Improving Robustness: When and How to Minimize or Maximize the Loss Variance

Valeriu Balaban
Department of Electrical and Computer Engineering
University of Southern California
Los Angeles, United States
vbalaban@usc.edu

Hoda Bidkhori
Department of Industrial Engineering
University of Pittsburgh
Pittsburgh, United States
bidkhori@pitt.edu

Paul Bogdan
Department of Electrical and Computer Engineering
University of Southern California
Los Angeles, United States
pbogdan@usc.edu

***Abstract*—We introduce distributional variance penalization, a strategy for learning with limited and/or mislabeled data. While minimizing the loss function currently stands as the training objective for many machine learning applications, it suffers from poor robustness. In this paper, we show that we can improve upon robustness issues by minimizing the average loss along with penalizing the variance. In particular, we expand on past studies of directly penalizing the variance which adjusts the weights of individual samples, resulting in improved robustness. However, the weights can take negative values and lead to unstable behavior. We introduce distributional variance penalization, which solves the issue of negative weights. Distributional variance penalization minimizes the expectation with respect to a distinct distribution that achieves a similar weighting scheme as direct variance penalization. We study the impact of both positive and negative variance penalization in the context of classification, and show that the generalization and the robustness against mislabeled data can be improved for a broad class of loss functions. Experimental results show that test accuracy improves by up to 20% compared to ERM when training with limited data or mislabeled data.**

***Index Terms*—Higher order statistics, image classification**

I. Introduction

Empirical risk minimization (ERM) is a key paradigm used in many machine learning applications. The objective of the ERM is to minimize a loss function averaged over the training data. A key drawback of ERM is its lack of robustness – poor generalization when training with limited data [1–3] and poor noise rejection when training with noise corrupted data [4, 5]. This paper aims to address this observed limitation by coupling the ERM strategy with a variance penalizing component.

We propose a unified framework that can improve robustness when the training data is either limited or mislabeled. We start by showing that penalizing the variance is equivalent to changing the weight of individual samples. Consequently, the variance penalization magnifies different data clusters, which drives the class boundaries to follow their alignment as shown in Fig. 1a. However, direct variance penalization leads to samples having negative weights which usually causes unstable behavior during training. We solve this issue by using probabilities instead of weights. Towards this end, we introduce distributional variance optimization that takes the expectation with respect to a new discrete distribution $\mathbb{Q}_n$ such that:

$$\mathbb{E}_{\mathbb{Q}_n}[\ell(\theta, x, y)] = \mathbb{E}_{\mathbb{P}_n}[\ell(\theta, x, y)] + \lambda \mathbb{V}_{\mathbb{P}_n}[\ell(\theta, x, y)] \quad (1)$$

where $\mathbb{P}_n$ is the empirical data distribution and $\ell(\theta, x, y)$ are the loss values. To improve generalization when training with limited data, we should proportionally increase the weights of samples with large loss values as those are not learned by the model. We accomplish this by using positive variance penalization, i.e., variance minimization. On the contrary, to improve noise rejection when training with mislabeled data, we should proportionally decrease the weights of samples with large loss values as those are usually the mislabeled samples. We accomplish this by using negative variance penalization.

A large and growing body of studies investigated minimizing the variance [1–4, 6–10], however, only a few for improving robustness when training with limited data. Duchi and Namkoong [2] proposed to penalize the variance by taking the expectation with respect to a different distribution. Li et al. [4] investigated the optimization of a tilted empirical risk which is equivalent to penalizing all central moments of the loss, including the variance. Moreover, we also explore the impact of increasing the variance to attenuate the effect of outliers when training with mislabeled data. Previous methods attenuated the impact of outliers by either using learning management [5, 11–15] or robust losses [5, 16–27].

The main contributions of our paper are as follows:

- (**C1**) **Novel robust framework.** We show that penalizing the variance improves robustness when training with limited data (Theorem 2) and mislabeled data (Theorem 4).
- (**C2**) **Positive variance penalization improves generalization.** We show that the variance minimization is equivalent to distributionally robust optimization (Equation 4).
- (**C3**) **Negative variance penalization improves robustness.** We prove that maximizing the variance bounds the loss function (Theorem 4) and improves robustness when training with mislabeled data.
- (**C4**) **Extensive experiments.** We show experimentally that both positive and negative variance penalization improve model accuracy (compared to ERM) by up to 20% when training with limited or mislabeled data, respectively.

In this paper, we limit the investigation to the classification problem using deep neural networks. However, the implications of the distributional variance penalization are much broader, and

can cover the bias-variance trade-off and regression problems.

We present the weighting scheme of direct variance penalization in Section II. In Section III we introduce distributional variance penalization which solves the issue of negative weights. In Sections IV and V, we analyze the robustness properties of positive and negative variance penalization. In Section VI, we provide a simplified procedure for finding the distribution $\mathbb{Q}_n$. Lastly, we present the experimental results in Section VII and the concluding remarks in Section VIII.

II. Direct Variance Penalization

We consider the training data to be distributed according to an unknown joint distribution $\mathbb{P}$, and consisting of n tuples of (x, y) where the attributes x are drawn from $\mathcal{X} \subseteq \mathbb{R}^d$ and the class labels y from $\mathcal{Y} = \{1, \dots, C\}$, where C represents the number of classes. Given a model $f : \Theta \times \mathcal{X} \to \mathbb{R}^C$ parameterizable by $\theta \in \Theta \subseteq \mathbb{R}^t$, our goal is to find the optimum parameters denoted by θ^* for which the model f correctly predicts the label y given x. To evaluate the prediction performance of the model, we use a loss function $\ell : \Theta \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$, with the objective to minimize $\mathbb{E}_{\mathbb{P}}[\ell(\theta, x, y)]$.

As the distribution $\mathbb{P}$ is unknown in the ERM setting, we minimize the empirical expectation $\mathbb{E}_{\mathbb{P}_n}[\ell(\theta, x, y)]$, or similarly:

$$\min_{\theta} \sum_{i=1}^{n} w_i \, \ell(\theta, x_i, y_i). \tag{2}$$

where w_i are the sample weights with $w_i = \frac{1}{n}, \forall i$. This fact simplifies the implementation, but hinders the robustness since not all samples are equal in practice [28].

We show in the next lemma that training with direct variance penalization is equivalent to changing the weights of individual samples. This result is independent of the loss function.

Lemma 1 (Variance Expansion)**.** *Let $\ell(\theta, x, y)$ be a loss function, λ a variance penalization factor, and w_i the samples' weights computed as $w_i = \frac{1}{n} + \frac{\lambda}{n}\big(\ell(\theta, x_i, y_i) - \mathbb{E}_{\mathbb{P}_n}[\ell(\theta, x, y)]\big)$, then the variance penalization is equivalent to the weighted sum with weights w_i:*

$$\mathbb{E}_{\mathbb{P}_n}[\ell(\theta, x, y)] + \lambda \mathbb{V}_{\mathbb{P}_n}[\ell(\theta, x, y)] \;=\; \sum_{i=1}^{n} w_i \, \ell(\theta, x_i, y_i).$$

Proof. Let Z be a random variable with $Z_i = \ell(\theta, x_i, y_i)$.

(i) $\sum_{i=1}^{n} w_i \, Z_i = \mathbb{E}_{\mathbb{P}_n}[Z] + \lambda \mathbb{E}_{\mathbb{P}_n}\big[(Z - \mathbb{E}_{\mathbb{P}_n}[Z])Z\big]$
By replacing w_i with its definition.

(ii)
$$\begin{aligned}
&\mathbb{E}_{\mathbb{P}_n}\big[\big(Z - \mathbb{E}_{\mathbb{P}_n}[Z]\big)Z\big] \\
&\quad= \mathbb{E}_{\mathbb{P}_n}\big[\big(Z - \mathbb{E}_{\mathbb{P}_n}[Z]\big)\big(Z - \mathbb{E}_{\mathbb{P}_n}[Z] + \mathbb{E}_{\mathbb{P}_n}[Z]\big)\big] \\
&\quad= \mathbb{E}_{\mathbb{P}_n}\big[\big(Z - \mathbb{E}_{\mathbb{P}_n}[Z]\big)\big(Z - \mathbb{E}_{\mathbb{P}_n}[Z]\big)\big] \\
&\qquad\qquad + \mathbb{E}_{\mathbb{P}_n}\big[\big(Z - \mathbb{E}_{\mathbb{P}_n}[Z]\big)\mathbb{E}_{\mathbb{P}_n}[Z]\big] \\
&\quad= \mathbb{V}_{\mathbb{P}_n}[Z] + \big(\mathbb{E}_{\mathbb{P}_n}[Z] - \mathbb{E}_{\mathbb{P}_n}[Z]\big)\mathbb{E}_{\mathbb{P}_n}[Z] \\
&\quad= \mathbb{V}_{\mathbb{P}_n}[Z].
\end{aligned}$$

Replacing (ii) in (i) completes the proof. □

Remarks. *From the formula for w_i, we can see that samples with a loss value $\ell(\theta, x_i, y_i)$ equal to $\mathbb{E}_{\mathbb{P}_n}[\ell(\theta, x, y)]$ receive a weight of $\frac{1}{n}$, the same as in the ERM setting. However, for all other samples, their weight depends on the deviation of their loss value from the empirical mean. The variance penalization factor λ defines the rate at which the weight changes based on this deviation. Moreover, for $\lambda > 0$, samples with loss values above average receive more weight, whereas for $\lambda < 0$, samples with loss values below average receive more weight.*

Although w_i sum to 1, we call them weights and not probabilities since w_i can take negative values. In particular, for $\lambda > 0$, a loss value $\ell(\theta, x_i, y_i) < \mathbb{E}_{\mathbb{P}_n}[\ell(\theta, x, y)] - \frac{1}{\lambda}$ will have a corresponding negative weight. Similarly, for $\lambda < 0$ a loss value $\ell(\theta, x_i, y_i) > \mathbb{E}_{\mathbb{P}_n}[\ell(\theta, x, y)] + \frac{1}{\lambda}$ will also have a corresponding negative weight.

The fact that the weights can take negative values usually leads to unstable behavior during optimization, as it can cause the objective to alternate between minimization and maximization. This is exacerbated for negative variance penalization, as in this case, samples with negative weights are the samples with large loss values and are more likely to skew the weighted sum; see the range highlighted in red in Fig. 1c. Though it also occurs for positive variance penalization, this issue is of less concern as the inset in Fig. 1b shows.

III. Distributional Variance Penalization

To solve the problem of having negative weights, we use a distributional variance penalization approach which achieves a similar weighting scheme as the direct variance penalization. We propose, instead of using the empirical distribution $\mathbb{P}_n$, to employ a new discrete distribution $\mathbb{Q}_n$ such that when taking the expectation $\mathbb{E}_{\mathbb{Q}_n}[\ell(\theta, x, y)]$ we recover $\mathbb{E}_{\mathbb{P}_n}[\ell(\theta, x, y)] + \lambda \mathbb{V}_{\mathbb{P}_n}[\ell(\theta, x, y)]$. As probabilities are nonnegative, we solve the problem of having negative weights. Henceforth, the optimization objective becomes:

$$\min_{\theta} \sum_{i=1}^{n} q_i \, \ell(\theta, x_i, y_i) \tag{3}$$

where the probabilities q_i are from a discrete distribution $\mathbb{Q}_n$.

To construct our surrogate distribution $\mathbb{Q}_n$, we measure how much it diverges from the empirical data distribution $\mathbb{P}_n$ by the chi-square divergence, D_{χ^2}. The chi-square divergence is calculated as $D_{\chi^2}(\mathbb{Q} \| \mathbb{P}) = \mathbb{E}_{x \sim \mathbb{P}}\left[\left(\frac{\mathbb{Q}(x)}{\mathbb{P}(x)} - 1\right)^2\right]$ where $\mathbb{P}(x)$ and $\mathbb{Q}(x)$ denote the two densities at x. The next theorem shows that we recover the variance penalization problem by selecting the distribution $\mathbb{Q}_n$ from an ambiguity set $\mathcal{A}$, which is constructed by bounding the chi-square divergence.

Theorem 2. *Let $\ell(\theta, x, y)$ be a loss function, $\lambda > 0$ a variance penalization factor, and $\gamma \in \mathbb{R}$ the variance exponent. We then have the following bounds for the two stochastic optimization problems:*

$$\max_{\mathbb{Q}_n \in \mathcal{A}} \mathbb{E}_{\mathbb{Q}_n}\left[\ell(\theta, x, y)\right] \le \mathbb{E}_{\mathbb{P}_n}[\ell(\theta, x, y)] + \lambda \mathbb{V}_{\mathbb{P}_n}[\ell(\theta, x, y)]^{\gamma}$$

$$\min_{\mathbb{Q}_n \in \mathcal{A}} \mathbb{E}_{\mathbb{Q}_n}\left[\ell(\theta, x, y)\right] \ge \mathbb{E}_{\mathbb{P}_n}[\ell(\theta, x, y)] - \lambda \mathbb{V}_{\mathbb{P}_n}[\ell(\theta, x, y)]^{\gamma}$$

where $\mathcal{A} = \left\{\mathbb{Q}_n : D_{\chi^2}(\mathbb{Q}_n \,\|\, \mathbb{P}_n) \le \frac{\lambda^2}{n^2} \mathbb{V}_{\mathbb{P}_n}[\ell(\theta, x, y)]^{2\gamma - 1}\right\}.$

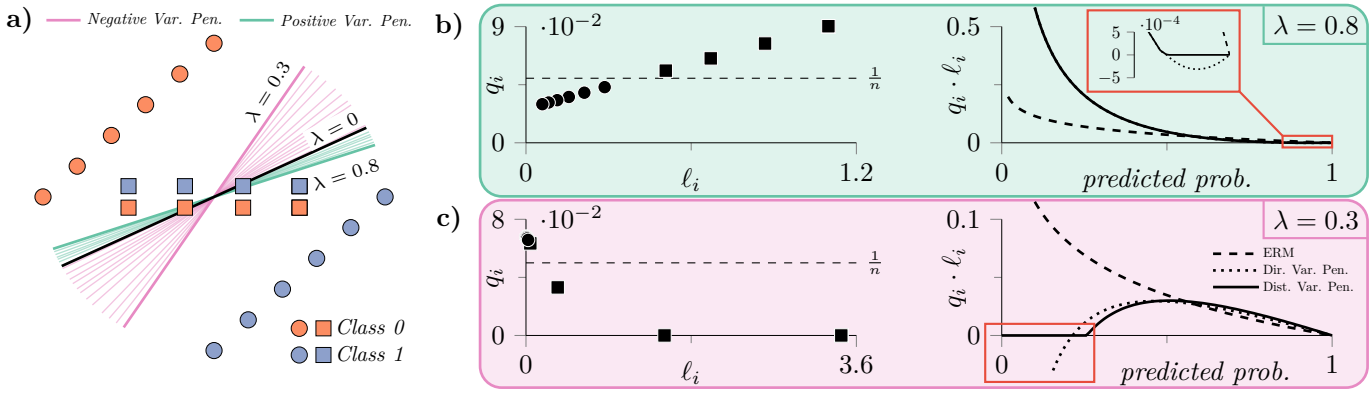

Figure 1: Toy example. (a) A binary classification problem with the decision boundaries of a linear classifier trained with $\gamma = 1$ and varying λ. (b) Sample probability for positive variance penalization with $\lambda = 0.8$. (c) Sample probability for negative penalization with $\lambda = 0.3$. The left-hand side subfigure in (b) and (c) shows the sample probability as a function of sample loss and compares it to the ERM case (dashed line). The right-hand side subfigure in (b) and (c) show the product of the crossentropy loss and sample probability as a function of predicted probability. The insets show where the two methods deviate.

Proof. Let Z be a random variable with $Z_i = \ell(\theta, x_i, y_i)$.

(i) $\mathbb{E}_{\mathbb{Q}_n}[Z] = \sum_{i=1}^{n} q_i Z_i$
s.t. $q_i \geq 0, \sum_{i=1}^{n} q_i = 1, \frac{1}{n}\sum_{i=1}^{n}(nq_i - 1)^2 \leq \frac{\lambda^2}{n^2}\mathbb{V}_{\mathbb{P}_n}[Z]^{2\gamma-1}$

(ii) Let $q_i = \frac{1}{n} + u_i$, then we have:
$\frac{1}{n}\sum_{i=1}^{n} Z_i + \sum_{i=1}^{n} u_i Z_i$
s.t. $u_i \geq -\frac{1}{n}, \ \sum_{i=1}^{n} u_i = 0, \ \frac{1}{n}\sum_{i=1}^{n} u_i^2 \leq \frac{\lambda^2}{n^2}\mathbb{V}_{\mathbb{P}_n}[Z]^{2\gamma-1}$

(iii) $\frac{1}{n}\sum_{i=1}^{n} Z_i + \sum_{i=1}^{n} u_i Z_i = \mathbb{E}_{\mathbb{P}_n}[Z] + n\mathbb{E}_{\mathbb{P}_n}[u(Z - \mathbb{E}_{\mathbb{P}_n}[Z])]$
By using the fact that $\mathbb{E}_{\mathbb{P}_n}[u] = 0$.

(iv) $n\,|\mathbb{E}_{\mathbb{P}_n}[u(Z - \mathbb{E}_{\mathbb{P}_n}[Z])]| \leq \lambda\mathbb{V}_{\mathbb{P}_n}[Z]^{\gamma}$
By Cauchy–Schwarz inequality.

(v) For $u_i = \pm\frac{\lambda}{n}\mathbb{V}_{\mathbb{P}_n}[Z]^{\gamma-1}(Z - \mathbb{E}_{\mathbb{P}_n}[Z])$
we have $n\,|\mathbb{E}_{\mathbb{P}_n}[u(Z - \mathbb{E}_{\mathbb{P}_n}[Z])]| = \lambda\mathbb{V}_{\mathbb{P}_n}[Z]^{\gamma}$.
By Cauchy–Schwarz, we attain equality if u and $(Z - \mathbb{E}_{\mathbb{P}_n}[Z])$ are linearly dependent, and given that the constraint $u_i \geq -\frac{1}{n}$ is satisfied.

(vi) Otherwise, $u_i = \pm\frac{\alpha\lambda}{n}\mathbb{V}_{\mathbb{P}_n}[Z]^{\gamma-1}(Z - \mathbb{E}_{\mathbb{P}_n}[Z] - \mu)$
and we have $n\,|\mathbb{E}_{\mathbb{P}_n}[u(Z - \mathbb{E}_{\mathbb{P}_n}[Z])]| < \lambda\mathbb{V}_{\mathbb{P}_n}[Z]^{\gamma}$
where $u_i \leq -\frac{1}{n}$ are set to $-\frac{1}{n}$, and $\mu > 0$ and $\alpha > 1$ are selected to satisfy the constraints from step (ii).

(vii) $\max_{\mathbb{Q}_n \in \mathcal{A}} \mathbb{E}_{\mathbb{Q}_n}[Z] \leq \mathbb{E}_{\mathbb{P}_n}[Z] + \lambda\mathbb{V}_{\mathbb{P}_n}[Z]^{\gamma}$
$\min_{\mathbb{Q}_n \in \mathcal{A}} \mathbb{E}_{\mathbb{Q}_n}[Z] \geq \mathbb{E}_{\mathbb{P}_n}[Z] - \lambda\mathbb{V}_{\mathbb{P}_n}[Z]^{\gamma}$
By combining the results of steps (iii) to (vi).

For an expanded version of this proof see (footnote 1). □

Remarks. *Of note, we generalize the variance penalization problem to any exponent γ of the variance term. In special cases, we have the classical penalization of variance for $\gamma = 1$ and the penalization of standard deviation for $\gamma = 1/2$. Moreover, the exponent γ controls the dynamic size of the ambiguity set $\mathcal{A}$. For $\gamma = 1/2$, the size of the set remains constant, whereas for all other values the size changes as the variance changes. In our experiments, we discover that using a $\gamma \neq 1$ usually yields better results. To find the distribution $\mathbb{Q}_n$, one must solve the optimization problem with constraints described in step (ii) of the proof. However, Algorithm 2 shows a simpler procedure for finding the distribution $\mathbb{Q}_n$.*

Both the direct and distributional variance penalization methods yield equivalent weights for small values of λ, i.e., when all the weights are positive. However, when some samples receive negative weights under direct variance penalization, these same samples will receive a probability of 0 under distributional variance penalization. This solves the issue of negative weights at the cost of a lower penalized variance. Moreover, samples with a corresponding probability of 0 will have no impact on the model parameters during the back-propagation phase.

Fig. 1 highlights how the distributional variance penalization helps to improve the generalization when training with limited or mislabeled data. Fig. 1a presents a toy dataset along with several linear decision boundaries obtained for different values of λ. In this dataset, each class contains two types of data points with distinct alignments. In particular, squared points are aligned horizontally and round points are aligned diagonally. In the ERM setting when $\lambda = 0$, i.e., all samples have equal weight, the decision boundary follows the alignment of the majority (in this case, the alignment of the round points as they are more numerous). However, by varying λ, we can bias the decision boundary to favor the alignment of either the round or the square data points, as shown by the colored lines. The different alignments result from using different distributions

$\mathbb{Q}_n$ which weigh each sample differently. The two colored boxes in Fig. 1b and Fig. 1c show the sample probabilities for $\lambda = 0.8$ (top) and $\lambda = -0.3$ (bottom), respectively.

Positive variance penalization increases the probability for samples with a higher than average loss value, as shown in the left plot of Fig. 1b. Here, the probability of the square data points increases as those samples are closer to the decision boundary and thus, have a higher loss value. Similarly, if a subset of data is uncaptured by the model, their loss values will be above average, and using a positive λ will increase their probability and help the model to learn.

In contrast, negative variance penalization increases the probability for samples with a lower than average loss value, as shown in the left plot of Fig. 1c. As a result, the decision boundary will follow the alignment of the round data points since those are the furthest from the boundary and thus have the smallest loss. Similarly, this helps reduce the impact of mislabeled data, as samples with large loss values are more likely to be mislabeled [29, 30].

The red rectangles in Fig. 1b and 1c highlight an important distinction between direct and distributional variance penalization. These insets reveal a bigger issue in the case of negative variance penalization since the weights are negative for large loss values and are more likely to skew the weighted sum. This causes unstable behavior as the objective will alternate between minimization and maximization.

IV. Positive Variance Penalization

In what follows, we link positive variance penalization with the distributionally robust optimization (DRO) [31, 32] where the goal is to minimize the expectation with respect to the worst-case data distribution. Moreover, we show that positive variance penalization promotes generalization by improving the confidence bound of the empirical mean.

Maurer and Pontil [1] introduced a concentration bound (Lemma 3) and proposed to extend the ERM objective by adding a variance penalization term. However, they noted that the variance penalization is non-convex in many situations where ERM is convex. To address this, Duchi and Namkoong [2] examined penalizing the standard deviation of the loss in the context of DRO which they showed to preserve the convexity of the loss function. We extend their previous work and analyze the DRO problem where the size of the ambiguity set is dynamic and changes with the loss variance.

Choosing the probabilities q_i in (3) according to the worst-case scenario from Theorem 2, naturally allows us to cast the objective in form of a DRO:

$$\min_{\theta\in\Theta} \max_{\mathbb{Q}_n\in\mathcal{A}} \mathbb{E}_{\mathbb{Q}_n}\left[\ell(\theta, x, y)\right] \tag{4}$$

where $\mathcal{A}$ represents a set of possible data distributions. The DRO goal is to protect against all deviations from a nominal model, which improves robustness against distributional shift [33]. Of note, the most common DRO formulations consider the size of the ambiguity set $\mathcal{A}$ to be constant. However, for distributional variance penalization, this is only the case for $\gamma = \frac{1}{2}$. In our experiments, we find that an expanding set, $\gamma < \frac{1}{2}$, yields better results. In this case, as the loss variance decreases, the selected distribution becomes progressively more unfavorable allowing for better optimization.

In addition, we have that by minimizing the variance through positive variance penalization we improve the following confidence bound.

Lemma 3 (Maurer and Pontil [1, Theorem 4])**.** *Let $Z_1, \ldots, Z_n$ be i.i.d random variables with values in $[0, M]$ and let $\delta > 0$. Then we have with probability at least $1 - \delta$ that:*

$$\mathbb{E}_{\mathbb{P}}[Z] - \mathbb{E}_{\mathbb{P}_n}[Z] \leq \sqrt{\frac{2M^2\mathbb{V}_{\mathbb{P}_n}[Z]\ln 2/\delta}{n}} + \frac{7M\ln 2/\delta}{3(n-1)}$$

where $\mathbb{P}$ represents the unknown data distribution and $\mathbb{P}_n$ the empirical one.

From this lemma, we see that the empirical variance $\mathbb{V}_{\mathbb{P}_n}[Z]$ bounds the deviation of the empirical mean from its theoretical value. Thus, by using a positive variance penalization, we direct the optimization algorithm to minimize both the mean and the variance together, which leads to a tighter concentration bound. Moreover, if during this, the maximum loss value, M, also decreases, then we attain an even tighter bound.

V. Negative Variance Penalization

In this section, we investigate negative variance penalization which translates to minimizing the expectation taken with respect to the best-case scenario distribution. Of note, this is the opposite of the DRO objective. Moreover, in this scenario, we show that the weight of inliers increases while the weight of outliers decreases. As a result, this constitutes a viable strategy when training with mislabeled data.

From prior research, we know that overparameterized models such as deep neural networks are capable of learning random datasets [29]. However, if a dataset contains real-world data along with random data, the model prioritizes learning real-world data before it memorizes the random data [30]. Henceforth, when training with mislabeled data, samples with a larger loss value are more likely to be the ones mislabeled, and thus, should not be relied upon. From Lemma 1, we know that we can decrease the weight of those samples using negative variance penalization. However, the drawback of this strategy is that samples with large loss values, i.e., $\ell(\theta, x_i, y_i) > \mathbb{E}_{\mathbb{P}_n}[\ell(\theta, x, y)] + \frac{1}{\lambda}$, will have negative weights which can lead to unstable behavior during training. In our experiments when using direct variance penalization, the optimization algorithm only converges when we use small values for λ and do not improve the results relative to the ERM case anyway. However, this is not the case for distributional variance penalization.

Choosing the probabilities q_i in (3) according to the best-case scenario of Theorem 2 recovers the following objective:

$$\min_{\theta\in\Theta} \min_{\mathbb{Q}_n\in\mathcal{A}} \mathbb{E}_{\mathbb{Q}_n}\left[\ell(\theta, x, y)\right] \tag{5}$$

This approach solves the convergence issue of the direct variance penalization as samples with corresponding negative

weights now receive a probability of 0. Samples that receive a probability of 0 will have no effect when computing the gradients during the back-propagation phase. In our experiments, similar to the positive variance penalization, we find that an expanding ambiguity set $\mathcal{A}$ yields better results. In this case, one must use a $\gamma > \frac{1}{2}$ for that.

Next, we show that negative variance penalization using the distributional approach bounds the loss function, and as a result, improves robustness when training with mislabeled data. We assume that we do not have access to clean data and that for each class, the correctly labeled samples represent the majority of the class. Formally, this assumption requires that $\mathbb{P}(\hat{y} = y) > \mathbb{P}(\hat{y} = j), \forall j \in \mathcal{Y} \backslash \{y\}$ where y denotes the true class labels, and $\hat{y}$ the mislabeled data. Moreover, here robustness implies that a model trained on mislabeled data has the same probability of misclassification as a model trained on clean data. Under these conditions, Ghosh et al. [17] outlined the distribution independent sufficient conditions for a loss function to be robust. Specifically, if the loss function satisfies the equality:

$$\sum_{j=1}^{C} \ell(\theta, x_i, j) = K \tag{6}$$

where C is the number of classes, K is a constant, and the equality holds $\forall \theta \in \Theta, \forall x_i \in \mathcal{X}$, then ℓ is a robust loss when all classes are equally mislabeled. However, if the ratio of the mislabeled data is different for each class, then for ℓ to be a robust loss, we must also have $\mathbb{E}_{\mathbb{P}_n}[\ell(\theta^*, x, y)] = 0$. The crossentropy (CE) loss, which is widely used for classification, is an unbounded loss and does not satisfy the above equality and is consequently extremely susceptible to mislabeled data. On the other hand, the mean absolute error (MAE) satisfies all the above conditions for a robust loss function but struggles to converge when training deep neural networks.

Recent studies [18–22] explored relaxing constraint (6) by requiring the sum to be bounded. Moreover, they showed that by tightening or relaxing the bounds, we can balance robustness with convergence. In a like manner, we next show that using negative variance penalization bounds the sum in (6).

Theorem 4. *Let $\ell(\theta, x, y)$ be an unbounded and non-negative loss function, and q_i the solution to $\min_{\mathbb{Q}_n \in \mathcal{A}} \mathbb{E}_{\mathbb{Q}_n}[\ell(\theta, x, y)]$ with $\mathcal{A} = \left\{\mathbb{Q}_n : D_{\chi^2}(\mathbb{Q}_n \,\|\, \mathbb{P}_n) \leq \frac{\lambda^2}{n^2} \mathbb{V}_{\mathbb{P}_n}[\ell(\theta, x, y)]^{2\gamma-1}\right\}$, then:*

$$0 \leq \sum_{j=1}^{C} q_i \ell(\theta, x_i, j) \leq \frac{C[1+\lambda\alpha\mathbb{V}_{\mathbb{P}_n}[\ell(\theta,x,y)]^{\gamma-1}(\mathbb{E}_{\mathbb{P}_n}[\ell(\theta,x,y)]+\mu)]^2}{4n\lambda\alpha\mathbb{V}_{\mathbb{P}_n}[\ell(\theta,x,y)]^{\gamma-1}}$$

where $\alpha \geq 1$ and $\mu \geq 0$ are two constants that emerge when finding the distribution $\mathbb{Q}_n$.

Proof. Let Z be a random variable with $Z_i = \ell(\theta, x_i, y_i)$ then from Theorem 2 we have the following expression for $q_i = \frac{1}{n}\left[1 - \lambda\alpha\mathbb{V}_{\mathbb{P}_n}[Z]^{\gamma-1}\big(Z_i - \mathbb{E}_{\mathbb{P}_n}[Z] - \mu\big)\right]_+$ where $\alpha \geq 1$ and $\mu \geq 0$ are two constants found when solving the stochastic optimization problem $\min_{\mathbb{Q}_n \in \mathcal{A}} \mathbb{E}_{\mathbb{Q}_n}[\ell(\theta, x, y)]$. The operator $[\cdot]_+$ truncates negative values to 0 and enforces the non-negativity constraint for q_i.

(i) $0 \leq q_i Z_i$
Since q_i and Z_i are non-negative.

(ii) $\frac{\partial}{\partial Z_i}(q_i Z_i) = \frac{1}{n}[1 - \lambda\alpha\mathbb{V}_{\mathbb{P}_n}[Z]^{\gamma-1}(2Z_i - \mathbb{E}_{\mathbb{P}_n}[Z] - \mu)]$
To find the maximum of $q_i Z_i$, the probability q_i must be positive and thus we drop the non-negativity constraint.

(iii) $\frac{\partial}{\partial Z_i}(q_i Z_i) = 0$ for $Z_i = \frac{1}{2\lambda\alpha\mathbb{V}_{\mathbb{P}_n}[Z]^{\gamma-1}} + \frac{\mathbb{E}_{\mathbb{P}_n}[Z] + \mu}{2}$

(iv) $q_i Z_i \leq \frac{[1 + \lambda\alpha\mathbb{V}_{\mathbb{P}_n}[Z]^{\gamma-1}(\mathbb{E}_{\mathbb{P}_n}[Z] + \mu)]^2}{4n\lambda\alpha\mathbb{V}_{\mathbb{P}_n}[Z]^{\gamma-1}}$
Since $\frac{\partial^2}{\partial Z_i^2}(q_i Z_i) = -2\lambda\alpha\mathbb{V}_{\mathbb{P}_n}[Z]^{\gamma-1}$ is negative, the value of Z_i from step (iii) maximizes the product. Thus, using this value in $q_i Z_i$ yields the above upper bound.

Combining (i) and (iv) completes the proof. □

Remarks. *The constants α and μ represent the Lagrange multipliers that satisfy the constraints of step (ii) in the proof of Theorem 2. The distribution $\mathbb{Q}_n$ can be found by solving the constraint optimization problem [34]. However, we present a much simpler approach in Section VI.*

From this theorem, we see that the upper bound depends on the statistics of the loss values. However, when penalizing the variance with $\gamma = 1$, only the empirical mean affects the upper bound. Moreover, in this case we achieve the tightest upper bound of $(\mathbb{E}_{\mathbb{P}_n}[\ell(\theta, x, y)] + \mu)/n$ for $\lambda = \frac{1}{\alpha(\mathbb{E}_{\mathbb{P}_n}[\ell(\theta,x,y)]+\mu)}$. However, using the tightest upper bound might not yield the best performance as it might trade convergence for robustness. As a result, we suggest performing hyper-parameter tuning to find the values for λ and γ which offer the best balance of robustness and convergence.

VI. Computational Aspects

Applying the method in practice requires including an additional step for finding the discrete distribution $\mathbb{Q}_n$. Algorithm 1 illustrates the training procedure. We provide a Pytorch implementation available on GitHub.[1]

Algorithm 1: Training with Variance Penalization

input :
- $\{x_i, y_i\}_1^n$ – training data
- $\ell(\theta, x, y)$ – loss function
- $\lambda,\ \gamma$ – hyper-parameters
- η – learning rate

while *stopping criteria not reached* **do**
- **for** $i \leftarrow 1$ **to** n **do**
  - $z_i \leftarrow \ell(\theta, x_i, y_i)$
- $q \leftarrow \textsc{Find}\mathbb{Q}_n(z, \lambda, \gamma)$
- $\mathcal{L} \leftarrow \sum_{i=1}^n q_i z_i$ // $\mathbb{E}_{\mathbb{Q}_n}[\ell(\theta, x, y)]$
- $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$

Algorithm 2 shows the procedure $\textsc{Find}\mathbb{Q}_n$ that solves the stochastic optimization problem of Theorem 2. The algorithm follows the approach of Duchi and Namkoong [2] of projecting

[1] https://github.com/valeriu-balaban/improving-robustness-with-variance-penalization

the loss values onto the probability simplex to find $\mathbb{Q}_n$. However, we extend the algorithm to encompass finding both the distribution that minimizes, and the distribution that maximizes the expectation. The advantage of this approach is that to recover the distribution that satisfies the constraints of Theorem 2, we only need to find the value for a single variable ρ, compared to two, α and μ. Alternatively, one can set ρ as a training parameter and update its value during back-propagation, which is not explored in this paper.

Algorithm 2: FIND$\mathbb{Q}_n$ Procedure

Input : z – loss values
λ, γ – hyper-parameters
ϵ – tolerance
Param : a, b – initial search range
s – $+1$ for $\max \mathbb{E}_{\mathbb{Q}_n}[z]$, -1 for $\min \mathbb{E}_{\mathbb{Q}_n}[z]$
Result : q – sample probabilities

$D_{\chi^2} \leftarrow \frac{\lambda^2}{n^2}\mathbb{V}_{\mathbb{P}_n}[z]^{2\gamma-1}$ `// target`
repeat
  $\rho \leftarrow \frac{1}{2}(a+b)$
  $r \leftarrow [s(z-\rho)]_+$
  $q \leftarrow r/\sum_{i=1}^n r_i$
  $\widehat{D}_{\chi^2} \leftarrow \sum_{i=1}^n \frac{1}{n}(nq_i - 1)^2$ `// current`
  if $s(\widehat{D}_{\chi^2} - D_{\chi^2}) < 0$ **then** $a \leftarrow \rho$ **else** $b \leftarrow \rho$
until $|\widehat{D}_{\chi^2} - D_{\chi^2}| < \epsilon$

Next, we highlight the influence of variance penalization on the update strategy of the parameters θ. Consider the case of negative variance penalization and let $R_{\mathbb{P}_n} = \mathbb{E}_{\mathbb{P}_n}[\ell(\theta, x, y)] - \lambda\mathbb{V}_{\mathbb{P}_n}[\ell(\theta, x, y)]^\gamma$, then by differentiating $R_{\mathbb{P}_n}$ with respect to θ, we have $\nabla_\theta R_{\mathbb{P}_n} = \sum_{i=1}^n w_i \nabla_\theta \ell(\theta, x_i, y_i)$ where $w_i = \frac{1}{n}\big[1 - 2\lambda\gamma\mathbb{V}_{\mathbb{P}_n}[\ell(\theta, x, y)]^{\gamma-1}\big(\ell(\theta, x_i, y_i) - \mathbb{E}_{\mathbb{P}_n}[\ell(\theta, x, y)]\big)\big]$. On the other hand, when penalizing the variance using the results of Theorem 2 by taking the expectation with respect to distribution $\mathbb{Q}_n$, then by differentiating with respect to θ we have $\nabla_\theta \mathbb{E}_{\mathbb{Q}_n}[\ell(\theta, x, y)] = \sum_{i=1}^n q_i \nabla_\theta \ell(\theta, x_i, y_i)$ where $q_i = \frac{1}{n}\big[1 - \lambda\alpha\mathbb{V}_{\mathbb{P}_n}[\ell(\theta, x, y)]^{\gamma-1}\big(\ell(\theta, x_i, y_i) - \mathbb{E}_{\mathbb{P}_n}[\ell(\theta, x, y)] - \mu\big)\big]_+$. Of note, the formulas for w_i and q_i are very much alike, they both change the impact of the gradient in a linear fashion based on the distance of the sample loss value to the mean. However, in the case of q_i, we can avoid unstable behavior during optimization since q_i are nonnegative. *We emphasize that probabilities q_i must be treated as constants.* Otherwise, it can lead to unstable behavior since when differentiating, we will have an additional term, $\ell(\theta, x_i, y_i)\nabla_\theta q_i$, which is negative in the case of negative variance penalization. To treat q_i as constants use `detach()` function in Pytorch, and `stop_gradient()` in Tensorflow.

VII. Experimental Results

We assess the performance of the proposed robustness framework on two datasets: *(i)* Fashion–MNIST which contains grayscale images associated with a label from 10 classes of apparel, and *(ii)* CIFAR–10 which contains color images associated with a label from 10 classes such as cat, truck, and airplane. We use the classical crossentropy loss function in the ERM and the variance penalization scenarios. The training procedure with the model configuration are described in Section VII-C and the results are shown in Fig. 2 and 3. These show the average of 5 runs for different model initialization, with the error bars denoting the standard deviation.

A. Variance Minimization when Training with Limited Data

When training with limited data, our goal is for the model to capture all the patterns present in the dataset. In this case, to evaluate the generalization capability for each dataset, we use only a small fraction of the samples of class 9, 2, 3, 5, and 4 (the rare group). For the remaining classes (the common group), we use all the samples. We retain 10% of samples from each group to form a validation subset which we use to select the hyper-parameters for each method. We recover the generalization capability by evaluating the trained model on the original test subset where each class is equally represented. Of note, we select hyper-parameters that yield the lowest loss variance while maintaining a high accuracy on the validation subset. In this case, using only the validation accuracy is not enough as the common group is overly represented in this set.

Fig. 2a and 2b provide experimental proof of Lemma 3 and show that a lower training loss variance improves generalization which we assess using the test accuracy. Moreover, in all scenarios, when using the optimum set of hyper-parameters, the test accuracy obtained for *Distributional Variance Penalization* was similar to or higher than that of the *Direct Variance Penalization*. In some cases, even up to 5% higher as shown in Fig. 2b. For this reason, we recommend always using *Distributional Variance Penalization* unless negative weights are required.

We compare our method of *Distributional Variance Penalization* with several baselines which similarly change the weight of the samples during training, *Tilted Empirical Risk Minimization (TERM)* [4] for classical crossentropy, and *Focal* loss [35]. As a result of hyper-parameter tuning, the best parameters for *Distributional Variance Penalization* are $\gamma = 0.45$ and $\lambda = 0.75$ when training on Fashion–MNIST, and $\gamma = 0.45$ and $\lambda = 2.4$ when training on CIFAR–10. We found for *TERM* that $t = 0.316$ and $t = 0.562$ yield the best results on Fashion–MNIST and CIFAR-10, respectively. For the *Focal* loss, we found that $\gamma = 2$ was the best fit for both datasets.

Fig. 2c and 2d show the results for the Fashion–MNIST and CIFAR–10 datasets, respectively. Compared to ERM, all methods increase the accuracy evaluated on the test dataset where each class is equally represented. However, only the *Distributional Variance Penalization* registers the highest improvement in all tested scenarios. Our calculated judgment for this is that the other two methods indirectly penalize the variance, i.e., $\gamma = 1$. Whereas Fig. 2b shows that the optimum exponent for the variance term is $\gamma = 0.45$.

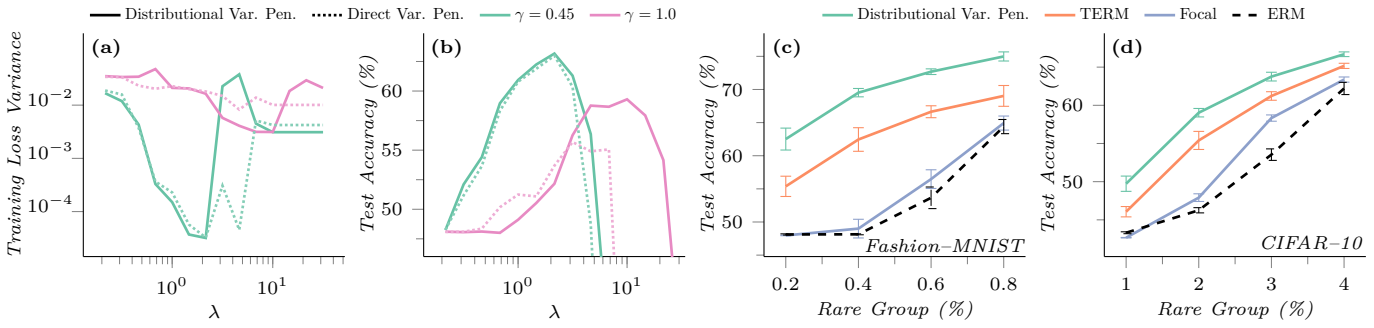

Figure 2: Applying positive variance penalization when training with limited data. The effect of hyper-parameters λ and γ on (a) final loss variance, and (b) test accuracy when training on Fashion–MNIST with the rare group representing 0.2% of the samples. Distributional variance penalization outperforms similar methods on (c) Fashion–MNIST, and (d) CIFAR–10 datasets.

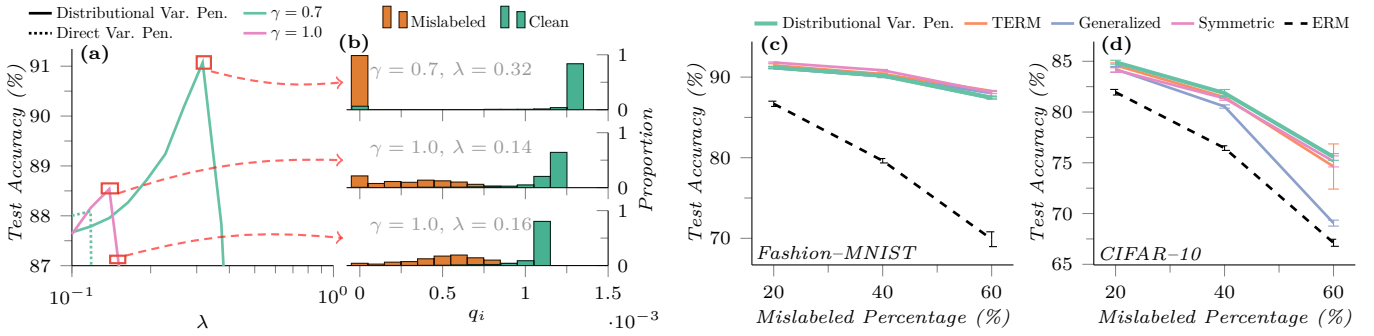

Figure 3: Applying negative variance penalization when training with noisy labels. (a) Direct variance penalization fails whereas distributional variance penalization improves accuracy by 4% when training on Fashion–MNIST with 20% mislabeled samples. (b) Histogram representation of probabilities q_i. Distributional variance penalization is competitive with methods tailored for training with mislabeled data on (c) Fashion–MNIST, and (d) CIFAR–10 datasets.

B. Variance Maximization for Training with Mislabeled Data

To evaluate the robustness when training with mislabeled data, we flip the labels to one of the other classes for each dataset based on the given mislabeled percentage. After that, we retain 10% of the samples to form a validation subset which we use to select the hyper-parameters for each method. By using the test subset of each dataset, we obtain the performance on clean data, labels of which we do not alter.

Fig. 3a and 3b show that *Distributional Variance Penalization*, given the right pair (γ, λ), can disregard the impact of mislabeled samples by assigning them a probability $q_i = 0$. However, in this case, a few correctly labeled but hard-to-learn samples also receive a probability of 0. In terms of hyper-parameter selection, we found that $\gamma = 0.75$ yields the best result in all tested scenarios. Whereas for λ, the higher the percentage of mislabeled samples, the higher its value should be. In particular, for scenarios with 20%, 40%, and 60% of mislabeled samples, we used for λ a value of 0.32, 0.44, and 0.61, respectively. Of note, the *Direct Variance Penalization* often fails to converge. However, when it does converge, it does not improve the validation accuracy nor the test accuracy compared to the ERM case. For this reason, we recommend using the *Distributional Variance Penalization* method instead.

We compare *Distributional Variance Penalization* against state-of-the-art methods, *TERM* [4] for classical crossentropy, *Symmetric* crossentropy [20], and *Generalized* crossentropy [19]. Using hyper-parameter tuning, we found that we obtain the best results for *TERM* when t equals -0.5, -0.7 and -0.9 in scenarios with 20%, 40%, and 60% of mislabeled samples, respectively. The following parameters worked best in all scenarios, for *Generalized* $q = 0.7$, and for *Symmetric* $\alpha = 0.1$ and $\beta = 1$.

Fig. 3c and 3d illustrate the accuracy evaluated on the test subsets which contain only clean labels. We observe that in all presented scenarios, the *Distributional Variance Penalization* obtains similar or higher final accuracy when compared to the specialized methods. Particularly, for the worst-case scenario (Fashion–MNIST with 60% mislabeled), our method trailed the best-performing method by merely 0.6%. Yet for the best-case scenario (CIFAR-10 with 60% mislabeled), our method outperformed all other methods by at least 0.9%.

C. Experimental Setup

We use the same experimental setup as Wang et al. [20]. In particular, for Fashion–MNIST, we trained a model containing 4 convolutional layers followed by 3 fully connected layers. For CIFAR–10, we trained an 8 layer network composed of 6 convolutional layers followed by 2 fully connected layers. The training batch size was 128 samples. The optimizer used was stochastic gradient descent (SGD) with 0.9 momentum and a learning rate of 0.01 which is divided by 10 every 20 epochs for a total of 60 training epochs. Moreover, we used a 0.005 and a 0.01 weight decay for the convolutional and the fully connected layers, respectively. For the complete implementation details, we refer to our GitHub repository (footnote 1).

VIII. Conclusion

This paper aims to present a novel robustness framework based on distributional variance penalization. We reveal two contrasting types of robustness that result from minimizing or maximizing the loss variance. Although we did not cover optimizing simultaneously for both types of robustness, we believe our framework can be extended to cover it. We also emphasize the role of the exponent for the variance term, an often overlooked factor. Lastly, we highlight that the proposed framework makes negative variance penalization a viable training technique when training with mislabeled data.

Acknowledgment

V.B. thanks Yannick Bliesener, Ramy Tadros, and Jenny Kan for the provided insights and expertise that greatly improved this manuscript.

References

[1] Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. *Proc. Computational Learning Theory Conference (COLT)*, 2009.

[2] John Duchi and Hongseok Namkoong. Variance-based regularization with convex objectives. *JMLR*, 2019.

[3] Alexander Robey, Luiz Chamon, George J. Pappas, and Hamed Hassani. Probabilistically robust learning: Balancing average-and worst-case performance. *ICML*, 2022.

[4] Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. Tilted empirical risk minimization. *ICLR*, 2021.

[5] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE TNNLS*, 2022.

[6] Hongseok Namkoong and John C Duchi. Variance-based regularization with convex objectives. In *NeurIPS*, 2017.

[7] Matthew Staib, Bryan Wilder, and Stefanie Jegelka. Distributionally robust submodular maximization. In *AISTATS*, 2019.

[8] Henry Lam. Recovering best statistical guarantees via the empirical divergence-based distributionally robust optimization. *Operations Research*, 67(4):1090–1105, 2019.

[9] Christina Heinze-Deml and Nicolai Meinshausen. Conditional variance penalties and domain shift robustness. *Machine Learning*, 110(2):303–348, 2021.

[10] Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *ICML*, 2018.

[11] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *ICCV*, 2017.

[12] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[13] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *ICML*, 2019.

[14] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *NeurIPS*, 2020.

[15] Hrayr Harutyunyan, Kyle Reing, Greg Ver Steeg, and Aram Galstyan. Improving generalization by controlling label-noise information in neural network weights. In *ICML*, 2020.

[16] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *JASA*, 2006.

[17] Aritra Ghosh, Himanshu Kumar, and PS Sastry. Robust loss functions under label noise for deep neural networks. In *AAAI*, 2017.

[18] X Wang, E Kodirov, Y Hua, and NM Robertson. Improved mean absolute error for learning meaningful patterns from abnormal training data. Technical report, Technical Report, 2019.

[19] Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, 2018.

[20] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *ICCV*, 2019.

[21] Lei Feng, Senlin Shu, Zhuoyi Lin, Fengmao Lv, Li Li, and Bo An. Can cross entropy loss be robust to label noise. In *IJCAI*, 2020.

[22] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *ICML*, 2020.

[23] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *NIPS*, 2013.

[24] Volodymyr Mnih and Geoffrey E Hinton. Learning to label aerial images from noisy data. In *ICML*, 2012.

[25] Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *NeurIPS*, 2019.

[26] Giorgio Patrini, Frank Nielsen, Richard Nock, and Marcello Carioni. Loss factorization, weakly supervised learning and label noise robustness. In *ICML*, 2016.

[27] Brendan van Rooyen, Aditya Krishna Menon, and Robert C Williamson. Learning with symmetric label noise: the importance of being unhinged. In *NeurIPS*, 2015.

[28] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009.

[29] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization (2016). *ICLR*, 2017.

[30] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *ICML*, 2017.

[31] Alexander Shapiro. Distributionally robust stochastic programming. *SIAM Journal on Optimization*, 27(4):2258–2275, 2017.

[32] Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv:1908.05659*, unpublished.

[33] Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *ICML*, 2018.

[34] Abhishek Kumar and Ehsan Amid. Constrained instance and class reweighting for robust learning under label noise. *arXiv:2111.05428*, unpublished.

[35] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.