Robust Learning under Label Noise by Optimizing the Tails of the Loss Distribution

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

Jayson Sia
Department of Electrical and Computer Engineering
University of Southern California
Los Angeles, United States
jsia@usc.edu

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

***Abstract*—We introduce a novel optimization strategy that combines empirical risk minimization (ERM) with a sample weighting scheme to decrease the bias induced by training with noisy labels. We accomplish this by penalizing the tails of the loss distribution, as samples associated with the right tail (high loss values) are more likely to be mislabeled. We implement the proposed sample weighting scheme by minimizing a discriminatory risk that downweighs mislabeled samples. Moreover, we show that the proposed method enables us to optimize the location and shape of the loss distribution simultaneously. In addition, we couple our method with a distributionally robust optimization stage. The goal of this coupling is to increase the weight of correctly labeled but hard-to-learn samples that exhibit a high loss value. This coupling also improves model accuracy on underrepresented minority subsets when training with imbalanced classes. Experimental results show that for 60% noise contamination, our method, when compared to the ERM, improves the final accuracy on both Fashion–MNIST and CIFAR–10 by 18.4% and 8.5%, respectively.**

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

I. Introduction

Empirical risk minimization (ERM) is the typical training strategy for most machine learning applications. However, it is also well-known that it is not robust against noisy data, which can severely bias the model [1]. In practice, label noise is considered more harmful than input noise [2, 3] and is widely present in real-world datasets, which are estimated to have a ratio of corrupted labels between 8% and 38.5% [3]. Many robust learning methods have been proposed to reduce the impact of noisy labels, which either use robust losses [4–10] or learning management [11–15]. In this paper, we introduce a novel learning management method that employs a sample weighting strategy to improve robustness to noisy labels.

Consider the following robust learning setting of Fujisawa and Eguchi [16]. Let $f(x)$ be the probability density function of the clean data, $g(x)$ of the noisy data, and $h(x)$ of the training data with $h(x) = (1-\epsilon)f(x) + \epsilon g(x)$ where $\epsilon \in [0,1]$ is the noise contamination rate. Let $\{x_i\}_{i=1}^{n}$ be the training samples independently drawn from h and let $\ell(\theta, x)$ be the loss function where θ are the model parameters. In the ERM case, we select θ that minimizes the empirical risk $\mathbb{E}_{\mathbb{P}_n}[\ell(\theta,x)] = \sum_{i=1}^{n} \frac{1}{n}\,\ell(\theta, x_i)$. Since the expectation is taken with respect to the empirical distribution $\mathbb{P}_n$, all samples receive the same weight $\frac{1}{n}$, including the noisy ones drawn from g, see Fig. 1a.

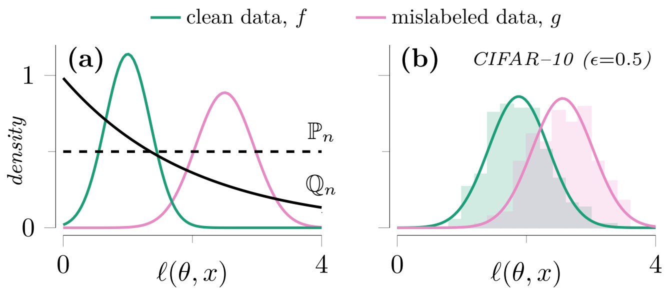

Figure 1: (a) Comparison between weighting scheme of ERM, $\mathbb{P}_n$, with $q_i = \frac{1}{n}$ for all i, and the robust learning one, $\mathbb{Q}_n$, with $q_i > 0$ for samples drawn from f, and $q_i \approx 0$ for the ones drawn from g. (b) Distribution of loss values after 50 training steps for *CIFAR–10* with 50% mislabeled data.

On the other hand, in the robust learning case, we aim to select θ that minimizes a discriminatory risk $\mathbb{E}_{\mathbb{Q}_n}[\ell(\theta, x)]$, or:

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

where $q_i > 0$ if x_i is drawn from f, and $q_i \approx 0$ if x_i is drawn from g. The distribution $\mathbb{Q}_n$ discriminates against noisy labels by giving those samples a probability close to zero. Many methods exist for estimating the distribution $\mathbb{Q}_n$ [3]; however, in this paper, we present a new approach that estimates this distribution from the loss values $\ell(\theta, x)$. Prior research has shown theoretically [17] and experimentally [2, 18] that models in the first few training iterations fit the correct labels, and that the overfitting on noisy labels happens in the latter stages of training. This supports the common wisdom that samples with small loss should be favored when training with noisy labels so that the model learns the clean data first [3]. Consequently, to improve robustness, we propose to decrease the weight of samples with large loss values proportionally to their loss, see Fig. 1a. Here, similar to Fujisawa and Eguchi [16], we assume that the density of loss values $\ell(\theta, x)$ of mislabeled data lies on the tail of the clean data loss distribution. This assumption holds in practice, see Fig. 1b, since models learn the correct labels first. However, when it does not, we can pre-train the model on a small but clean dataset to initialize the weights [17].

We obtain the weighting scheme shown in Fig. 1a by penalizing the tails of the loss distribution, i.e., maximizing the central moments of even order, such as variance and kurtosis, and introducing a right skew in the distribution of loss values, i.e., making the right tail longer.

The main contributions of our paper are as follows:

(**C1**) We present a novel optimization strategy for controlling the tails of the loss distribution using Alpha-divergence (Theorems 1 and 2) and Beta-divergence (Theorem 5).

(**C2**) We show that our proposed *Distributional Moments Penalization* method simultaneously optimizes the location and the shape of the loss distribution by penalizing the mean and the central moments (Theorems 3 and 4).

(**C3**) We couple our method with a distributionally robust optimization stage which improves performance on underrepresented minority subsets, Equation (6).

(**C4**) We conduct extensive experiments showing that our methods successfully reject the outliers even in scenarios with heavy contamination.

In this paper, we focus our investigation on classification problems using deep neural networks. However, the mathematical framework can be directly applied to other machine learning problems where the mean of a loss function is minimized, such as regression, clustering, and ranking problems.

The paper is structured as follows. In Section II, we present the theoretical results for selecting the distribution $\mathbb{Q}_n$. In Section III, we provide efficient algorithms for finding the distribution $\mathbb{Q}_n$. Lastly, we present the experimental results in Section IV and the concluding remarks in Section V.

II. Distributional Moments Penalization

We expand on the problem setup we briefly presented in the Introduction. Let the training data for a classification task be distributed according to an unknown joint distribution $\mathbb{P}$ and consisting of n training samples. Each sample is represented as a tuple (x, y) where x depicts the attributes drawn from $\mathcal{X} \subseteq \mathbb{R}^d$, and y the class labels drawn from $\mathcal{Y} = \{1, \ldots, C\}$, where C is the number of classes. However, the class label data can be contaminated by noise. Let y represent the noise-contaminated labels and y^* the clean labels, we have $\mathbb{P}(y \neq y^*) = \epsilon$ where the contamination rate $\epsilon > 0$. Given a model parameterizable by $\theta \in \Theta \subseteq \mathbb{R}^t$, our goal is to find the optimum set of parameters for which the model correctly predicts the clean 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 $\mathbb{E}_{\mathbb{P}}[\ell(\theta, x, y^*)]$. However, throughout this paper, we assume that we do not have access to clean data, and thus, we can only minimize $\mathbb{E}_{\mathbb{P}_n}[\ell(\theta, x, y)]$, where $\mathbb{P}_n$ is the empirical distribution. Our goal is to develop a robust training procedure for data with mislabeled samples yielding a model with a similar misclassification probability as a model trained on clean data.

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

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

where w_i are the sample weights with $w_i = \frac{1}{n}, \forall i$. This simplifies the implementation but hinders robustness to noise. Next, we introduce the weighting scheme shown in Fig. 1a, which uses Alpha-divergence to prioritize samples with small loss values and downweight the ones with a large loss.

A. Sample Weighting Scheme Based on Alpha-divergence

Divergences are functions that measure the differences between two probability distributions $\mathbb{P}$ and $\mathbb{Q}$ and satisfy the condition $D(\mathbb{Q} \,\|\, \mathbb{P}) \geq 0$ with equality if and only if $\mathbb{P} = \mathbb{Q}$. In general, divergences do not satisfy the symmetry condition, i.e., $D(\mathbb{Q} \,\|\, \mathbb{P}) \neq D(\mathbb{P} \,\|\, \mathbb{Q})$, and thus the order of arguments matters. Moreover, the second argument indicates the reference distribution, which throughout this paper will be the empirical distribution $\mathbb{P}_n$, as it allows a simpler closed-form solution.

To obtain the weighting scheme shown in Fig. 1a, we will use Alpha-divergence to bound the deviation of the distribution $\mathbb{Q}_n$ from the empirical one. The asymmetric Alpha-divergence is parameterized by α and represents a special case of f-divergence. Throughout this paper, we will use the f-divergence formulation of Alpha-divergence [19] defined as:

$$D_\alpha(\mathbb{Q}_n \,\|\, \mathbb{P}_n) = \sum_{i=1}^{n} p_i f_\alpha\left(\frac{q_i}{p_i}\right), \tag{3}$$

where $f_\alpha(t) = \frac{t^\alpha - 1}{\alpha(\alpha-1)}$ is the generating function. Note, as Alpha-divergence is a special case of f-divergence, it inherits all the properties such as convexity in both arguments and continuity [19]. Notable members of the Alpha-divergence family are Kullback–Leibler divergence (or KL-divergence) obtained for $\alpha = 1$ and Pearson χ^2-divergence for $\alpha = 2$.

The following theorem incorporates Alpha-divergence to obtain the weighting scheme shown in Fig. 1a, where α controls the shape of the decay curve.

Theorem 1 (Alpha-Divergence Constrained Optimization, Kumar and Amid [20]). *Let $\mathbb{P}_n$ be the discrete empirical distribution, $\mathcal{A}$ its neighborhood of radius $\rho > 0$ and $\alpha \geq 1$ where $\mathcal{A} = \{\mathbb{Q}_n : D_\alpha(\mathbb{Q}_n \,\|\, \mathbb{P}_n) \leq \rho\}$. We have:*

$$\min_{\mathbb{Q}_n \in \mathcal{A}} \mathbb{E}_{\mathbb{Q}_n}\left[\ell(\theta, x, y)\right] = \sum_{i=1}^{n} q_i \ell(\theta, x_i, y_i)$$

with the probabilities q_i computed as:

$$q_i = \begin{cases} \dfrac{\exp\left(-\ell(\theta, x_i, y_i)/\delta\right)}{\sum_{j=1}^{n} \exp\left(-\ell(\theta, x_j, y_j)/\delta\right)}, & \alpha = 1, \\[2ex] \dfrac{[-\ell(\theta, x_i, y_i) + \delta]_+^{\frac{1}{\alpha-1}}}{\sum_{j=1}^{n} [-\ell(\theta, x_j, y_j) + \delta]_+^{\frac{1}{\alpha-1}}}, & \alpha > 1, \end{cases}$$

where $[x]_+ = \max(0, x)$ and δ is selected such that the obtained probabilities q_i satisfy $\sum_{i=1}^{n} \frac{1}{n} f_\alpha(nq_i) = \rho$.

Proof. We reproduce the proof from Kumar and Amid [20] as the steps of the proof will be referenced later in the paper. For compactness, 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. $q_i \geq 0, \sum_{i=1}^{n} q_i = 1, \sum_{i=1}^{n} \frac{f_\alpha(nq_i)}{n} \leq \rho.$

(ii) $q_i = \frac{1}{n} f_\alpha'^{-1}\left(\frac{-z_i - \mu + \nu_i}{\lambda}\right)$
where λ, μ, ν_i are the Lagrange multipliers.
We obtain this by first forming the Lagrangian function

$$\sum_{i=1}^{n} q_i z_i + \lambda\left[\sum_{i=1}^{n} \frac{f_\alpha(nq_i)}{n} - \rho\right] + \mu\left[\sum_{i=1}^{n} q_i - 1\right] - \sum_{i=1}^{n} \nu_i q_i.$$

Differentiating with respect to q_i and equating to 0 yields $z_i + \lambda f_\alpha^{'}(nq_i) + \mu - \nu_i = 0$, which we solve for q_i to obtain the result of this step.

(iii) $q_i = \frac{\exp\left(-\ell(\theta, x_i, y_i)/\delta\right)}{\sum_{j=1}^{n} \exp\left(-\ell(\theta, x_j, y_j)/\delta\right)}$ for $\alpha = 1$.
For $\alpha = 1$, $f_\alpha(t) = t\log(t)$ and $f_\alpha^{'-1}(t) = \exp(t-1)$. Since $f_\alpha^{'-1}(t) > 0, \forall t$ the non-negative constraint is never active, $\nu_i = 0$. Moreover, we can write $q_i = \frac{q_i}{\sum_{j=1}^{n} q_j}$ as the denominator sums to 1. Replacing q_i and q_j with the expression from step (ii), denoting λ with δ, then simplifying the fraction yields the result of this step.

(iv) $q_i = \frac{[-\ell(\theta, x_i, y_i) + \delta]_+^{\frac{1}{\alpha-1}}}{\sum_{j=1}^{n} [-\ell(\theta, x_j, y_j) + \delta]_+^{\frac{1}{\alpha-1}}}$ for $\alpha \neq 1$.
For $\alpha \neq 1$, $f_\alpha(t) = \frac{t^\alpha - 1}{\alpha - 1}$ and $f_\alpha^{'-1}(t) = [(\alpha-1)t]^{\frac{1}{\alpha-1}}$. When non-negative constraint is active, $\nu_i \neq 0$, the resulting probability will be 0 and thus $f_\alpha^{'-1}(t) = [(\alpha-1)t]_+^{\frac{1}{\alpha-1}}$. Similar to previous step we can write $q_i = \frac{q_i}{\sum_{j=1}^{n} q_j}$ as the denominator sums to 1. Replacing q_i and q_j with the expression from step (ii), letting $\delta = -\mu$, then simplifying the fraction yields the result of this step.

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

Remarks. *To increase the weight of samples with small loss values, we select a distribution $\mathbb{Q}_n$ that minimizes the expectation. However, selecting $\mathbb{Q}_n$ requires finding δ that satisfies the divergence constraint. This can be efficiently done using the bisection algorithm [21, 22] as the divergence value is inversely proportional with δ.*

We introduce *Distributional Moments Penalization* method that optimizes the objective of Theorem 1, and at each iteration, δ is selected such that $\mathbb{Q}_n$ minimizes the expectation. This is different from the approach of Kumar and Amid [20], where δ is a hyper-parameter selected before training. In this case, the method does not respond to changes in the loss distribution that naturally happen during training.

Fig. 2 illustrates how the shape parameter α and the rate parameter ρ impact the curvature of the distribution $\mathbb{Q}_n$. From Fig. 2a, we can see that as the loss value increases, the decay in probability is exponential for $\alpha = 1$, and linear for $\alpha = 2$. Whereas the parameter ρ controls the rate of decay, see Fig. 2b. The higher the value of ρ, the more we neglect the samples with large loss values. Note, for $\alpha > 1$, samples with loss values higher than δ will receive zero weights due to the $[\cdot]_+$ operator. Thus, such samples will not impact model parameters.

We extend the result of Theorem 1 to find the distribution $\mathbb{Q}_n$ that maximizes the expectation. It has the opposite effect of prioritizing samples with large loss values, see Fig. 2c.

Theorem 2. *Let $\mathbb{P}_n$ be the discrete empirical distribution and $\mathcal{A}$ be its neighborhood of radius ρ, where $\mathcal{A} = \{\mathbb{Q}_n : D_\alpha(\mathbb{Q}_n \,\|\, \mathbb{P}_n) \leq \rho\}$ with $\alpha \geq 1$. Then probabilities q_i that solve $\max_{\mathbb{Q}_n \in \mathcal{A}} \mathbb{E}_{\mathbb{Q}_n}[\ell(\theta, x, y)]$ have the following expression:*

$$q_i = \begin{cases} \dfrac{\exp\left(\ell(\theta, x_i, y_i)/\delta\right)}{\sum_{j=1}^{n} \exp\left(\ell(\theta, x_j, y_j)/\delta\right)}, & \alpha = 1, \\ \dfrac{[\ell(\theta, x_i, y_i) - \delta]_+^{\frac{1}{\alpha-1}}}{\sum_{j=1}^{n} [\ell(\theta, x_j, y_j) - \delta]_+^{\frac{1}{\alpha-1}}}, & \alpha > 1, \end{cases}$$

where δ is selected such that $\sum_{i=1}^{n} \frac{1}{n} f_\alpha(nq_i) = \rho$.

Proof. The steps of this proof are identical to the ones of Theorem 1. The only difference is we maximize the expectation; hence, the first sum in the Lagrangian function from step (ii) now has a minus sign in front. This changes the result of step (ii) to $q_i = \frac{1}{n} f_\alpha'^{-1}\left(\frac{z_i - \mu + \nu_i}{\lambda}\right)$; note the absence of the negative sign in front of z_i. Following steps (iii) and (iv) using the new expression for q_i, we obtain the final result of this theorem. □

By minimizing the expectation taken with respect $\mathbb{Q}_n$, we minimize both the empirical mean $\mathbb{E}_{\mathbb{P}_n}[\ell(\theta, x, y)]$ and linear combination of central moments that control the shape of the loss distribution. The following two theorems illustrate this connection for $\alpha = 1$ (KL-divergence) and $\alpha > 1$, respectively.

Theorem 3 (Central Moments Expansion of KL-Divergence)**.** *Let z be a random variable with $z_i = \ell(\theta, x_i, y_i)$ and $\mathcal{A} = \{\mathbb{Q}_n : D_{KL}(\mathbb{Q}_n \,\|\, \mathbb{P}_n) \leq \rho\}$ be the neighborhood of the empirical distribution $\mathbb{P}_n$ of radius ρ, and δ selected according to Theorem 1, then the following expansion holds:*

$$\min_{\mathbb{Q}_n \in \mathcal{A}} \mathbb{E}_{\mathbb{Q}_n}[z] = c_1 \mathbb{E}_{\mathbb{P}_n}[z] + \sum_{m=2}^{\infty} c_m \mathbb{E}_{\mathbb{P}_n}\left[(z - \mathbb{E}_{\mathbb{P}_n}[z])^m\right]$$

where $c_1 = \frac{1}{\mathbb{E}\left[\exp\left(-\frac{z - \mathbb{E}[z]}{\delta}\right)\right]}$ and $c_m = \frac{(-1)^{m-1}(m\delta - \mathbb{E}[z])}{m!\delta^m \mathbb{E}\left[\exp\left(\frac{\mathbb{E}[z] - z}{\delta}\right)\right]}$.

Proof. Here, all the expectations are taken with respect to the empirical distribution $\mathbb{P}_n$, i.e., $\mathbb{E}[z]$ implies $\mathbb{E}_{\mathbb{P}_n}[z]$.

(i) $q_i = \frac{\exp\left(-(z_i - \mathbb{E}[z])/\delta\right)}{\sum_{j=1}^{n} \exp\left(-(z_j - \mathbb{E}[z])/\delta\right)}$.
By multiply q_i formula from Theorem 1 with $\frac{\exp(\mathbb{E}[z]/\delta)}{\exp(\mathbb{E}[z]/\delta)}$.

(ii) $\exp\left(-\frac{z_i - \mathbb{E}[z]}{\delta}\right) = \sum_{m=0}^{\infty} b_m (z - \mathbb{E}[z])^m$
where $b_m = \left(\frac{-1}{\delta}\right)^m \frac{1}{m!}$. By Taylor series expansion.

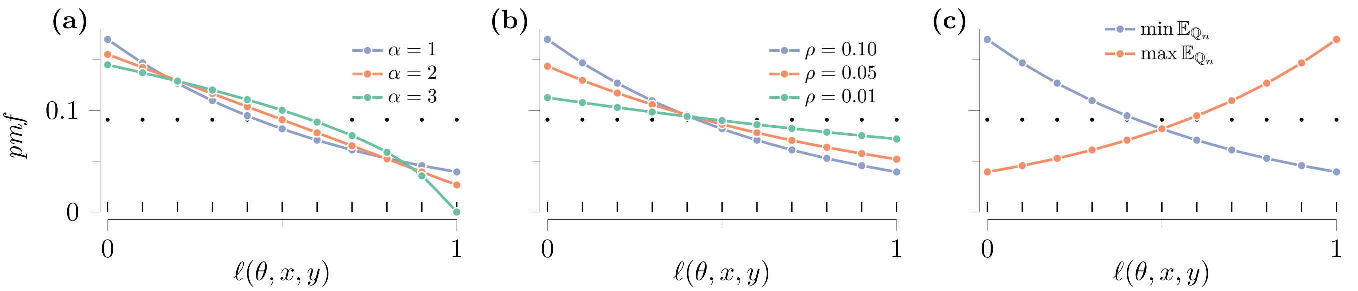

Figure 2: Comparison of different distributions $\mathbb{Q}_n$ obtained using the proposed method based on Theorem 1. Distributions were computed based on the loss values of 11 samples shown with black vertical bars that are uniformly distributed between 0 and 1. The black dots indicate the probability mass of the empirical distribution $\mathbb{P}_n$. (a) Comparison between $\mathbb{Q}_n$ distributions obtained for $\rho = 0.1$ and various α. (b) Comparison between $\mathbb{Q}_n$ distributions obtained for $\alpha = 1$ and various ρ. (c) Comparison between $\mathbb{Q}_n$ distributions that minimize (Theorem 1) and maximize (Theorem 2) the expectation for $\rho = 0.1$ and $\alpha = 1$.

(iii) $\displaystyle\sum_{j=1}^{n} \exp\left(-\frac{z_j - \mathbb{E}[z]}{\delta}\right) = nM$

where $M = \mathbb{E}\left[\exp\left(-\frac{z-\mathbb{E}[z]}{\delta}\right)\right]$.

(iv) $\displaystyle q_i = \sum_{m=0}^{\infty} \frac{b_m}{nM}(z - \mathbb{E}[z])^m$ by replacing the expressions from steps (ii) and (iii) into the expression from step (i).

(v)
$$\begin{aligned}\sum_{i=1}^{n} q_i z_i &= \sum_{i=1}^{n}\sum_{m=0}^{\infty} \frac{b_m}{nM}(z - \mathbb{E}[z])^m z_i \\ &= \sum_{i=1}^{n}\sum_{m=0}^{\infty} \frac{b_m}{nM}(z - \mathbb{E}[z])^m (z_i - \mathbb{E}[z] + \mathbb{E}[z]) \\ &= \sum_{m=0}^{\infty} \frac{b_m}{M}\left[\mathbb{E}[(z - \mathbb{E}[z])^{m+1}] + \mathbb{E}[z]\mathbb{E}[(z - \mathbb{E}[z])^m]\right] \\ &= \frac{b_0}{M}\mathbb{E}[z] + \sum_{m=2}^{\infty} \frac{b_{m-1} + b_m\mathbb{E}[z]}{M}\mathbb{E}[(z - \mathbb{E}[z])^m]\end{aligned}$$

where in the last step, we group the terms based on the order m of the central moment and drop the terms with $m = 1$ since $\mathbb{E}[(z - \mathbb{E}[z])] = \mathbb{E}[z] - \mathbb{E}[z] = 0$.

(vi)
$$c_1 = \frac{b_0}{M} = \frac{1}{\mathbb{E}\left[\exp\left(-\frac{z-\mathbb{E}[z]}{\delta}\right)\right]},$$
$$c_m = \frac{b_{m-1} + b_m\mathbb{E}[z]}{M} = \frac{(-1)^{m-1}(m\delta - \mathbb{E}[z])}{m!\delta^m \mathbb{E}\left[\exp\left(\frac{\mathbb{E}[z]-z}{\delta}\right)\right]}.$$

□

Theorem 4 (Central Moments Expansion of Alpha-Divergence). *Let z be a random variable with $z_i = \ell(\theta, x_i, y_i)$, $\mathcal{A}$ be the neighborhood of the empirical distribution $\mathbb{P}_n$ of radius ρ with $\mathcal{A} = \{\mathbb{Q}_n : D_\alpha(\mathbb{Q}_n \,\|\, \mathbb{P}_n) \leq \rho\}$ for $\alpha > 1$ and define $\alpha^* = \frac{1}{\alpha-1}$. If $\forall i$ we have $\delta - z_i \geq 0$ and $\left|\frac{\mathbb{E}[z]-z_i}{\delta-\mathbb{E}[z]}\right| < 1$ with δ selected according to Theorem 1, then:*

$$\min_{\mathbb{Q}_n \in \mathcal{A}} \mathbb{E}_{\mathbb{Q}_n}[z] = c_1\mathbb{E}_{\mathbb{P}_n}[z] + \sum_{m=2}^{\infty} c_m \mathbb{E}_{\mathbb{P}_n}\left[(z - \mathbb{E}_{\mathbb{P}_n}[z])^m\right]$$

where $c_1 = \frac{(\delta-\mathbb{E}[z])^{\alpha^*}}{\mathbb{E}\left[(\delta-z)^{\alpha^*}\right]}$ *and* $c_m = \frac{(-1)^{m-1}\delta^{\alpha^*-m}\left(\binom{\alpha^*}{m-1}\delta - \binom{\alpha^*}{m}\mathbb{E}[z]\right)}{\mathbb{E}\left[(\delta-z)^{\alpha^*}\right]}$.

Proof. In what follows, all expectations are taken with respect to the empirical distribution $\mathbb{P}_n$, i.e., $\mathbb{E}[z]$ implies $\mathbb{E}_{\mathbb{P}_n}[z]$.

(i) $\displaystyle q_i = \frac{(-z_i + \delta)^{\alpha^*}}{\sum_{i=1}^{n}(-z_i + \delta)^{\alpha^*}}$.

We use q_i formula from Theorem 1, drop the non-negative constraint as $-z_i + \delta \geq 0$, and replace $\frac{1}{\alpha-1}$ with α^*.

(ii) $\displaystyle(-z_i + \delta)^{\alpha^*} = \sum_{m=0}^{\infty} b_m (z - \mathbb{E}[z])^m$

where $b_m = \dbinom{\alpha^*}{m}(-1)^m(\delta - \mathbb{E}[z])^{\alpha^*-m}$.

By using binomial theorem on $(-z_i + \mathbb{E}[z] - \mathbb{E}[z] + \delta)^{\alpha^*}$. Note, the above series always converges for integer α^*, and in this case the series stops at $m = \alpha^*$ since the binomial coefficients for $m > \alpha^*$ are 0. Moreover, since from theorem statement we have $\left|\frac{\mathbb{E}[z]-z_i}{\delta-\mathbb{E}[z]}\right| < 1$, it means the series also converges for non-integer α^*.

(iii) $\displaystyle\sum_{j=1}^{n}(-z_j + \delta)^{\alpha^*} = nM$ where $M = \mathbb{E}\left[(\delta - z)^{\alpha^*}\right]$.

(iv) $\displaystyle q_i = \sum_{m=0}^{\infty} \frac{b_m}{nM}(z - \mathbb{E}[z])^m$ by replacing the expressions from steps (ii) and (iii) into the expression from step (i).

(v)
$$\begin{aligned}\sum_{i=1}^{n} q_i z_i &= \sum_{i=1}^{n}\sum_{m=0}^{\infty} \frac{b_m}{nM}(z - \mathbb{E}[z])^m z_i \\ &= \sum_{i=1}^{n}\sum_{m=0}^{\infty} \frac{b_m}{nM}(z - \mathbb{E}[z])^m (z_i - \mathbb{E}[z] + \mathbb{E}[z]) \\ &= \frac{b_0}{M}\mathbb{E}[z] + \sum_{m=2}^{\infty} \frac{b_{m-1} + b_m\mathbb{E}[z]}{M}\mathbb{E}[(z - \mathbb{E}[z])^m]\end{aligned}$$

where in the last step, we group the terms based on the order m of the central moment and drop the terms with $m = 1$ since $\mathbb{E}[(z - \mathbb{E}[z])] = \mathbb{E}[z] - \mathbb{E}[z] = 0$. Note, this is a finite series for integer α^*, the series stops at $m = \alpha^* + 1$ where $b_{\alpha^*+1} = 0$ since the binomial coefficients for $m > \alpha^*$ are 0.

(vi) $c_1 = \dfrac{b_0}{M} = \dfrac{1}{\mathbb{E}\left[(-z+\delta)^{\alpha^*}\right]}$, $c_m = \dfrac{b_{m-1} + b_m \mathbb{E}[z]}{M}$ or

$$c_m = \frac{(-1)^{m-1}\delta^{\alpha^*-m}\left(\binom{\alpha^*}{m-1}\delta - \binom{\alpha^*}{m}\mathbb{E}[z]\right)}{\mathbb{E}\left[(-z+\delta)^{\alpha^*}\right]} \text{ for } m \geq 2.$$

□

Remarks. *As the only difference between Theorems 1 and 2 is the presence or absence of a minus sign in the expression of q_i, similar expansions in terms of central moments can also be obtained for Theorem 2. Condition $\left|\frac{\mathbb{E}[z]-z_i}{\delta-\mathbb{E}[z]}\right| < 1$ is only required for non-integer α^* to ensure the binomial series converges. In addition, the condition $\delta - \ell(\theta, x_i, y_i) \geq 0$ is satisfied for small ρ when the distribution $\mathbb{Q}_n$ does not diverge significantly from $\mathbb{P}_n$ and thus $\delta \gg \ell(\theta, x, y)$.*

Note, when the condition $\delta - \ell(\theta, x_i, y_i) \geq 0$ is not satisfied for some samples i, the weight assigned to those samples is 0 based on the results of Theorem 1. Consequently, those samples are not considered during training. Furthermore, only the central moments of samples that satisfy the above condition are optimized during training. In particular, we want to select ρ such that all outliers receive zero, or close to zero, weight and only the central moments of clean data are optimized.

To the best of our knowledge, this is the first direct connection between such optimization objectives and the penalization of central moments. Prior works only revealed the link between χ^2-divergence ($\alpha = 2$) and penalizing the standard deviation [21] or variance [22].

B. Sample Weighting Scheme Based on Beta-divergence

An equally popular divergence family is the Bregman divergence. Similar to f-divergence, Bregman divergence is a pseudo-distance for measuring the difference between two distributions. For the discrete case, it is defined as $D_\phi(\mathbb{Q}_n \,\|\, \mathbb{P}_n) = \sum_{i=1}^n \phi(p_i) - \phi(q_i) - (p_i - q_i)\phi'(q_i)$, where ϕ is the generating function and must be strictly convex, and $\phi'(q_i)$ indicates the derivative with respect to q_i. Beta-divergence, introduced by Basu et al. [23] and Mihoko and Eguchi [24], is obtained from the Bregman divergence using the following strictly convex function $f_\beta(t) = \frac{t^\beta - 1}{\beta(\beta-1)}$. Of note, this is the same generating function used for Alpha-divergence with $\alpha = \beta$. Alternatively, Beta-divergence can be obtained from a generalized f-divergence [19]:

$$D_\beta(\mathbb{Q}_n \,\|\, \mathbb{P}_n) = \sum_{i=1}^n p_i^\beta f_\beta\left(\frac{q_i}{p_i}\right). \tag{4}$$

Notable members of the Beta-divergence family are KL-divergence obtained for $\beta = 1$, and the standard squared Euclidean distance (L_2-norm) obtained for $\beta = 2$.

Surprisingly, we obtain the same weighting scheme if we replace Alpha-divergence in Theorem 1 with Beta-divergence.

Theorem 5 (Beta-divergence Constrained Optimization)**.** *Let $\mathbb{P}_n$ be the discrete empirical distribution and $\mathcal{A}$ be its neighborhood, $\mathcal{A} = \{\mathbb{Q}_n : D_\beta(\mathbb{Q}_n \,\|\, \mathbb{P}_n) \leq \rho\}$ with $\beta \geq 1$, then for probabilities q_i that solve $\min_{\mathbb{Q}_n \in \mathcal{A}} \mathbb{E}_{\mathbb{Q}_n}[\ell(\theta, x, y)]$ we have:*

$$q_i = \begin{cases} \dfrac{\exp\left(-\ell(\theta, x_i, y_i)/\delta\right)}{\sum_{j=1}^n \exp\left(-\ell(\theta, x_j, y_j)/\delta\right)}, & \beta = 1, \\[2ex] \dfrac{[-\ell(\theta, x_i, y_i) - \delta]_+^{\frac{1}{\beta-1}}}{\sum_{j=1}^n [-\ell(\theta, x_j, y_j) - \delta]_+^{\frac{1}{\beta-1}}}, & \beta > 1, \end{cases}$$

where δ is selected such that $\sum_{i=1}^n \left(\frac{1}{n}\right)^\beta f_\beta(nq_i) = \rho$.

Proof. The steps of the proof are identical to Theorem 1. Note, for step (ii), expressing Beta-divergence as a generalized f-divergence when forming the Lagrangian and differentiating with respect to q_i yields the same result as in Theorem 1. Thus, following the same steps (iii) and (iv) yields the same final result. □

Remarks. *Since probabilities q_i have the same expression for both Alpha and Beta-divergence, the same algorithm for finding the value of δ that satisfies the divergence constraint can be used in both cases. Moreover, the central moments expansion of Theorems 3 and 4, and the extension to maximization of Theorem 2 also hold in the Beta-divergence case.*

Note that the results of Theorems 1 and 5 are the same since the reference distribution used in both cases is the empirical one. Moreover, the two produce the same results up to a multiplicative constant $\frac{1}{n^{\alpha-1}}$:

$$D_\beta(\mathbb{Q}_n \,\|\, \mathbb{P}_n) = \frac{1}{n^{\beta-1}} \sum_{i=1}^n \frac{1}{n} f_\beta(nq_i) = \frac{D_\alpha(\mathbb{Q}_n \,\|\, \mathbb{P}_n)}{n^{\alpha-1}}. \tag{5}$$

In the remainder of the paper, we will only use Alpha-divergence, as similar results can be obtained for Beta-divergence by adjusting the value of ρ accordingly.

C. Incorporating Distributional Robust Optimization

The biggest limitation of the proposed *Distributional Moments Penalization* method is that by decreasing the weight of noisy samples, we also inherently decrease the weight of clean samples with large loss values, see Fig. 1a. We can counteract part of this effect by including an additional step that increases the weight of clean samples with a loss value that is large, but still below the value of most mislabeled samples. We can do this by maximizing the expectation using the result of Theorem 2. In particular, let the probabilities $q^{(\eta)}$ be the solution to $\min_{\mathbb{Q}_n^{(\eta)} \in \mathcal{A}} \mathbb{E}_{\mathbb{Q}_n}[\ell(\theta, x, y)]$ (the result of Theorem 1) that assign zero weight to most of the noisy samples, then we improve distributional robustness by maximizing the expectation:

$$\max_{\mathbb{Q}_n^{(c)} \in \mathcal{B}} \mathbb{E}_{\mathbb{Q}_n}\left[nq^{(\eta)}\ell(\theta, x, y)\right] \tag{6}$$

where $\mathcal{A}$ and $\mathcal{B}$ are two non-necessarily distinct neighborhoods, and n is the number of samples used to compute the expectation. Note, for this step to work, it is crucial that $q^{(\eta)}$ assigns zero weight to the majority of the noisy samples. Otherwise, the impact of noisy samples will be amplified since maximizing the expectation increases the weight of samples with a larger loss which are more likely to be noisy, see Fig. 2c. Note, for $\alpha = 1$ due to the exponentiation, none of the probabilities would ever be 0, and thus, a threshold must be used to set probabilities below it to zero. Performing the maximization step is equivalent to performing distributionally robust optimization, which protects against deviations from a nominal model [25, 26], leads to improved robustness against distributional shift [27], and improves model performance on underrepresented minority subsets when training with imbalanced classes [22].

D. Literature review.

Methods for training with noisy labels are classified into two broad categories: noise model-free and model-based strategies. Noise model-free methods aim to reduce the impact of outliers without relying on a noise model. They can be further split into two main subcategories: using robust losses [1, 4–10] and using learning management such as meta-learning or regularization [11–15]. On the other hand, noise model-based methods estimate a noise model and use it to mitigate the bias induced by noisy samples [28–33]. Ghosh et al. [1] outlined distribution-independent and sufficient conditions for a loss function to be robust. The loss function must satisfy the equality $\sum_{j=1}^{C} \ell(\theta, x_i, j) = K$ where K is any arbitrary constant. The commonly used loss for classification, crossentropy loss, does not satisfy this equality and is severely susceptible to mislabeled data. Recent studies [4–8] explored relaxing the above equality by making it an inequality. This helped balance robustness with convergence. In particular, Balaban et al. [22] showed that for $\alpha = 2$, prioritizing samples with small loss values as in Theorem 1 is equivalent to bounding the above constraint.

Many models perform much worse on minority subsets underrepresented in the dataset despite a strong average performance. Distributionally robust optimization aims to equalize the performance across subsets by perturbing the data generating distribution [26], which can be done by bounding the perturbations using Alpha-divergence [21, 22, 34] or focusing on the worst performing samples [35].

III. Implementation Details

Applying the method in practice requires including an additional step for finding the discrete distribution $\mathbb{Q}_n$. Algorithm 1 illustrates the training procedure. To implement the version with the distributional robust optimization stage, Equation (6), call $\textsc{Find}\mathbb{Q}_n$ function twice, second time with $q^{(\eta)}z$ as argument.

Algorithm 2 shows the function $\textsc{Find}\mathbb{Q}_n$ that solves the stochastic optimization problem of Theorems 1, 2 and 5. Here, the function $\textsc{Bracket}(f, a, b)$ given the initial guess $[a, b]$ returns an interval in which the function f changes the sign; and, the function $\textsc{Bisection}(f, a, b)$ given a bracketed interval $[a, b]$ finds the root of function f. This algorithm generalizes the one proposed by Balaban et al. [22] to any arbitrary Alpha-divergence with $\alpha \geq 1$. This algorithm can be further improved by: *(i)* replacing bisection with a faster root finding algorithm such as Brent-Dekker, and *(ii)* caching the value of δ for a few steps as it changes gradually during training. To apply the algorithm as intended, the operations used for calculating q_i must not be included in the computational graph. For that, use `detach()` function in Pytorch, or `stop_gradient()` in Tensorflow when passing the loss values z_i to $\textsc{Find}\mathbb{Q}_n$ function. We provide a Pytorch implementation on GitHub.[1]

Algorithm 1: Distributional Moments Penalization

```
Input:  $\{x_i, y_i\}_1^n$    – training data
        $\ell(\theta, x, y)$  – loss function
        $\rho,\ \alpha$       – hyper-parameters
        $\lambda$             – learning rate
while stopping criteria not reached do
    for $i \leftarrow 1$ to $n$ do
        $z_i \leftarrow \ell(\theta, x_i, y_i)$
    $q \leftarrow \textsc{Find}\mathbb{Q}_n(z, \rho, \alpha)$
    $\mathcal{L} \leftarrow \sum_{i=1}^{n} q_i z_i$
    $\theta \leftarrow \theta - \lambda \nabla_\theta \mathcal{L}$
```

Algorithm 2: $\textsc{Find}\mathbb{Q}_n$ Function

```
Input :  $z$            – loss values
         $\rho, \alpha$ – hyper-parameters
Param:   $\mathbb{Q}_n(z, \delta)$ – $q_i$ expression (Theorems 1, 2 and 5)
$f(x) \leftarrow D_\alpha(\mathbb{Q}_n(z, x) \,\|\, \mathbb{P}_n) - \rho$
$a, b \leftarrow \textsc{Bracket}(f, \min(z), \max(z))$
$\delta \leftarrow \textsc{Bisection}(f, a, b)$
return $\mathbb{Q}_n(z, \delta)$
```

IV. Experiments

The following setup was applied to all the conducted experiments. We compare the performance on two image datasets: *(i)* Fashion–MNIST, and *(ii)* CIFAR–10, both with 10 label classes. The noisy versions of those datasets were obtained by flipping the training labels to one of the other classes. We retained 10% of the training samples for validation and hyper-parameter selection. The test subset was not corrupted and was used to obtain the test accuracy on clean data. All experiments were repeated five times using different random seeds to generate noisy labels and initialize the model parameters. The model setup follows that of [6, 22] with a batch size of 128, and SGD optimizer (0.9 momentum, 0.01 learning rate, 60 epochs). See footnote 1 for the complete implementation details.

Understanding Distributional Moments Penalization. Fig. 3a and 3b show the improvement in accuracy obtained for various values of α and ρ when training with *Distributional Moments Penalization*. From the figures, we can see that as we increase ρ, we increase the accuracy on both datasets up to a certain point, after which it starts decreasing. Note that the peak increase in accuracy is achieved for the same value of ρ for both datasets allowing us to use the validation dataset for hyper-parameter selection. Moreover, for the same dataset but different contamination levels, different α values achieve the

[1] github.com/valeriu-balaban/robust-learning-optimizing-tails-of-loss-distribution

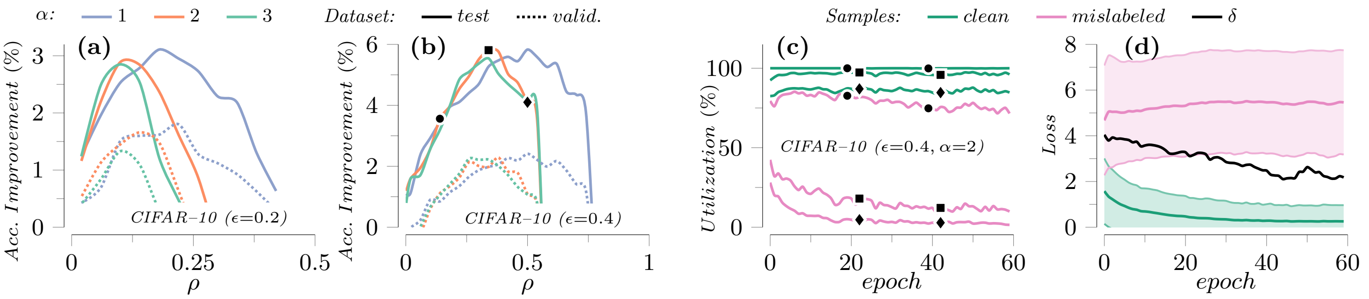

Figure 3: Understanding *Distributional Moments Penalization*. Subplots (a) and (b) show improvement in accuracy on test and validation datasets as a function of ρ for various α, provided for (a) 20% contamination, and (b) 40% contamination. (c) Percentage of samples with non-zero probability, for clean and mislabeled data, as a function of training epochs shown for the three scenarios marked in subplot (b). (d) Average loss for clean and mislabeled data as a function of training epochs. Filled areas display the standard deviation, and the black line indicates the slow-changing value of δ used for computing $\mathbb{Q}_n$.

maximum increase in accuracy, as contamination level affects the tails of the loss distribution. Fig. 3c shows the percentage of samples with non-zero probability for the three scenarios marked in Fig. 3b. As ρ increases, more mislabeled samples receive zero weight and are excluded from training. However, increasing ρ also leads to more clean samples receiving a zero probability, and as this number increases, the improvement in accuracy starts to drop. Recall that samples receive zero weight if their loss value is above δ, see Theorem 1. Fig. 3d shows that proper selection of ρ leads to a δ value that assigns zero weight to the majority of the mislabeled samples.

Experimental Results. We compare *Distributional Moments Penalization* against the following state-of-the-art (SOTA) methods: *(i) Generalized* crossentropy [5] with q=0.7, *(ii) TERM* [36] with t equal to -0.5, -0.7 and -0.9 for 20%, 40%, and 60% contamination, and *(iii) CIW* [20] with δ=1.5 and α=1. For *Distributional Moments Penalization* we use ρ=$\{0.3, 0.5, 0.8\}$ for α=1, and ρ=$\{0.1, 0.3, 0.45\}$ for α=2 and α=3, for contamination of 20%, 40%, and 60%, respectively. The above parameters were found using hyper-parameter tuning on the validation dataset. As a baseline, we use ERM with classical crossentropy loss.

Table I presents the test accuracy for all analyzed methods, and in general, all performed well given a properly selected set of hyper-parameters. Markedly, the *Distributional Moments Penalization* ranked first or second in all scenarios given a properly selected value of α. Nevertheless, all values of α yielded results close to each other.

To evaluate the performance of the method when it includes a distributionally robust optimization (DRO) stage, we use the noisy version of CIFAR–10 (ϵ=0.2) and form a *rare group* comprised of 5 classes, $y \in \{0, 1, 3, 5, 7\}$. During training, to generate underrepresented minority subsets, we use a small fraction of rare group samples (4%, 6%, and 8%). For *Distributional Moments Penalization* we use ρ=0.1 for α=2, and for the DRO version we use ρ=0.1 and α=2 for neighborhood $\mathcal{A}$ (Theorem 1) and ρ=0.2 and α=2 for neighborhood $\mathcal{B}$ (Equation (6)). Moreover, we start applying the DRO stage after 30 epochs to allow the model to learn the clean samples and reject the noisy ones. The hyper-parameters for the two methods were chosen using the same validation procedure as in Table I. Results are summarized in Table II, which show modest improvement for the DRO over the non-DRO version. These modest improvements highlight the challenge of solely using the loss value to distinguish noisy samples from clean but hard-to-learn ones, as both yield a high loss value.

		Noise contamination, ϵ		
	Method	0.2	0.4	0.6
Fashion–MNIST	Baseline	86.6 0.1	79.7 0.4	70.0 0.4
	Generalized	91.1 0.2	90.1 0.2	88.0 0.3
	TERM	91.5 0.2	90.3 0.1	88.3 0.2
	CIW	91.3 0.1	90.3 0.2	85.6 0.6
	Dist. Mom. Pen. (α=1)	90.6 0.1	90.4 0.1	88.4 0.1
	Dist. Mom. Pen. (α=2)	91.3 0.1	89.9 0.3	87.9 0.3
	Dist. Mom. Pen. (α=3)	91.4 0.1	90.4 0.1	86.8 0.2
CIFAR–10	Baseline	82.0 0.3	76.2 0.4	67.2 0.5
	Generalized	84.1 0.2	80.7 0.2	68.9 0.4
	TERM	84.7 0.2	81.6 0.2	75.7 0.4
	CIW	84.7 0.2	81.5 0.1	73.9 0.5
	Dist. Mom. Pen. (α=1)	84.7 0.1	81.8 0.2	75.5 0.5
	Dist. Mom. Pen. (α=2)	84.7 0.2	81.6 0.2	72.5 0.4
	Dist. Mom. Pen. (α=3)	84.9 0.3	80.7 1.2	66.5 0.9

Table I: Clean data test accuracy, mean and standard deviation.

	Rare group		
Method	4%	6%	8%
Baseline	40.6 0.2	48.1 0.4	55.1 0.3
Dist. Mom. Pen. (α=2)	41.3 0.1	48.8 0.2	55.6 0.4
Dist. Mom. Pen. DRO (α=2)	42.4 0.2	49.3 0.1	55.7 0.2

Table II: Test accuracy for clean and balanced class data.

V. Conclusion

We built a novel sample weighting strategy based on the premise that noisy samples have a higher loss value than clean ones since models learn clean data first. We showed theoretically that the proposed method simultaneously optimizes the mean and central moments of the loss distribution, downweighting samples on the right tail. In addition, we coupled our method with a distributionally robust optimization stage to equalize the loss values across subsets of training data.

Acknowledgment

The authors gratefully acknowledge the support by the National Science Foundation Career award under Grant No. Cyber-Physical Systems/CNS-1453860, the NSF award under Grant CCF-1837131, MCB-1936775, CNS-1932620, the U.S. Army Research Office (ARO) under Grant No. W911NF-23-1-0111, the Defense Advanced Research Projects Agency (DARPA) Young Faculty Award and DARPA Director Award under Grant No. N66001-17-1-4044, an Intel faculty award, the Okawa Foundation award, and a Northrop Grumman grant. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied by the DARPA, the ARO, the DoD, or the NSF.

References

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

[2] Lang Huang, Chao Zhang, and Hongyang Zhang. Self-adaptive training: beyond empirical risk minimization. *NeurIPS*, 2020.

[3] 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.

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[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] Hironori Fujisawa and Shinto Eguchi. Robust parameter estimation with a small bias against heavy contamination. *Journal of Multivariate Analysis*, 99(9):2053–2081, 2008.

[17] Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *AISTATS*, 2020.

[18] 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.

[19] Andrzej Cichocki and Shun-ichi Amari. Families of alpha-beta-and gamma-divergences: Flexible and robust measures of similarities. *Entropy*, 12(6):1532–1568, 2010.

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

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

[22] Valeriu Balaban, Hoda Bidkhori, and Paul Bogdan. Improving robustness: When and how to minimize or maximize the loss variance. In *ICMLA*. IEEE, 2022.

[23] Ayanendranath Basu, Ian R Harris, Nils L Hjort, and MC Jones. Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85(3):549–559, 1998.

[24] Minami Mihoko and Shinto Eguchi. Robust blind source separation by beta divergence. *Neural computation*, 2002.

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

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

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

[28] Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *ICML*, 2020.

[29] Zizhao Zhang, Han Zhang, Sercan O Arik, Honglak Lee, and Tomas Pfister. Distilling effective supervision from severe label noise. In *CVPR*, 2020.

[30] Baharan Mirzasoleiman, Kaidi Cao, and Jure Leskovec. Coresets for robust training of deep neural networks against noisy labels. In *NeurIPS*, 2020.

[31] Aritra Ghosh and Andrew Lan. Do we really need gold samples for sample weighting under label noise? In *WACV*, 2021.

[32] Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. Learning with feature-dependent label noise: A progressive approach. In *ICLR*, 2021.

[33] Pengxiang Wu, Songzhu Zheng, Mayank Goswami, Dimitris Metaxas, and Chao Chen. A topological filter for learning with label noise. In *NeurIPS*, 2020.

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

[35] Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. *NeurIPS*, 2020.

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