Improving Generalization via Meta-Learning on Hard Samples Supplementary Material

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

1. Algorithmic Description

Algorithm 1 in the main paper covers the details of the one-shot LRWOpt scheme. It requires a set of initialized Splitter parameters ϕ, Meta-Network parameters Θ and classifier parameters Θ and finally outputs a set of optimal classifier parameters θ^*. Also, the splitting of the total dataset ($\mathcal{D}$) into train ($\mathcal{S}^{tr}$) and validation set $\mathcal{S}^{val}$ is done using the Splitter parameterized as a neural network f_Θ (Θ being the parameters) and $0 < f_\Theta(x,y) < 1$ for any instance (x,y). The examples with $f_\Theta(x,y) > 0.5$ are put into validation set. F_Θ denotes the application of this splitting function onto the overall dataset outputting train and validation subsets, *i.e.* $F_\Theta(\mathcal{D}) = \{x_i, y_i : (x_i, y_i) \in \mathcal{D}; f_\Theta(x_i, y_i) \geq 0.5\}, \{x_i, y_i : (x_i, y_i) \in \mathcal{D}; f_\Theta(x_i, y_i) < 0.5\}$. Here, instead of applying the nested loops for the bi-level setup at the epoch level, we have done it at the batch level. Both of them yield nearly similar results.

2. Proof of Theorem 1

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*

$$\max_{S':|S'|=\delta(N+M)} \min_{\theta} \sum_{(x,y)\in S'} \ell(y, f_\theta(x)). \tag{1}$$

intuitively, the model picks points close to boundary into validation set.

Proof. Let S' and $S \backslash S'$ be any partitioning of the dataset S. Let $Q^{\text{val}}, Q^{\text{tr}}$ be the probability distributions corresponding to S', and $S \setminus S'$. The proof proceeds by showing that the following two optimization problems are equivalent

$$\min_\theta \mathbb{E}_{(x,y)\sim Q^{\text{val}}}[\ell(y, f_\theta(x))]. \tag{2}$$

$$\min_\phi \mathbb{E}_{(x,y)\sim Q^{\text{val}}}[\ell(y, f_{\theta^*(\phi)}(x))]$$
$$\texttt{s.t.}\,\theta^*(\phi) = \arg\min_\theta \mathbb{E}_{(x,y)\sim Q^{\text{tr}}}[\phi(x,y)\ell(y, f_\theta(x))]. \tag{3}$$

Observe that the above two optimization problems are the inner optimization problems of objectives (2) (from main paper) and (1). Showing that these two are equivalent would then immediately imply that objective (2) (from main paper) is equivalent to objective (1).

First let's consider the case where $\text{supp}(Q^{\text{tr}}) = \text{supp}(Q^{\text{val}})$. By choosing $\phi(x,y) = \frac{Q^{\text{val}}(x,y)}{Q^{\text{tr}}(x,y)}$, the constraint in Equation (3) can be rewritten as

$$\theta^*(\phi) = \arg\min_\theta \mathbb{E}_{(x,y)\sim Q^{\text{val}}}[\ell(y, f_\theta(x))].$$

Observe that this is the same as the optimization problem in Equation (2). This shows that both the optimization problems in Equation (2), (3) are equivalent.

Next, consider the case where $\text{supp}(Q^{\text{tr}}) \neq \text{supp}(Q^{\text{val}})$. This is the easy case to handle. To see this, consider the extreme case where $\text{supp}(Q^{\text{tr}}) \cap \text{supp}(Q^{\text{val}}) = \{\}$. Since the domain of θ contains the set of all measurable functions, we can set $\phi(x,y) = 1$ and choose $\theta^*(\phi)$ to be the Bayes optimal classifier[1] on both $\text{supp}(Q^{\text{tr}}), \text{supp}(Q^{\text{val}})$. It is easy to verify that this is an optimizer of Equation (3). This shows that Equation (2), (3) are equivalent. A similar argument can be used to handle the more general case of $\text{supp}(Q^{\text{tr}}) \neq \text{supp}(Q^{\text{val}})$. Here, we choose a ϕ that performs probability matching on the intersection of the two supports, and set $\phi(x,y) = 1$ on the rest of the support. □

3. Deriving the Update Equations

Let us now discuss the update equation for each of the neural networks namely the Splitter Network (Θ), the Meta-Network (ϕ) and the target Prediction Network (θ). As discussed in the paper and in the algorithmic description provided above, we have formulated the problem as a bi-level

[1] A Bayes optimal classifier is a classifier that minimizes the expected population risk

optimization task with Θ, ϕ being optimized at the outer level and θ at the inner level. Therefore, for every update in Θ, ϕ, we update θ for K steps as an approximation for most optimal θ for the current value of Θ and ϕ, *i.e.* $\theta^*(\phi, \Theta)$.

Splitter Network (Θ). Updated at the outer loop level using the validation set to minimize the loss to identify whether a given input label pair would be predicted correctly and to maximize the loss on the validation set when using the current classifier parameters, for maximum generalization error. After e epochs of the complete bi-level setup, its update equation can be written as:

$$\Theta_{e+1} = \Theta_e - \frac{\beta_1}{M}\sum_{i=1}^{M}\nabla(CE(\mathbb{P}_{splitter}(z_i^v|x_i^v, y_i^v), \mathbb{I}_{y_i^v}(\hat{y}_i^v)) \\ - l_{val}(y_i^v, f_{\theta^*(\Theta,\phi)}(x_i^v)) \tag{4}$$

As discussed in the paper, along side this loss, the regularizers proposed in [3] are also used to update the splitter.

Meta-Network (ϕ). This is always updated alongside the splitter objective where the loss term corresponds to minimizing the error on validation set. Thus, after e epochs, it can be written as:

$$\phi_e = \phi_e - \beta_2\nabla_\phi\sum_{i=1}^{M}(l_{val}(y_i^v, f_{\theta^*(\Theta,\phi)}(x_i^v)) \\ -CE(\mathbb{P}_{splitter}(z_i^v|x_i^v, y_i^v), \mathbb{I}_{y_i^v}(\hat{y}_i^v))) \tag{5}$$

The overall setup is based on Meta-Network being independent of the Splitter and only aimed at making classifier generalize well on the validation set and thus, the second term inside the gradient can be asssumed independent of ϕ leading to:

$$\phi_e = \phi_e - \beta_2\sum_{i=1}^{M}\frac{\partial}{\partial\phi}l_{val}(y_i^v, f_{\theta^*(\Theta,\phi)}(x_i^v)) \tag{6}$$

Classifier Network (θ). Given the algorithm, after e epochs of the complete bi-level setup, it would have led to $Ke^{'}$ epochs over the training data, of the classifier where $e^{'}$ corresponds to number of epochs on train data while one epoch on val data is completed and K is the number of times inner loop is run for every outer loop. We have $e^{'} = \frac{batch\_t}{batch\_v}e$. The classifier has to be just updated through the weighted training loss on the split $\mathcal{D}^t$:

$$\theta_{Ke'+1} = \theta_{Ke'} - \beta_3\sum_{j=1}^{N}\nabla_\theta g_\phi(x_i)l(f_\theta(x_i), y_i) \tag{7}$$

Following the recent reweighting works[12, 21], we also approximate it as:

$$\theta_{Ke'+1} = \theta_{Ke'} - \beta_3\sum_{j=1}^{N}g_\phi(x_i)\nabla_\theta l(f_\theta(x_i), y_i) \tag{8}$$

The Classifier and Meta-Network update equations (eqs. 8 and 6) are same as the existing instance based reweighting works [11, 12] having a validation set bi-level set which they approximate as a single level optimization set. This involves creating a copy of the classifier ($\hat{\theta}$) and using that to update the meta-network (ϕ). Thus, ϕ For more details and derivations for the update equations of classifier and Meta-Network, following works [11, 12] can be referred.

Early Stage Performance and Convergence: Initially, the splitter network is likely to randomly assign data to training and validation, and the scorer network will assign random weights – in expectation, we believe this will fall back to the baseline ERM performance during initial training epochs. Empirically, examining learning curves of LRWOpt vs ERM, we see similarity in early epochs followed by gradual divergence. Also, previous work provides convergence guarantees for bi-level [21, 25] and min-max [6] objectives using alternating updates for learning.

4. Experimental Details

4.1. Training and Evaluation

Architectures. We have used following architectures for classifer: WRN28-10 for CIFAR-100, VGG-16 for ImageNet-100, ResNet-152 for Oxford-IIIT dataset, ResNet-32 for the aircraft and stanford cars datasets, and ResNet-50 for the rest. We add dropout regularization, following [12], to the classifier. We used a pretrained backbone as the base of the meta network having the same architecture as the classifier, to which we attach a fully connected layer for predicting the instance weights. For Splitter we follow the architecture from the learning-to-split[10] paper again the classifier backbone in a read-only manner followed by a learnable MLP layer to predict the splitting decision.

Training. For training the classifier, we use a batch size of 64 and image size of 224 × 224 for all experiments except CIFAR-100 where it is 32 × 32. We use an initial learning rate of 0.1 for the classifier, followed by a factor 10 decay every 50 epochs. For the meta network and splitter, we fix the learning rate at $1e-3$. We use a momentum value of 0.9 for all three. We run each experiment for 100 epochs of training, for which we observed convergence in all our experiments. We warm-start the main classifier by training for 25 epochs on a random split, followed by updating the meta-network and splitter for every 5 updates to the classifier (Q=5). We used a dropout rate of 0.25 for the classifier network, with 5 evaluations for estimating variance. The split of training/validation data, for different datasets, into the train and meta train sets is provided below along with dataset descriptions. We keep the length of training set same for all the methods and the baselines and do hyperparameter tuning on the validation set for the methods not using it in their optimization. We fix delta to be 0.1, resulting in a validation set of at most 10% of the training set; based on our splitting criterion, this target is always reached. In our current method, even if delta is set higher, only those examples will be included for which $\Theta(x, y) < 0.5$, thereby enforcing a hardness constraint. Also, both from experiments and DRO literature, we discovered increasing delta decreases variance but increases bias.

Regularizer details. We used the following two regularizers [3]:

$$\Omega_1 = D_{KL}(\mathbb{P}(z|B(\delta))), \Omega_2 = \sum_{k\in\{0,1\}} D_{KL}(\mathbb{P}(y|z = k))$$

where $\mathbb{P}(z)$ denotes the percentage of examples in train or val set and $B(\delta)$ is Bernoulli(δ). The first regularizer guides the splitter to maintain train/validation ratio close to δ and the second aims to balance labels across splits. We included them based on prior work; however, their contribution was modest – roughly 0.23% accuracy across datasets, with the second regularizer contributing most of it ($\sim$ 0.2%).

4.2. Datasets

As discussed in the paper, we have used popular classification benchmarks CIFAR-100, ImageNet-100, ImageNet-1K, Aircraft, Stanford Cars, Oxford-IIIT Fine-grained classification (Cats v/s Dogs) and Clothing-1M. Alongside these we have used ImageNet-A, ImageNet-R, Camelyon, iWildCam and Diabetic Retiopathy dataset with a country shift setup for OOD analysis using ImageNet-1K trained models.

ImageNet-1K [7] consisting of 1.3M images across 1000 classes, is the largest of the datasets in our experiments. We have used tha training set as the overall dataset (train+val) for applying our setup and training baselines.

ImageNet-100 [23]. A subset of ImageNet-1K with 100 classes, 130k training instances and 5k examples for testing as a validation set. We use 13k examples from the train set as our validation set, and the rest for training.

Clothing-1M [26]. Around 1M images from 14 apparel classes; since images and labels are programmatically extracted from the web, there is significant label noise. Around 72k manually refined examples form a clean subset, of which 10k examples comprise the test set and the remaining 11k are marked as validation.

CIFAR-100 [14]. This dataset consists of 60k images of size 32×32 spread across 100 categories. We use 50k images for training and 10k for testing. For the meta-network, we use 5k examples from the train set as validation data, and the remaining 45k examples for training the classifier.

Inst. C-10[24]. The setup here is same as the CIFAR-100 dataset with 30% noise. **Stanford Cars** [13]. This dataset contains around 16k images from 102 categories of cars based on Model, company, etc. with a roughly equal split between train and test set. We use 1K images (around 10 images per class), from the train set use as our validation set.

Aircraft [17]. It consists of 10.2k images belonging to different types of aircrafts covering 102 categories. The task involves fine-grained image classification into these 102 classes. The train, validation and test sets are equal splits of data. We combine the train and validation data for applying our scheme and limit the length of our validation set to be the length of original validation set.

Oxford-IIIT pet dataset [19]. It consists of 37 categories representing breeds of dogs and cats, with 200 images per category equally divided into train and test sets. We test our method on the fine-grained image classification task for this dataset. For the validation set, we use 600 examples from the train set.

ImageNet-A[9]. It comprises of real-world naturally existing (unmodified) examples which have been mostly misclassified by ResNet models. It has been proposed as a test set, comprising 7500 images belonging to 200 classes of the ImageNet-1K dataset.

ImageNet-R[8]. It comprises Images styled to various artistic renditions like paintings, drawings, etc. belonging to subset of classes of the ImageNet-1K dataset. It was basically designed to test generalization onto such renditions as the ImageNet dataset is restricted to photos. It consists of 30k images belonging to 200 classes.

Camelyon Dataset [2]. It consists from training data from various sources treated as different domains and a test dataset from completely different sources or domains. It involves the task of classification of breast cancer patients into various stages.

iWildCam Dataset [4]. It is again a classification task with the train dataset consisting of 441 different locations and a total of 217k images and the test dataset consist of a disjoint set of 111 different locations and a total of 63k images, spread throughout the globe. The aim is to identify the animal species from the given image.

Diabetic Retinopathy [1]. It involves the classification of a given retina scans into various levels of diabetic retinopathy (total 5 levels). For this work, we have binarized the task to 2 categories : (0,1) and(2,3,4). We train the model on the Kaggle dataset extracted from hospitals in the US. For OOD testing, we use the APTOS dataset [22], extracted from a different country's hospital, resulting in a significant domain shift presumably due to equipment and protocol differences. The test set consists of around 3k images and val set around 3k images.

4.3. Baselines

ERM. This is the standard empirical risk minimization classifier, obtained by training a classifier with cross-entropy

DATA SET	EASY	HARD	RANDOM	OPT	ERM	MBW	RHO-LOSS	FSR	MWN	MAPLE	BILAW
CIFAR-100	79.13	81.17	80.02	81.42	79.45	79.22	79.63	80.10	79.13	80.12	79.87
AIRCRAFT	80.87	81.28	80.69	81.78	80.22	80.12	80.34	80.55	80.11	80.58	80.37
STANFORD CARS	80.22	82.23	81.35	82.47	80.32	80.67	80.48	80.55	80.11	81.40	81.07
IMAGENET-100	86.82	87.95	87.62	87.67	86.91	86.34	86.96	87.18	86.78	87.67	87.25
IMAGENET-1K	74.17	76.26	74.88	76.61	75.65	75.23	75.13	75.76	75.12	75.35	75.60
OXFORD-IIIT	91.15	92.72	92.38	93.09	92.33	92.18	92.35	92.51	92.04	92.17	92.38
DR (IN-DIST)	89.86	91.78	91.00	91.89	90.65	90.80	90.74	90.91	90.72	91.06	91.12

Table 1. Comparison of accuracies of LRW-Hard/Easy/Random and the existing baseline-reweighting/data selecting methods along with the standard ERM classifier, on various datasets discussed in the paper.

DATA SET	HARD	EASY	RANDOM	OPT	MAPLE	STABLENET	RHO-LOSS	FSR	MWN	MBR	ERM
CAMELYON	71.06	70.13	70.22	71.43	70.35	70.31	71.12	70.34	70.06	70.46	70.22
IWILDCAM	72.56	72.12	71.46	72.68	71.59	71.52	71.13	71.45	71.02	71.40	71.32
IMAGENET-A	5.6	4.9	5.2	5.5	5.4	5.4	5.4	5.5	5.2	5.2	5.3
DR (OOD)	86.9	85.8	86.1	86.8	86.2	86.3	86.2	86.2	85.9	85.9	86.1

Table 2. Comparison of accuracies of LRW-Hard/Easy/Random and the existing baseline-reweighting/data selecting methods along with the standard ERM classifier, on various Out-of-Distribution benchmarks discussed in the paper.

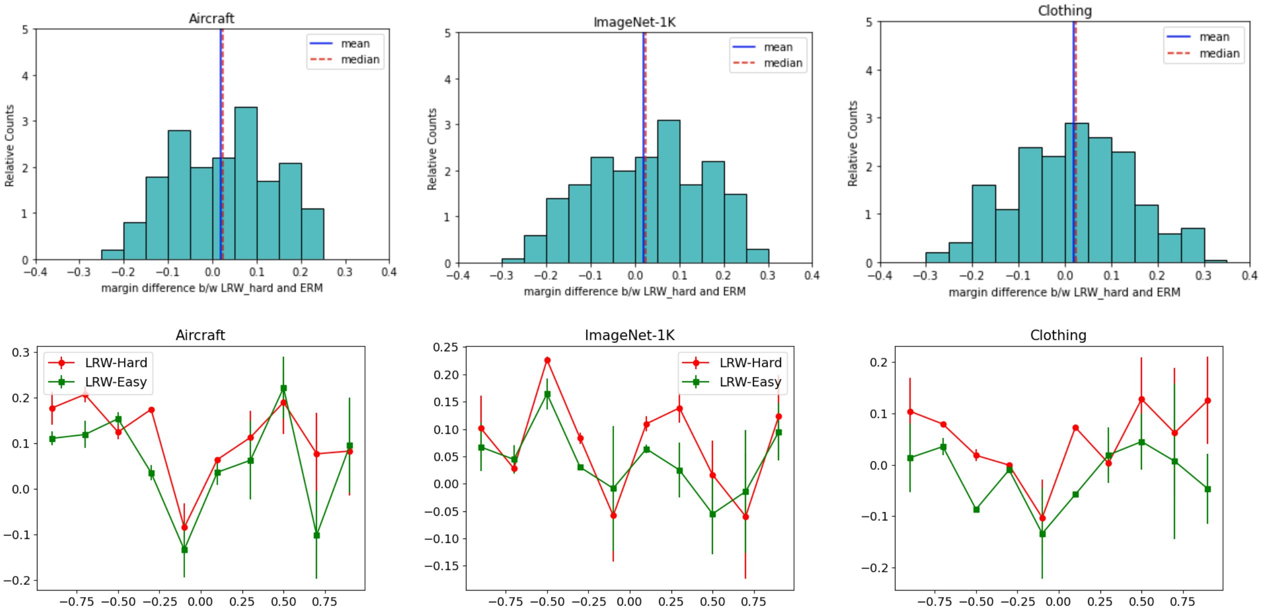

Figure 1. **Top.** Histograms of difference in margin of the LRW-Hard trained classifier and ERM classifier. **Bottom.** Mean and Standard deviation of margin deltas between the LRW Hard/Easy methods and the ERM classifier on the test examples, binned by ERM classifier margins with a bin width of 0.2 units. On the x-axis is the starting of the margin interval bin of the ERM classifier. We see that LRW-Hard classifiers have a tendency to *increase* margin over and above the ERM margin, whereas LRW-easy classifiers appear to reduce margin.

loss in a conventional batch-learning manner, and validation data used for hyperparameter tuning. Its margins are then used for selecting the validation set for our method.

Margin-based re-weighting. Liu et al. [16] suggest the use of probabilistic margin as an ad-hoc scaling term for adversarial samples, in an adversarial training regime. We extended this method to simply reweight all data points according to ERM margin, and training a second classifier from scratch. This is a control baseline to contrast the contributions of the LRW framework against the notion of using margin directly in the training loss.

Meta-Weight-Net. Shu et al. [21] proposed a learned re-weighting scheme based on bi-level optimization introduced in [20] but using a 2-layer meta network to predict instance level weights using taking loss as input.

Fast-Sample Re-weighting. Zhang and Pfister [28] proposed a meta-learning scheme which generates a pseudo validation set and then proposed an efficient and faster sample re-weighting based meta-learning technique to re-weight train examples.

RHO-Loss. Mindermann et al. [18] proposes selecting examples based on a Reducible Holdout Set Loss to maximize generalization–this results in an implicit reweighting of training instances. We use our validation set as its hold-out set for all the datasets.

MAPLE. Proposed in [29] Similar to [20, 28] based on free-parameters based reweighting setup but advocates for certain ”OOD risk” objectives for improving generalization/robustness of the classifier.

BiLAW. Another bi-level optimization based learned re-weighting similar to Meta-weight-Net, proposed by Holtz et al. [10], which feeds multi-class margin to the meta-network, resulting in a more robust (both in-dist and adversarially) classifier.

StableNet. Proposed by Zhang et al. [27] and optimizes sample weights such that the dependence among various features is decreased.

GDW. A recently proposed [5] re-weighting method designed for handling skewed or noisy label scenarios.

5. Time Complexity, Compute, Tuning:

Training time: Averaging over datasets in Figure 1 (from main paper), runtime as a function of ERM cost is (LRWOpt, LRW-hard, MWN, L2R) := (1.6x, 2.4x, 1.4x, 1.4x). LRWOpt is marginally more expensive than MWN for noticeably higher accuracy, and substantially lower than the train-twice heuristic (LRW-hard) while meeting or exceeding its accuracy.

FLOPS: (LRWOpt, LRW-hard) are $\sim$ (1.7x, 2.3x) ERM.

Hyperparams: Compared to LRW, we have one additional tunable hyperparameter for splitter's learning rate. LRW itself requires a meta-network learning rate and Q. Sensitivity analysis suggests any moderate value of Q is sufficient; we set $Q = 5$ across datasets. The parameter δ is fixed at 0.1; we believe this is a reasonable general tradeoff between training & validation sizes. We also found that ERM hyperparameters are sufficient for LRW; the meta-network and splitter learning rates can be tied to classifier learning rates without much degradation.

6. Comparison with *only* hard examples in validation set

We further analyze three new variants which involve using only hard examples in the validation set: First we incorporate the loss highlighted in [15] for the validation set along with our LRW-Hard method. Second, in our LRWOpt method we decrease the threshold Θ for train set to 0.2, such that only the hardest of the examples are there in the validation set and third where in the LRW-hard, we limit the validation set to only negative margin (i.e., incorrectly classified) examples from ERM. Table 3 shows accuracy % gains of LRWOpt over these variants on 4 randomly picked datasets including 1 OOD challenge.

	IN-100	DR	CIFAR-100	iWildCam
Variant 1	0.8	0.6	1.1	0.8
Variant 2	0.7	0.9	1.3	0.6
Variant 3	0.7	0.7	1.5	0.9

Table 3. Accuracy gain % of LRWOpt over variants.

7. Accuracy Comparison

We present the raw accuracy values of all methods reported in the paper. Table 1 shows the results for this comparison corresponding to Figure 1 and Table 2 corresponding to Figure 2 in the main paper. This underscores shows the effectiveness of our method as it shows gains even at high accuracy values like for the Oxford-IIIT pets dataset. Furthermore, it is also effective on relatively difficult datasets where model suffer in performance like Imagenet-1K dataset.

8. Analysis of Predicted Margins

We show the histograms of difference in margins predicted by LRW-Hard and ERM classifier on the remaining datasets including Aircraft, Stanford Cars and ImageNet-100. Figure 1 shows the results. Here also, a pattern similar to the main draft is observed, *i.e.*, more examples are on the positive side, thus showing that LRW-Hard is able to optimize margins. This is supported by both mean and median being on the positive side.

We also show experiment involving grouping the points based on ERM margin values and reporting the mean and std of the margin difference between LRW Hard/Easy and ERM classifier, for the remaining datasets including Aircraft, ImageNet-100 and ImageNet-1K, in Figure 1. The results are similar as in the main paper for the CIFAR-100 and Clothing datasets. Here also, for LRW-Hard, the mean is positive and significant, especially for positive margin examples showing the effectiveness of LRW Hard compared with ERM. Furthermore, the difference between LRW Hard and LRW Easy plots is significant, further backing the claim regarding importance of validation set and its effectiveness in margin maximization.

References

[1] Kaggle. diabetic retinopathy detection challenge, 2015, 2015. 3

[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. 3

[3] Yujia Bao and Regina Barzilay. Learning to split for automatic bias detection. *arXiv preprint arXiv:2204.13749*, 2022. 2, 3

[4] Sara Beery, Elijah Cole, and Arvi Gjoka. The iwildcam 2020 competition dataset. *arXiv preprint arXiv:2004.10340*, 2020. 3

[5] 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

[6] Ziyi Chen, Shaocong Ma, and Yi Zhou. Accelerated proximal alternating gradient-descent-ascent for nonconvex minimax machine learning. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pages 672–677. IEEE, 2022. 2

[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. 3

[8] 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. 3

[9] 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. 3

[10] Chester Holtz, Tsui-Wei Weng, and Gal Mishne. Learning sample reweighting for accuracy and adversarial robustness. *arXiv preprint arXiv:2210.11513*, 2022. 2, 5

[11] 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. 2

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

[13] 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. 3

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 3

[15] Xiaomeng Li, Lequan Yu, Yueming Jin, Chi-Wing Fu, Lei Xing, and Pheng-Ann Heng. Difficulty-aware meta-learning for rare disease diagnosis. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part I 23*, pages 357–366. Springer, 2020. 5

[16] 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. 4

[17] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 3

[18] 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. 5

[19] 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. 3

[20] 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. 5

[21] 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. 2, 5

[22] Asia Pacific Tele-Ophthalmology Society. Aptos 2019 blindness detection dataset, 2019. 3

[23] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European conference on computer vision*, pages 776–794. Springer, 2020. 3

[24] 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. 3

[25] Quan Xiao, Songtao Lu, and Tianyi Chen. A generalized alternating method for bilevel learning under the polyak-łojasiewicz condition. *arXiv e-prints*, pages arXiv–2306, 2023. 2

[26] 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. 3

[27] 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

[28] 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. 5

[29] 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. 5