumar. Doro: Distributional and outlier robust optimization. In *International Conference on Machine Learning*, pages 12345–12355. PMLR, 2021. 7

[39] Xingxuan Zhang, Peng Cui, Renzhe Xu, Linjun Zhou, Yue He, and Zheyan Shen. Deep stable learning for out-of-distribution generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5372–5382, 2021. 5, 6

[40] Zizhao Zhang and Tomas Pfister. Learning fast sample re-weighting without reward data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 725–734, 2021. 1, 2, 5, 6, 7

[41] Xiao Zhou, Yong Lin, Renjie Pi, Weizhong Zhang, Renzhe Xu, Peng Cui, and Tong Zhang. Model agnostic sample reweighting for out-of-distribution learning. In *International Conference on Machine Learning*, pages 27203–27221. PMLR, 2022. 1, 3, 5, 6

[42] Luisa M Zintgraf, Kyriacos Shiarlis, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Caml: Fast context adaptation via meta-learning. 2018. 2