A CURRICULUM PERSPECTIVE TO ROBUST LOSS FUNCTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning with noisy labels is a fundamental problem in machine learning. Much work has been done in designing loss functions that are theoretically robust against label noise. However, it remains unclear why robust loss functions can underfit and why loss functions deviating from theoretical robustness conditions can appear robust. To elucidate these questions, we show that most robust loss functions differ only in the sample-weighting curriculums they implicitly define. The curriculum perspective enables straightforward analysis of the training dynamics with each loss function, which has not been considered in existing theoretical approaches. We show that underfitting can be attributed to marginal sample weights during training, and noise robustness can be attributed to larger weights for clean samples than noisy samples. With a simple fix to the curriculums, robust loss functions that severely underfit can become competitive with the state-of-the-art.[1]

1 INTRODUCTION

Labeling errors are non-negligible from automatic annotation (Liu et al., 2021; Khayrallah & Koehn, 2018), crowd-sourcing (Russakovsky et al., 2015) and expert annotation (Kato & Matsubara, 2010; Bridge et al., 2016). The resulting noisy labels may hamper generalization since over-parameterized neural networks can memorize the training set (Zhang et al., 2017). To combat the adverse impact of noisy labels, a large body of research (Song et al., 2020) aims to design loss functions robust against label noise. Existing approaches derive sufficient conditions (Ghosh et al., 2017; Zhou et al., 2021b) for noise robustness. Despite the theoretical appeal being agnostic to models and training dynamics, they may fail to comprehensively characterize the performance of robust loss functions. Specifically, it has been shown that (1) robust loss functions can underfit difficult tasks (Zhang & Sabuncu, 2018; Wang et al., 2019c; Ma et al., 2020; Song et al., 2020), while (2) loss functions violating existing robustness conditions (Zhang & Sabuncu, 2018; Wang et al., 2019c;b) can exhibit robustness. For (1), existing explanations (Ma et al., 2020; Wang et al., 2019a) can be limited as discussed in §2.2. For (2), to our knowledge, there has been no work directly addressing it.

We analyze training dynamics with various loss functions to elucidate the above observations, which complements existing theoretical approaches. Specifically, we rewrite a broad array of loss functions into a standard form with the same implicit loss function and varied sample-weighting functions (§3), each implicitly defining a sample-weighting curriculum. The interaction between the sample-weighting function and the distribution of implicit losses of samples thus reveals training dynamics with each loss function. Here a curriculum by definition (Wang et al., 2020) specifies a sequence of re-weighting for the distribution of training samples, e.g., sample weighting (Chang et al., 2017) or sample selection (Zhou et al., 2021a), based on a metric for sample difficulty. Notably, our novel curriculum perspective first connects robust loss functions to the seemingly distinct curriculum learning approaches (Song et al., 2020) for noise-robust training.

With our curriculum perspective, we first attribute the underfitting issue of robust loss functions to marginal sample weights during training (§4.1). In particular, for classification tasks with numerous classes, the initial sample weights under the curriculum of some robust loss functions can become marginal. However, when modifying the curriculums accordingly, robust loss functions that severely underfit can become competitive with the state-of-the-art. We then attribute noise robustness of loss functions to larger sample weights for clean samples than for noisy ones during training (§4.2). By

[1] Our code will be available at `github`.

examining the evolution of implicit losses during training, we find that dynamics of SGD suppress the learning of noisy samples. Curriculums of robust loss functions further suppress the learning of noisy samples by magnifying the difference in learning pace between clean and noisy samples while neglecting unlearned noisy samples. Based on our analysis, we present two unexpected results when viewed from existing theoretical results. By simply changing the learning rate schedule, (1) robust loss functions can become vulnerable to label noise, while (2) cross entropy can appear robust.

2 BACKGROUND

Classification k-ary classification with input $\boldsymbol{x} \in \mathbb{R}^d$ can be solved by classifier $\arg\max_i s_i$, where s_i is the score for the i-th class in the class scoring function $\boldsymbol{s} : \mathbb{R}^d \to \mathbb{R}^k$ parameterized by $\boldsymbol{\theta}$. Class scores $\boldsymbol{s}(\boldsymbol{x};\boldsymbol{\theta})$ can be turned into class probabilities with the softmax function $p_i = e^{s_i}/(\sum_{j=1}^k e^{s_j})$, where p_i is the probability of class i. Given a loss function $L(\boldsymbol{s}(\boldsymbol{x};\boldsymbol{\theta}), y)$ and data $(\boldsymbol{x}, y)$ with $y \in \{1..k\}$ the ground truth label, $\boldsymbol{\theta}$ can be estimated by risk minimization $\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},y} L(\boldsymbol{s}(\boldsymbol{x};\boldsymbol{\theta}), y)$, whose solutions are called risk minimizers. We use $\boldsymbol{s}$ in place of $\boldsymbol{s}(\boldsymbol{x};\boldsymbol{\theta})$ for notation simplicity.

Noise robustness Mistakes in the labeling process can lead to a potentially corrupted noisy label

$$\tilde{y} = \begin{cases} y, & \text{with probability } P(\tilde{y} = y|\boldsymbol{x}, y) \\ i,\ i \neq y & \text{with probability } P(\tilde{y} = i|\boldsymbol{x}, y) \end{cases}$$

Label noise is symmetric (or uniform) if $P(\tilde{y} = i|\boldsymbol{x}, y) = \eta/(k-1), \forall i \neq y$, with $\eta = P(\tilde{y} \neq y)$ the noise rate constant. Label noise is asymmetric (or class-conditional) if $P(\tilde{y} = i|\boldsymbol{x}, y) = P(\tilde{y} = i|y)$. Given data $(\boldsymbol{x}, \tilde{y})$ with noisy label $\tilde{y}$, a loss function L is robust against label noise if

$$\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},\tilde{y}} L(\boldsymbol{s}(\boldsymbol{x};\boldsymbol{\theta}), \tilde{y}) = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},y} L(\boldsymbol{s}(\boldsymbol{x};\boldsymbol{\theta}), y) \tag{1}$$

Conditions for noise robustness Existing approaches on robust loss function (Ghosh et al., 2017; Ma et al., 2020; Liu & Guo, 2020; Feng et al., 2020; Zhou et al., 2021b) focus on bounding the difference between risk minimizers obtained with noisy and clean data, i.e., ensuring that Eq. (1) approximately holds. The bounds only depend on the loss functions and mild assumptions about the dataset. To contrast our curriculum perspective with these approaches, we review two typical sufficient conditions for noise robustness. Loss function L is symmetric (Ghosh et al., 2017) if

$$\sum\nolimits_{i=1}^k L(\boldsymbol{s}, i) = C,\ \forall \boldsymbol{s} \in \mathbb{R}^k, \tag{2}$$

where C is a constant. It is robust against symmetric label noise with $\eta < (k-1)/k$. This stringent condition was later relaxed to the asymmetric condition. To rephrase, a loss function as a function of the softmax probability p_i, i.e., $L(\boldsymbol{s}, i) = l(p_i)$, is asymmetric (Zhou et al., 2021b) if

$$\max_{i \neq y} \frac{P(\tilde{y} = i|\boldsymbol{x}, y)}{P(\tilde{y} = y|\boldsymbol{x}, y)} = \tilde{r} \leq r = \inf_{\substack{0 \leq p_i, \Delta p \leq 1 \\ p_i + \Delta p \leq 1}} \frac{l(p_i) - l(p_i + \Delta p)}{l(0) - l(\Delta p)}, \tag{3}$$

where Δp is a valid increment of p_i. When clean labels dominate the data, i.e., $\tilde{r} < 1$, an asymmetric loss is robust against *generic* label noise.

The active-passive dichotomy Ma et al. (2020) draw a distinction between active and passive loss functions. By rewriting loss function L into a sum of basic functions, $L(\boldsymbol{s}, y) = \sum_{i=1}^k l(\boldsymbol{s}, i)$, an active loss function can be defined with the condition $\forall i \neq y,\ l(\boldsymbol{s}, i) = 0$, which emphasizes learning the target label. In contrast, a passive loss function defined with condition $\exists i \neq y,\ l(\boldsymbol{s}, i) \neq 0$ can be improved by unlearning the non-target labels.

In summary, research reviewed above is agnostic to the training dynamics, leading to open questions elaborated in §2.2. We address these issues with our curriculum perspective in §3 and §4.

2.1 TYPICAL ROBUST LOSS FUNCTIONS

In addition to cross entropy (CE) that is vulnerable to label noise (Ghosh et al., 2017), we review typical robust loss functions for analysis in §3 and §4. Differences in constant scaling factors and additive biases are ignored, as they are either equivalent to learning rate scaling in SGD or irrelevant in the gradient computation. See Table 1 for the exact formulas and Appendix A for an extended review with additional loss functions.

Type	Name	Function	Sample Weight w	Constraints
/	CE	$-\log p_y$	$1-p_y$	/
Sym.	MAE/RCE	$1-p_y$	$p_y(1-p_y)$	/
	NCE	$\frac{-\log p_y}{\sum_{i=1}^{k}-\log p_i}$	/	/
Asym.	AUL	$\frac{(a-p_y)^q-(a-1)^q}{q}$	$p_y(1-p_y)(a-p_y)^{q-1}$	$a>1, q>0$
	AGCE	$\frac{(a+1)-(a+p_y)^q}{q}$	$p_y(a+p_y)^{q-1}(1-p_y)$	$a>0, q>0$
Comb.	GCE	$\frac{1-p_y^q}{q}$	$p_y^q(1-p_y)$	$0<q\leq 1$
	SCE	$(1-q)\cdot L_{\mathrm{CE}}+q\cdot L_{\mathrm{MAE}}$	$(1-q+q\cdot p_y)(1-p_y)$	$0<q<1$
	NCE+MAE	$(1-q)\cdot L_{\mathrm{NCE}}+q\cdot L_{\mathrm{MAE}}$	/	$0<q<1$

Table 1: Expressions, constraints of hyperparameters and sample weights of the implicit curriculums (§3) for loss functions reviewed in §2.1.

Symmetric The mean absolute error (MAE; Ghosh et al., 2017) and the equivalent reverse cross entropy (RCE;Wang et al., 2019c) are both symmetric as they satisfy Eq. (2). Ma et al. (2020) make loss functions satisfying $L(\boldsymbol{s}, i) > 0, \forall i \in \{1..k\}$ symmetric by normalizing them with $L_{\mathrm{N}}(\boldsymbol{s}, y) = L(\boldsymbol{s}, y)/(\sum_{i=1}^{k} L(\boldsymbol{s}, i))$. We include normalized cross entropy (NCE; Ma et al. 2020) as an example.

Asymmetric We include asymmetric generalized cross entropy (AGCE) and asymmetric unhinged loss (AUL) proposed by Zhou et al. (2021b) as typical asymmetric loss functions. Notably, AGCE with $q \geq 1$ and AUL with $q \leq 1$ are completely asymmetric, i.e., Eq. (3) always holds when $\tilde{r} < 1$.

Combined Loss functions can be combined for both robustness and sufficient learning. For example, generalized cross entropy (GCE; Zhang & Sabuncu, 2018) is a smooth interpolation between CE and MAE. Alternatively, symmetric cross entropy (SCE; Wang et al., 2019c) is a weighted average of CE and RCE (MAE). Ma et al. (2020) argue that robust and sufficient training requires a balanced combination of active and passive loss functions. Accordingly to their active-passive dichotomy, CE and NCE are active while MAE (RCE) is passive. We include NCE+MAE as an example.

2.2 Open Questions

Why do robust loss functions underfit? Ma et al. (2020) attribute underfitting to failure in balancing active-passive components. However, different specifications of the base functions can lead to ambiguities in the active-passive dichotomy. For example, since

$$L_{\mathrm{MAE}}(\boldsymbol{s}, y) \propto \sum\nolimits_{i=1}^{k} |\mathbb{I}(i=y) - p_i| \propto \sum\nolimits_{i=1}^{k} \mathbb{I}(i=y)(1-p_i)$$

where $\mathbb{I}(\cdot)$ is the indicator function, MAE is passive with the first specification but active with the second. Wang et al. (2019a) view $\|\nabla_{\boldsymbol{s}} L(\boldsymbol{s}, y)\|_1$ as weights for sample gradients and attribute underfitting to their low variance, which makes clean and noisy samples less distinguishable. However, as we show in §4.1, MAE also underfits data with clean labels. In summary, neither Ma et al. (2020) nor Wang et al. (2019a) fully explain the underfitting issue.

What affects the robustness of a loss function? Although combined loss functions such as GCE and SCE fail to satisfy Eq. (2) and (3), it is unclear why they exhibit robustness against label noise (Zhang & Sabuncu, 2018; Wang et al., 2019c). Furthermore, it is unclear how training dynamics with loss functions, which are irrelevant in theoretical robustness guarantees (Ghosh et al., 2017; Zhou et al., 2021b; Ma et al., 2020; Liu & Guo, 2020; Feng et al., 2020), affect their noise robustness.

3 IMPLICIT CURRICULUMS OF LOSS FUNCTIONS

Most loss functions in Table 1 can be written as a function of the target softmax probability p_y, i.e., $L(\boldsymbol{s}, y) = l(p_y)$. A close examination of p_y gives

$$p_y = \frac{e^{s_y}}{\sum_{i=1}^{k} e^{s_i}} = \frac{1}{e^{\log \sum_{i \neq y} e^{s_i - s_y}} + 1} = \frac{1}{e^{-\Delta_y} + 1} = \text{sigmoid}(\Delta_y), \tag{4}$$

where

$$\Delta_y = s_y - \log \sum\nolimits_{i \neq y} e^{s_i} \leq s_y - \max\nolimits_{i \neq y} s_i \tag{5}$$

indicates how well a sample is learned, as $\Delta_y \geq 0$ ensures successful classification $y = \arg\max_i s_i$. Since $\nabla_{\boldsymbol{s}} l(p_y) = l'(p_y) \cdot p'_y(\Delta_y) \cdot \nabla_{\boldsymbol{s}} \Delta_y$, these loss functions can be rewritten into a standard form with equivalent gradients, i.e.,

$$L(\boldsymbol{s}, y) = -w(\Delta_y) \cdot \Delta_y, \tag{6}$$

where $w(\Delta_y) = l'(p_y) \cdot p'_y(\Delta_y)$ is a scalar weight wrapped with the stop-gradient operator and Δ_y is the implicit loss function embedded in $L(\boldsymbol{s}, y)$. Loss functions in the form of Eq. (6) thus implicitly define different sample-weighting curriculums, with $w(\Delta_y)$ the sample-weighting function and Δ_y the metric for sample difficulty. The implicit loss function Δ_y factors out the preference of $w(\Delta_y)$, making it a more direct metric for sample difficulty compared to those based on losses (Kumar et al., 2010) or gradient magnitudes (Gopal, 2016). In addition, the interaction between $w(\Delta_y)$ and distributions of Δ_y reveals the training dynamic with each loss function. Table 1 summarizes $w(\Delta_y)$ for the reviewed loss functions. See Appendix A for how hyperparameters affect $w(\Delta_y)$.

3.1 THE ADDITIONAL REGULARIZER OF NCE

NCE does not follow Eq. (6) as it additionally depends on Δ_i with $i \neq y$. Since

$$\begin{aligned}\nabla_{\boldsymbol{s}} L_{\text{NCE}}(\boldsymbol{s}, y) &= \frac{1}{\sum_{i=1}^{k} L_{\text{CE}}(\boldsymbol{s}, i)} \left\{ \nabla_{\boldsymbol{s}} L_{\text{CE}}(\boldsymbol{s}, y) + \frac{k L_{\text{CE}}(\boldsymbol{s}, y)}{\sum_{i=1}^{k} L_{\text{CE}}(\boldsymbol{s}, i)} \cdot \nabla_{\boldsymbol{s}} \left[-\frac{1}{k} \sum_{i=1}^{k} L_{\text{CE}}(\boldsymbol{s}, i) \right] \right\} \\ &= \gamma_{\text{NCE}} \cdot \left[\nabla_{\boldsymbol{s}} L_{\text{CE}}(\boldsymbol{s}, y) + \epsilon_{\text{NCE}} \cdot \nabla_{\boldsymbol{s}} R_{\text{NCE}}(\boldsymbol{s}) \right],\end{aligned}$$

it can be rewritten as

$$L_{\text{NCE}}(\boldsymbol{s}, y) = \gamma_{\text{NCE}} \cdot L_{\text{CE}}(\boldsymbol{s}, y) + \gamma_{\text{NCE}} \cdot \epsilon_{\text{NCE}} \cdot R_{\text{NCE}}(\boldsymbol{s}), \tag{7}$$

where $\gamma_{\text{NCE}} = 1/(\sum_{i=1}^{k} - \log p_i) \leq 1/(k \log k)$ and $\epsilon_{\text{NCE}} = k(-\log p_y)/(\sum_{i=1}^{k} - \log p_i)$ are scalar weights wrapped with the stop-gradient operator. The first term in Eq. (7) is a primary loss function inducing a curriculum similar to that of CE but with an additional weight γ_{NCE}. The second term is a regularizer

$$R_{\text{NCE}}(\boldsymbol{s}) = -\frac{1}{k} \sum_{i=1}^{k} L_{\text{CE}}(\boldsymbol{s}, i)$$

reducing the entropy of the softmax output. Although the training dynamics of NCE is complicated by $R_{\text{NCE}}(\boldsymbol{s})$, we can use the upperbound for the L1 norm of gradient $\nabla_{\boldsymbol{s}} L_{\text{NCE}}(\boldsymbol{s}, y)$,

$$\hat{w}_{\text{NCE}} = 2\gamma_{\text{NCE}} \left(w_{\text{CE}} + \frac{k-1}{k} \epsilon_{\text{NCE}} \right) \geq \|\nabla_{\boldsymbol{s}} L_{\text{NCE}}(\boldsymbol{s}, y)\|_1, \tag{8}$$

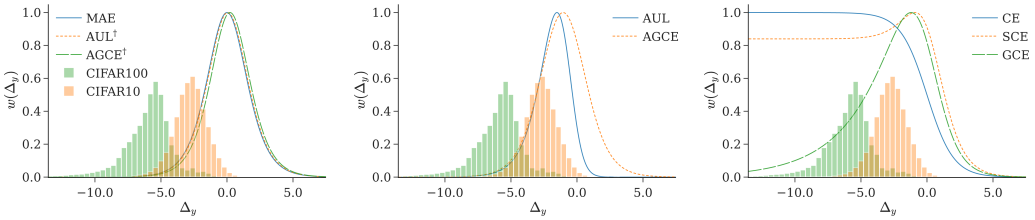
as the weight of each sample in gradient updates, which helps analyze the underfitting issue in §4.1. See Appendix A.2 for detailed derivations. We discuss more loss functions with extra regularizers, such as peer loss (Liu & Guo, 2020) and mean square error (Ghosh et al., 2017), in Appendix A.5.

4 ROBUST LOSS FUNCTIONS FROM THE CURRICULUM PERSPECTIVE

We examine the interaction between $w(\Delta_y)$ and the Δ_y distribution to address the questions in §2.2. Results are reported on MNIST (Lecun et al., 1998) and CIFAR10/100 (Krizhevsky et al., 2009) with synthetic symmetric and asymmetric label noises following standard settings (Ma et al., 2020; Zhou et al., 2021b). For real world settings we also include results on CIFAR10/100 with human noisy labels (Wei et al., 2022b) and the large scale noisy dataset WebVision (Li et al., 2017). Different from standard settings, we scale $w(\Delta_y)$ to unit maximum to avoid complications, since hyperparameters of loss functions can change the maximum of $w(\Delta_y)$, essentially adjusting the learning rate of SGD. See Appendix B for more experimental details.

Underfitting	Loss	CIFAR100 Acc.	CIFAR100 $\bar{\alpha}_t^*$	CIFAR10 Acc.	CIFAR10 $\bar{\alpha}_t^*$
/	CE	71.33 ± 0.23	8.183	92.76 ± 0.30	5.541
No	SCE	71.36 ± 0.39	9.541	93.17 ± 0.06	7.018
	GCE	69.95 ± 0.40	8.861	92.96 ± 0.13	6.151
	NCE+MAE	68.89 ± 0.23	/	92.37 ± 0.33	/
Moderate	AUL	58.75 ± 1.07	5.278	92.43 ± 0.19	5.171
	AGCE	49.27 ± 1.03	4.537	92.61 ± 0.18	5.225
	NCE	43.18 ± 1.55	/	91.28 ± 0.22	/
Severe	MAE	3.69 ± 0.59	0.035	91.56 ± 0.11	2.492
	AUL†	3.13 ± 0.43	0.033	91.13 ± 0.06	2.308
	AGCE†	1.62 ± 0.69	0.009	87.14 ± 4.96	1.701

Table 2: Without label noise, robust loss functions can underfit CIFAR100 but CIFAR10. Hyperparameters of loss functions are tuned on CIFAR100 and listed in Table 8. We report test accuracy and $\bar{\alpha}_t^*$ (scaled by 10^3) at the final training step from 3 different runs with learning rate $\alpha = 0.1$. Loss functions with inferior hyperparamters (denoted with †) are included as reference. See Table 9 for similar results with learning rate $\alpha = 0.01$.



(a) Severe underfitting (b) Moderate underfitting (c) No underfitting

Figure 1: Sample-weighting functions $w(\Delta_y)$ of loss functions in Table 2 with hyperparameters in Table 8. We include the initial Δ_y distributions of CIFAR10 and CIFAR100 for reference.

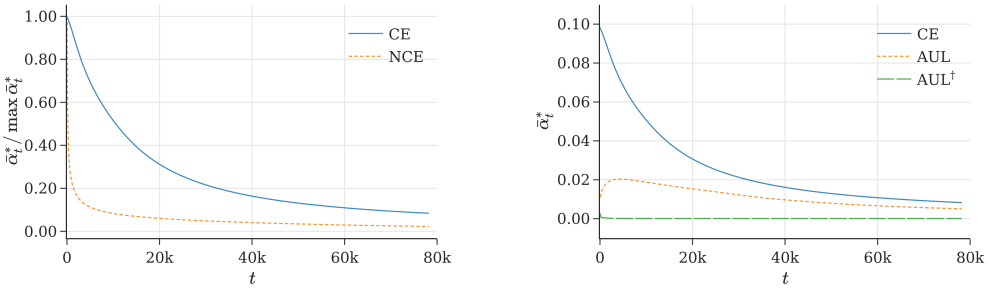
4.1 Understanding Underfitting of Robust Loss Functions

Robust loss functions can underfit We confirm that on difficult tasks like CIFAR100 (Song et al., 2020), underfiting can result from robust loss functions themselves rather than inferior hyperparameters. As shown in Table 2, CE outperforms NCE, AGCE and AUL by a nontrivial margin on CIFAR100. In particular, MAE performs much worse than CE, similar to AGCE† and AUL† with inferior hyperparameters. See Table 9 for similar results with learning rate $\alpha = 0.01$.

Marginal sample weights explains underfitting The effective parameter update size at step t can be approximated by $\alpha_t^* = \alpha_t \cdot \bar{w}_t$, which depends on the average sample weight $\bar{w}_t$ of the batch and the learning rate α_t. We use $\bar{\alpha}_t^* = \sum_{i=1}^t \alpha_i^*/t$ to characterize the overall α_t^* up to step t. In Table 2, for loss functions that heavily underfit on CIFAR100, $\bar{\alpha}_t^*$ at the final step is marginal compare to CE, suggesting a marginal overall sample weight during training given the same learning rate schedule.

Underfitting from fast diminishing sample weights Similar to CE but distinct from other loss functions, in Fig. 2a, $\bar{\alpha}_t^*$ of NCE based on $\hat{w}_{\text{NCE}}$ peaks at initialization. However, it decreases much faster compared to CE since both γ_{NCE} and w_{CE} decrease with improved Δ_y. In addition, the regularizer $R_{\text{NCE}}(\boldsymbol{s})$ further decreases the entropy of softmax output and thus γ_{NCE}. The resulting fast decreasing $\hat{w}_{\text{NCE}}$ quickly prevents the improvement of Δ_y, which can lead to underfitting.

Underfitting from marginal initial sample weights In Fig. 1a, loss functions that severely underfit in Table 2 assign marginal weights to samples in CIFAR100 at initialization, which leads to marginal initial $\bar{\alpha}_t^*$. Δ_y of these samples can barely improve before the learning rate vanishes, thus leading to underfitting. In contrast, loss functions with non-trivial initial weights for these samples (Fig. 1b and 1c) result in moderate or no underfitting. For further corroboration, we plot $\bar{\alpha}_t^*$ of AUL with superior and inferior hyperparameters (AUL and AUL† in Table 2) in Fig. 2b. $\bar{\alpha}_t^*$ stays marginal with AUL†, while with AUL, $\bar{\alpha}_t^*$ quickly increases to a non-negligible value before gradual decreasing.



(a) NCE with estimated weight upperbound.

(b) AUL with inferior/superior hyperparameters.

Figure 2: Different explanations for underfitting: (a) fast diminishing sample weights; (b) marginal initial sample weights. We plot variation of $\bar{\alpha}_t^*$ across training steps t on CIFAR100 without label noise for each loss function. $\bar{\alpha}_t^*$ of NCE are estimated with $\hat{w}_{\mathrm{NCE}}$. Since $\hat{w}_{\mathrm{NCE}}$ is not comparable to w_{CE}, we normalize $\bar{\alpha}_t^*$ with its maximum in (a) to emphasize its variation during training.

Loss	Clean $\eta = 0$	Symmetric $\eta = 0.4$	Symmetric $\eta = 0.8$	Asymmetric $\eta = 0.4$	Human $\eta = 0.4$
CE‡	71.33 ± 0.43	39.92 ± 0.10	7.59 ± 0.20	40.17 ± 1.31	/
GCE‡	63.09 ± 1.39	56.11 ± 1.35	17.42 ± 0.06	40.91 ± 0.57	/
NCE‡	29.96 ± 0.73	19.54 ± 0.52	8.55 ± 0.37	20.64 ± 0.40	/
NCE+AUL‡	68.96 ± 0.16	59.25 ± 0.23	23.03 ± 0.64	38.59 ± 0.48	/
AGCE	49.27 ± 1.03	47.76 ± 1.75	16.03 ± 0.59	33.40 ± 1.57	30.45 ± 1.50
AGCE shift	69.39 ± 0.84	48.21 ± 1.06	14.49 ± 0.17	40.76 ± 0.74	48.71 ± 0.45
AGCE scale	70.57 ± 0.62	56.69 ± 0.33	14.64 ± 0.79	39.71 ± 0.17	50.85 ± 0.11
MAE	3.69 ± 0.59	1.29 ± 0.50	1.00 ± 0.00	2.53 ± 1.34	2.09 ± 0.55
MAE shift	68.57 ± 0.54	49.95 ± 0.16	13.10 ± 0.41	39.83 ± 0.18	47.91 ± 0.36
MAE scale	**70.97 ± 0.41**	**60.57 ± 1.04**	**24.44 ± 0.73**	**44.48 ± 1.05**	**54.70 ± 0.48**

Table 3: Shifting or scaling $w(\Delta_y)$ mitigates underfitting on CIFAR100 under different label noise. We report test accuracies with 3 different runs. See Table 11 for hyperparameter τ of each setting. Results from Zhou et al. (2021b) are included as context (denoted with ‡). We also report results with more noise rates in Tables 12 and 13.

Increased number of classes leads to marginal initial sample weights. Unlike CIFAR100, all loss functions in Table 2 perform equally well on CIFAR10. Such a difference has been vaguely attributed to the increased task difficulty of CIFAR100 (Zhang & Sabuncu, 2018; Song et al., 2020). Intuitively, the more classes, the more subtile differences to be distinguished. In addition, the number of classes k determines the initial distribution of Δ_y. Assume that class scores s_i at *initialization* are i.i.d. normal variables $s_i \sim \mathcal{N}(\mu, \sigma)$. In particular, $\mu = 0$ and $\sigma = 1$ for most neural networks with standard initializations (Glorot & Bengio, 2010; He et al., 2015) and normalization layers (Ioffe & Szegedy, 2015; Ba et al., 2016). The expected Δ_y can be approximated with

$$\mathbb{E}[\Delta_y] \approx -\log(k-1) - \sigma^2/2 + \frac{e^{\sigma^2} - 1}{2(k-1)} \tag{9}$$

We leave derivations and comparison between our assumptions and real settings to Appendix C.1. A large k results in small Δ_y at initialization, leading to marginal sample weights with the fixed sample-weighting curriculums of robust loss functions that underfit CIFAR100.

4.1.1 Addressing Underfitting from Marginal Initial Sample Weights

Our analysis suggests that the fixed sample-weighting function $w(\Delta_y)$ is to blame for underfitting. To make the initial sample weights agnostic to the number of classes, we can either scale

$$w^*(\Delta_y) = w(\Delta_y / \mathbb{E}[\Delta_y] \cdot \tau)$$

or shift

$$w^+(\Delta_y) = w(\Delta_y + \mathbb{E}[\Delta_y] - \tau)$$

Settings	$k = 50$ $\tau = 2.0$	$k = 200$ $\tau = 1.8$	$k = 400$ $\tau = 1.6$
CE	66.40	70.26	70.16
MAE	3.68	0.50	0.25
MAE shift	60.76	59.31	47.32
MAE scale	**66.72**	**71.92**	**71.87**

Table 4: Shifting or scaling $w(\Delta_y)$ mitigates underfitting on WebVision subsampled with different number of classes. $k = 50$ is the standard mini setting in previous work (Ma et al., 2020; Zhou et al., 2021b). We report test accuracy with a single run due to limited computation budget.

	Clean	Asymmetric		Symmetric						Human	
		$\eta = 0.2$		$\eta = 0.2$		$\eta = 0.4$		$\eta = 0.8$		$\eta = 0.4$	
Loss	Acc	diff	snr	diff	snr	diff	snr	diff	snr	diff	snr
CE	90.64	-7.06	0.32	-15.47	0.39	-31.95	0.57	-50.87	0.77	-28.51	0.53
SCE	89.87	-5.39	0.51	-3.84	0.99	-10.47	1.27	-27.25	1.51	-15.84	0.86
GCE	90.44	-7.42	0.36	-6.80	0.96	-23.23	0.89	-45.32	1.04	-21.94	0.72
AUL	89.90	-2.51	0.81	-2.07	3.10	-5.87	2.96	-13.90	2.83	-12.08	1.11
MAE	89.29	-2.21	1.00	-1.92	3.56	-4.36	3.33	-11.53	3.22	-10.35	1.32
AGCE	82.62	-9.42	0.92	-1.55	3.02	-19.90	2.16	-41.11	1.83	-21.73	1.28

Table 5: Robust loss functions assign larger weights to clean samples. We report snr and diff with the best of 5 different runs on CIFAR10 under different label noise since worse initialization can lead to a substantial drop in performance. Hyperparameters of loss functions are chosen to cover more variants of sample weighting functions, which are not necessarily optimal for each setting. See Table 10 for hyperparameters and Fig. 7 for the sample weighting curves.

the sample-weighting functions, where $\tau > 0$ is a hyperparameter. Intuitively, $\mathbb{E}[\Delta_y]$ in $w^*(\Delta_y)$ and $w^+(\Delta_y)$ cancels the effect of k on the initial sample weights. A small τ thus leads to high initial sample weights regardless of the number of classes k.

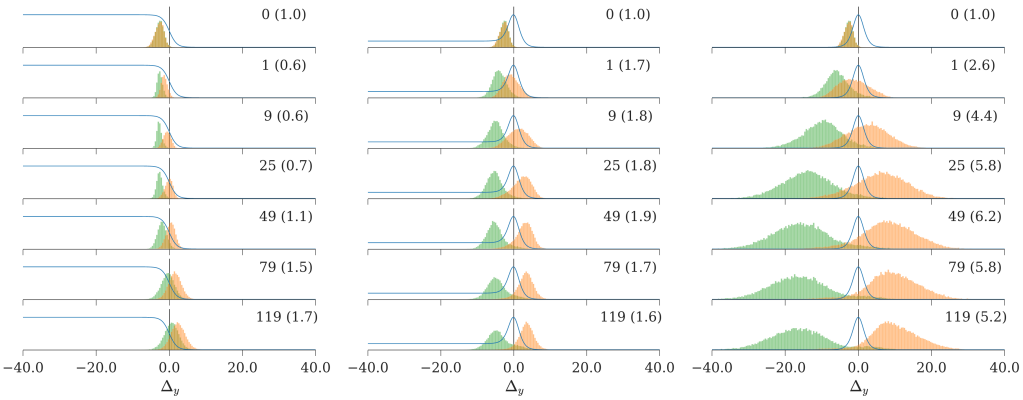
Results on CIFAR100 with different label noise are reported in Table 3. See Tables 12 and 13 for results with additional noise rates. We also report results on the large scale WebVision dataset with different number of classes in Table 4. In summary, shifting and scaling alleviate underfitting, making MAE and AGCE comparable to the previous state-of-the-art (NCE+AUL; Zhou et al. 2021b). Notably, $w^*(\Delta_y)$ leads to a dramatic improvements for MAE under all settings.

4.2 Understanding Noise Robustness of Loss Functions

For robustness against label noise, research on sample-weighting curriculums (Jiang et al., 2018; Ren et al., 2018) *explicitly* identifies clean samples and assign them with larger weights. In contrast, we show that robust loss functions in the form of Eq. (6) *implicitly* achieve similar sample weights. The underlying reasons are explored by examining how distributions of Δ_y evolve during training.

Robust loss functions assign larger weights to clean samples. We use the ratio between average weights of clean ($\bar{w}_{\text{clean}}$) and noisy ($\bar{w}_{\text{noise}}$) samples, $\mathrm{snr} = \bar{w}_{\text{clean}}/\bar{w}_{\text{noise}}$, to characterize their relative contribution during training. See Appendix C.2 for the exact formulas. Robustness is characterized by difference in test accuracy compared to results with clean labels (diff). We report diff and snr under different label noise on CIFAR10 in Table 5. Loss functions with higher snr have less performance drop with label noise in general, thus being more robust.

To explain what leads to a large snr, we plot the evolution of Δ_y distributions during training on CIFAR10 with symmetric label noise in Fig. 3, which reveals the training dynamics with different loss functions. See Fig. 8 and 9 for results with other types of label noise and loss functions. When trained with more robust loss functions (Fig. 3b and 3c), Δ_y distributions of noisy and clean samples spread wider and get better separated. In addition, the consistent decrease of Δ_y for noisy samples suggests that they can be *unlearned*. In contrast, training with CE (Fig. 3a) results in more compact and less separated Δ_y distributions. Furthermore, Δ_y of noisy samples consistently increase.



(a) CE: 57.74 (b) SCE: 79.56 (c) MAE: 83.88

Figure 3: How Δ_y distribution of noisy (green, left) and clean (orange, right) samples evolve during training on CIFAR10 under symmetric label noise with $\eta = 0.4$. Vertical axes denoting probability density are scaled to the peak of histograms for better readability, with epoch number (axis scaling factor) denoted on the right of each subplot. We plot $w(\Delta_y)$ and report the test accuracy of each setting for reference. See Appendix C.2 for results with different label noise and more loss functions.

Dynamics of SGD suppress learning of noisy samples. As shown in Fig. 3a, noisy samples are learned slower than clean samples as measured by improvements of Δ_y, which can be explained by more coherent gradients among clean samples (Chatterjee & Zielinski, 2022). Similar results have been reported by Zhang et al. (2017); Arpit et al. (2017) and utilized in curriculum-based robust training (Yao et al., 2019; Han et al., 2018). In addition, noisy samples can be unlearned as shown in Fig. 3b and 3c, which can stem from generalization with clean samples. Both dynamics suppress the learning of noisy samples but clean ones, thus leading to robustness against label noise.

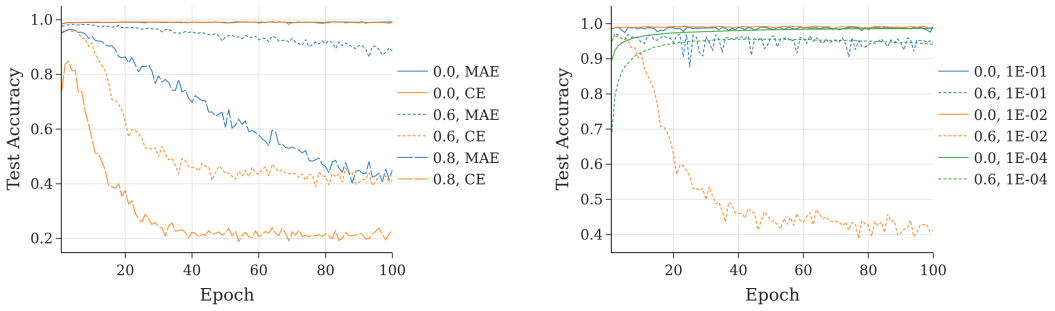
Robust $w(\Delta_y)$ synergizes with SGD dynamics for noise robustness. In Fig. 1, the bell-shaped $w(\Delta_y)$ of robust loss functions only assigns large weights to samples with moderate Δ_y. Since Δ_y distributions initially concentrate at the monotonically increasing interval of $w(\Delta_y)$, (1) samples with faster improving Δ_y, due to either larger initial weights or faster learning as clean samples, are weighted more during early training and learned faster. The magnified learning pace difference explains the widely spread distributions in Fig. 3b and 3c. In addition, (2) the unlearned samples with small Δ_y receive diminishing weights from $w(\Delta_y)$, which hampers their pace of learning. Noisy samples in Fig. 3b and 3c are consistently unlearned and ignored with marginal sample weights, leading to a consistent decrease of Δ_y. In addition to the SGD dynamics, (1) and (2) further suppress the learning of noisy samples and enhance that of clean samples, thus leading to increased robustness against label noise. In contrast, the monotonically decreasing $w_{\mathrm{CE}}(\Delta_y)$ emphasizes samples with smaller Δ_y, essentially acting against the SGD dynamics for noise robustness. Thus training with CE results in increased vulnerability to label noise as shown in Table 5.

4.2.1 Training Schedules Affect Noise Robustness

Although the learning pace of noisy samples get suppressed, the expected gradient will eventually be dominated by noisy samples since well learned clean samples receive marginal sample weights thanks to the monotonically decreasing interval of $w(\Delta_y)$. Models with extended training can thus risk overfitting noisy samples. By adjusting the training schedules, we can modify the learning pace of samples and thus affect the noise robustness of models.

Robust loss functions are vulnerable to label noise with extended training. In Fig. 4a we show the learning curve of CE and MAE using *constant* learning rate under different symmetric noises on MNIST. Despite the theoretically guaranteed noise robustness (Ghosh et al., 2017), similar to CE, MAE eventually overfits to noisy samples, resulting in vulnerability to label noise.

CE can become robust by adjusting the learning rate schedule. Despite the compensation of $w_{\mathrm{CE}}(\Delta_y)$, the SGD dynamics still result in partially-separated Δ_y distributions of noisy and clean samples. We can improve the noise robustness of CE by preventing weights of clean samples from diminishing, which can be achieve by slowing down the learning pace or early stopping (Song et al.,



(a) $\alpha = 0.01$ with different (η, loss).

(b) CE with different (η, α).

Figure 4: Learning curves with fixed learning rate and extended training epochs on MNIST with symmetric label noise, where α is the learning rate and η the noise rate.

2019). In Fig. 4b we show the learning curve of CE using fixed learning rates under symmetric label noise on MNIST. By simply increasing or decreasing the learning rate, which strengthens the implicit regularization of SGD (Smith et al., 2021) or directly slows down the convergence, the noise robustness of CE can be substantially improved.

5 Related Work

Our work is closely related to robust loss functions against label noise (Song et al., 2020). Existing research (Ghosh et al., 2017; Zhang & Sabuncu, 2018; Wang et al., 2019c; Ma et al., 2020; Liu & Guo, 2020; Cheng et al., 2020; Feng et al., 2020; Zhou et al., 2021b) focuses on bounding the difference between risk minimizers obtained using noisy and clean data without considering the training dynamics. In contrast, with our novel curriculum perspective, we analyze the training dynamics with robust loss functions for reasons behind their underfitting issue and noise robustness. The underfitting problem has been heuristically mitigated with loss combination (Zhang & Sabuncu, 2018; Wang et al., 2019c; Ma et al., 2020). We identify the cause and provide an effective solution.

Curriculum-based approaches combat label noise with either sample selection (Chen et al., 2019; Zhou et al., 2021a) or sample-weighting (Chang et al., 2017; Jiang et al., 2018; Ren et al., 2018). In particular, sample weights are *explicitly* designed (Wang et al., 2019a;b; Chang et al., 2017) or predicted by a model trained on a separated dataset (Jiang et al., 2018; Ren et al., 2018). In contrast, the sample-weighting curriculums considered in this work are *implicitly* defined by robust loss functions. Most related to our work, Wang et al. (2019b) identify gradient norms as weights for sample gradients and propose heuristic designs of weighting functions for noise robust training. In contrast, we explicitly identify the implicit loss function, which connects robust loss functions to curriculum learning, facilitates our analysis on the training dynamics and helps elucidate the robustness of loss functions from a curriculum perspective.

Our work is also related to the ongoing debate (Hacohen & Weinshall, 2019; Wang et al., 2020) on strategies for selecting or weighting samples in curriculum learning: either easier first (Bengio et al., 2009; Kumar et al., 2010) or harder first (Loshchilov & Hutter, 2015; Zhang et al., 2018). The implicit curriculums of robust loss functions can be viewed as a combination of both strategies, which emphasize samples with moderate difficulty. In addition, the implicit loss function we identified is a more direct metric for sample difficulty compared to loss functions (Kumar et al., 2010; Loshchilov & Hutter, 2015) and gradient magnitudes (Gopal, 2016).

6 Conclusion

We identified the implicit sample-weighting curriculums of a broad array of loss functions. Our novel curriculum perspective enables examining the training dynamics with loss functions through the interaction between the sample-weighting function and the distribution of implicit losses. It connects robust loss functions to the seemingly distinct curriculum learning. Notably, the implicit loss function we identified is a direct metric for sample difficulty in curriculum learning as it factors out the preference of sample-weighting functions. We elucidate the reasons behind underfitting and robustness against label noise and propose a simple approach to address the underfitting issue.

REFERENCES

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 233–242. PMLR, 2017. URL http://proceedings.mlr.press/v70/arpit17a.html.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman (eds.), *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pp. 41–48. ACM, 2009. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

Pete Bridge, Andrew Fielding, Pamela Rowntree, and Andrew Pullar. Intraobserver variability: should we worry? *Journal of medical imaging and radiation sciences*, 47(3):217–220, 2016.

Haw-Shiuan Chang, Erik G. Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 1002–1012, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/2f37d10131f2a483a8dd005b3d14b0d9-Abstract.html.

Satrajit Chatterjee and Piotr Zielinski. On the generalization mystery in deep learning. *ArXiv preprint*, abs/2203.10036, 2022. URL https://arxiv.org/abs/2203.10036.

Pengfei Chen, Benben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1062–1070. PMLR, 2019. URL http://proceedings.mlr.press/v97/chen19g.html.

Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach, 2020. URL https://arxiv.org/abs/2010.02347.

Barry Cobb, Rafael Rumí, and Antonio Salmerón. Approximating the distribution of a sum of log-normal random variables. 2012.

Lei Feng, Senlin Shu, Zhuoyi Lin, Fengmao Lv, Li Li, and Bo An. Can cross entropy loss be robust to label noise? In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pp. 2206–2212, Yokohama, Japan, Jul 2020. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-6-5. doi: 10.24963/ijcai.2020/305. URL https://www.ijcai.org/proceedings/2020/305.

Aritra Ghosh, Himanshu Kumar, and P. S. Sastry. Robust loss functions under label noise for deep neural networks. In Satinder P. Singh and Shaul Markovitch (eds.), *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pp. 1919–1925. AAAI Press, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14759.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010. URL https://proceedings.mlr.press/v9/glorot10a.html.

Siddharth Gopal. Adaptive sampling for SGD by exploiting side information. In Maria-Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 364–372. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/gopal16.html.

Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2535–2544. PMLR, 2019. URL http://proceedings.mlr.press/v97/hacohen19a.html.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8536–8546, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/a19744e268754fb0148b017647355b7b-Abstract.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pp. 1026–1034. IEEE Computer Society, 2015. doi: 10.1109/ICCV.2015.123. URL https://doi.org/10.1109/ICCV.2015.123.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis R. Bach and David M. Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 448–456. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/ioffe15.html.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2309–2318. PMLR, 2018. URL http://proceedings.mlr.press/v80/jiang18c.html.

Yoshihide Kato and Shigeki Matsubara. Correcting errors in a treebank based on synchronous tree substitution grammar. In *Proceedings of the ACL 2010 Conference Short Papers*, pp. 74–79, Uppsala, Sweden, 2010. Association for Computational Linguistics. URL https://aclanthology.org/P10-2014.

Huda Khayrallah and Philipp Koehn. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pp. 74–83, Melbourne, Australia, 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-2709. URL https://aclanthology.org/W18-2709.

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

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta (eds.), *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting*

held 6-9 December 2010, Vancouver, British Columbia, Canada, pp. 1189–1197. Curran Associates, Inc., 2010. URL https://proceedings.neurips.cc/paper/2010/hash/e57c6b956a6521b28495f2886ca0977a-Abstract.html.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data, 2017. URL https://arxiv.org/abs/1708.02862.

Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 2999–3007. IEEE Computer Society, 2017. doi: 10.1109/ICCV.2017.324. URL https://doi.org/10.1109/ICCV.2017.324.

Kun Liu, Yao Fu, Chuanqi Tan, Mosha Chen, Ningyu Zhang, Songfang Huang, and Sheng Gao. Noisy-labeled NER with confidence estimation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3437–3445, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.269. URL https://aclanthology.org/2021.naacl-main.269.

Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. (arXiv:1910.03231), Aug 2020. doi: 10.48550/arXiv.1910.03231. URL http://arxiv.org/abs/1910.03231. arXiv:1910.03231 [cs, stat].

Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. *ArXiv preprint*, abs/1511.06343, 2015. URL https://arxiv.org/abs/1511.06343.

Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *Proceedings of the 37th International Conference on Machine Learning*, pp. 6448–6458. PMLR, Nov 2020. URL https://proceedings.mlr.press/v119/lukasik20a.html.

Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah M. Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6543–6553. PMLR, 2020. URL http://proceedings.mlr.press/v119/ma20c.html.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 2233–2241. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.240. URL https://doi.org/10.1109/CVPR.2017.240.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4331–4340. PMLR, 2018. URL http://proceedings.mlr.press/v80/ren18a.html.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *arXiv:1409.0575 [cs]*, 2015. URL http://arxiv.org/abs/1409.0575. arXiv: 1409.0575.

Samuel L. Smith, Benoit Dherin, David G. T. Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=rq_Qr0c1Hyo.

Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. How does early stopping help generalization against label noise?, 2019. URL https://arxiv.org/abs/1911.08059.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *ArXiv preprint*, abs/2007.08199, 2020. URL https://arxiv.org/abs/2007.08199.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015. URL https://arxiv.org/abs/1512.00567.

Yee Whye Teh, David Newman, and Max Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In Bernhard Schölkopf, John C. Platt, and Thomas Hofmann (eds.), *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pp. 1353–1360. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2006/hash/532b7cbe070a3579f424988a040752f2-Abstract.html.

Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *ArXiv preprint*, abs/2010.13166, 2020. URL https://arxiv.org/abs/2010.13166.

Xinshao Wang, Yang Hua, Elyor Kodirov, and Neil M. Robertson. Imae for noise-robust learning: Mean absolute error does not treat examples equally and gradient magnitude's variance matters. *ArXiv preprint*, abs/1903.12141, 2019a. URL https://arxiv.org/abs/1903.12141.

Xinshao Wang, Elyor Kodirov, Yang Hua, and Neil M. Robertson. Derivative manipulation for general example weighting. *ArXiv preprint*, abs/1905.11233, 2019b. URL https://arxiv.org/abs/1905.11233.

Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 322–330. IEEE, 2019c. doi: 10.1109/ICCV.2019.00041. URL https://doi.org/10.1109/ICCV.2019.00041.

Jiaheng Wei, Hangyu Liu, Tongliang Liu, Gang Niu, Masashi Sugiyama, and Yang Liu. To smooth or not? when label smoothing meets noisy labels. (arXiv:2106.04149), Jun 2022a. doi: 10.48550/arXiv.2106.04149. URL http://arxiv.org/abs/2106.04149. arXiv:2106.04149 [cs].

Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. Mar 2022b. URL https://openreview.net/forum?id=TBWA6PLJZQm.

Quanming Yao, Hansi Yang, Bo Han, Gang Niu, and James Kwok. Searching to exploit memorization effect in learning from corrupted labels, 2019. URL https://arxiv.org/abs/1911.02377.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J. Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat. An empirical exploration of curriculum learning for neural machine translation. *ArXiv preprint*, abs/1811.00739, 2018. URL https://arxiv.org/abs/1811.00739.

Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp.

8792–8802, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/f2925f97bc13ad2852a7a551802feea0-Abstract.html.

Tianyi Zhou, Shengjie Wang, and Jeff A. Bilmes. Robust curriculum learning: from clean label detection to noisy label self-correction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a. URL https://openreview.net/forum?id=lmTWnm3coJJ.

Xiong Zhou, Xianming Liu, Junjun Jiang, Xin Gao, and Xiangyang Ji. Asymmetric loss functions for learning with noisy labels. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12846–12856. PMLR, 2021b. URL http://proceedings.mlr.press/v139/zhou21f.html.

A Extended Review of Loss Functions

Due to limited space, we only briefly describe typical robust loss functions in §2.1. As a general reference, here we provide an extended review of loss functions related to the standard form Eq. (6). Similar to §2.1, we ignore the differences in constant scaling factors and additive bias. Loss functions and their sample-weighting functions are summarized in Table 6. We examine how hyperparameters affect their sample-weighting functions in Fig. 5.

A.1 Loss Functions without Robustness Guarantees

Cross Entropy (CE)

$$L_{\mathrm{CE}}(\boldsymbol{s}, y) = -\log p_y$$

is the standard loss function for classification.

Focal Loss (FL; Lin et al. 2017)

$$L_{\mathrm{FL}}(\boldsymbol{s}, y) = -(1 - p_y)^q \log p_y$$

aims to address label imbalance when training object detection models. Both CE and FL are neither symmetric (Ma et al., 2020) nor asymmetric (Zhou et al., 2021b).

A.2 Symmetric Loss Functions

Mean Absolute Error (MAE; Ghosh et al. 2017)

$$L_{\mathrm{MAE}}(\boldsymbol{s}, y) = \sum_{i=1}^{k} |\mathbb{I}(i = y) - p_i| = 2 - 2p_y \propto 1 - p_y$$

is a classic symmetric loss function, where $\mathbb{I}(i = y)$ is the indicator function.

Reverse Cross Entropy (RCE; Wang et al. 2019c)

$$L_{\mathrm{RCE}}(\boldsymbol{s}, y) = \sum_{i=1}^{k} p_i \log \mathbf{1}(i = y) = \sum_{i \neq y} p_i A = (1 - p_y)A \propto 1 - p_y = L_{\mathrm{MAE}}(\boldsymbol{s}, y)$$

is equivalent to MAE in implementation, where $\log 0$ is truncated to a negative constant A to avoid numerical overflow.

Ma et al. (2020) argued that any generic loss functions with $L(\boldsymbol{s}, i) > 0, \forall i \in \{1..k\}$ can become symmetric by simply normalizing them. As an example,

Normalized Cross Entropy (NCE, Ma et al. 2020)

$$L_{\mathrm{NCE}}(\boldsymbol{s}, y) = \frac{L_{\mathrm{CE}}(\boldsymbol{s}, y)}{\sum_{i=1}^{k} L_{\mathrm{CE}}(\boldsymbol{s}, i)} = \frac{-\log p_y}{\sum_{i=1}^{k} -\log p_i}$$

is a symmetric loss function. However, NCE does not follow the standard form of Eq. (6). It involves an additional regularizer as discussed in §3.1, thus being more relevant to discussions in Appendix A.5. Here we derive the upperbound of $\|\nabla_{\boldsymbol{s}} L_{\mathrm{NCE}}(\boldsymbol{s}, y)\|_1$ discussed in §3.1:

$$\begin{aligned}
\|\nabla_{\boldsymbol{s}} L_{\mathrm{NCE}}(\boldsymbol{s}, y)\|_1 &\leq \gamma_{\mathrm{NCE}} \cdot (\|\nabla_{\boldsymbol{s}} L_{\mathrm{CE}}(\boldsymbol{s}, y)\|_1 + \epsilon_{\mathrm{NCE}} \cdot \|\nabla_{\boldsymbol{s}} R_{\mathrm{NCE}}(\boldsymbol{s})\|_1) \\
&\leq \gamma_{\mathrm{NCE}} \cdot \left(\|\nabla_{\boldsymbol{s}} L_{\mathrm{CE}}(\boldsymbol{s}, y)\|_1 + \epsilon_{\mathrm{NCE}} \cdot \frac{1}{k} \sum_i \|\nabla_{\boldsymbol{s}} L_{\mathrm{CE}}(\boldsymbol{s}, i)\|_1 \right) \\
&= \gamma_{\mathrm{NCE}} \cdot \left(w_{\mathrm{CE}} \cdot \|\nabla_{\boldsymbol{s}} \Delta_y\|_1 + \epsilon_{\mathrm{NCE}} \cdot \frac{1}{k} \sum_i w_{\mathrm{CE}} \cdot \|\nabla_{\boldsymbol{s}} \Delta_i\|_1 \right) \\
&= 2\gamma_{\mathrm{NCE}} \cdot \left(w_{\mathrm{CE}} + \epsilon_{\mathrm{NCE}} \cdot \frac{k-1}{k} w_{\mathrm{CE}} \right)
\end{aligned}$$

The derivation is based on the inequality $|x \pm y| \leq |x| + |y|$.

A.3 Asymmetric Loss Functions

Zhou et al. (2021b) derived the asymmetric condition for noise robustness and propose numerous asymmetric loss functions:

Asymmetric Generalized Cross Entropy (AGCE)

$$L_{\mathrm{AGCE}}(\boldsymbol{s}, y) = \frac{(a+1) - (a+p_y)^q}{q}$$

where $a > 0$ and $q > 0$. It is asymmetric when $\mathbb{I}(q \leq 1)(\frac{a+1}{a})^{1-q} + \mathbb{I}(q > 1) \leq 1/\tilde{r}$.

Asymmetric Unhinged Loss (AUL)

$$L_{\mathrm{AUL}}(\boldsymbol{s}, y) = \frac{(a-p_y)^q - (a-1)^q}{q}$$

where $a > 1$ and $q > 0$. It is asymmetric when $\mathbb{I}(q \leq 1)(\frac{a}{a-1})^{q-1} + \mathbb{I}(q \leq 1) \leq 1/\tilde{r}$.

Asymmetric Exponential Loss (AEL)

$$L_{\mathrm{AEL}}(\boldsymbol{s}, y) = e^{-p_y/q}$$

where $q > 0$. It is assymetric when $e^{1/q} \leq 1/\tilde{r}$.

A.4 Combined Loss Functions

Generalized Cross Entropy (GCE; Zhang & Sabuncu 2018)

$$L_{\mathrm{GCE}}(\boldsymbol{s}, y) = \frac{1 - p_y^q}{q}$$

can be viewed as a smooth interpolation between CE and MAE, where $0 < q \leq 1$. CE or MAE can be recovered by setting $q \to 0$ or $q = 1$.

Symmetric Cross Entropy (SCE; Wang et al. 2019c)

$$L_{\mathrm{SCE}}(\boldsymbol{s}, y) = a \cdot L_{\mathrm{CE}}(\boldsymbol{s}, y) + b \cdot L_{\mathrm{RCE}}(\boldsymbol{s}, y) \propto (1-q) \cdot (-\log p_i) + q \cdot (1 - p_i)$$

is a weighted average of CE and RCE (MAE), where $a > 0$, $b > 0$, and $0 < q < 1$.

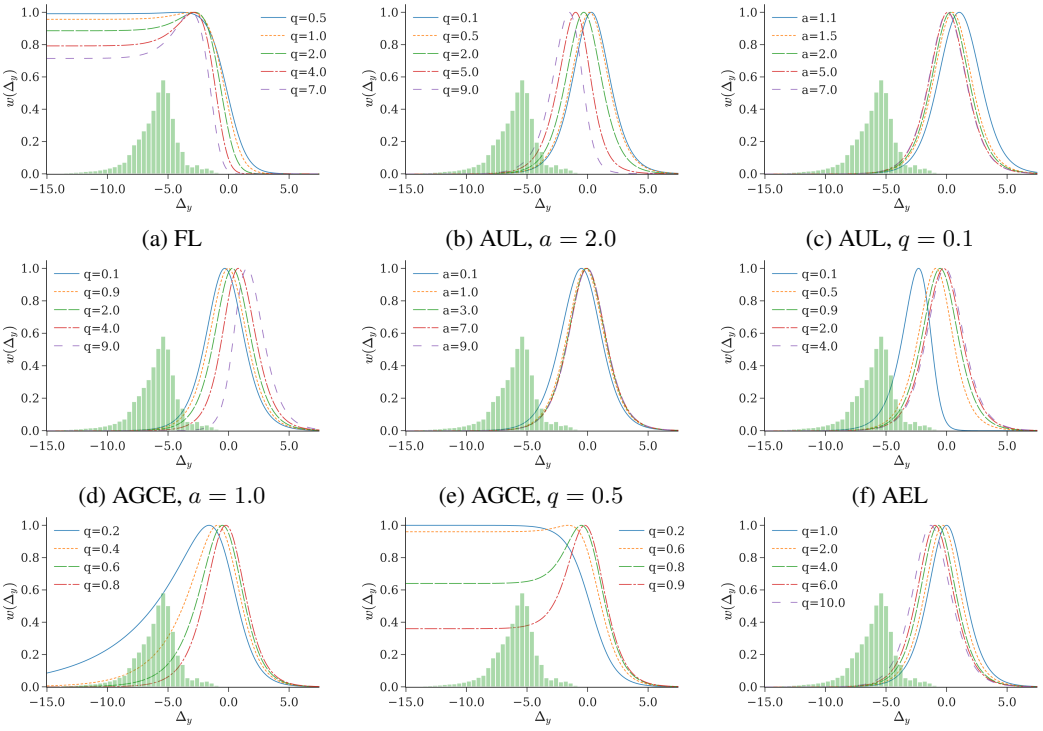
Taylor Cross Entropy (TCE; Feng et al. 2020)

$$L_{\mathrm{TCE}}(\boldsymbol{s}, y) = \sum_{i=1}^{q} \frac{(1-p_y)^i}{i}$$

is derived from Taylor series of the $\log$ function. It reduces to MAE when $q = 1$. Interestingly, the summand of TCE $(1-p_y)^i/i$ with $i > 2$ is proportional to AUL with $a = 1$ and $q = i$. Thus TCE can be viewed as a combination of symmetric and asymmetric loss functions.

Name	Function	Sample Weight w	Constraints
CE	$-\log p_y$	$1-p_y$	
FL	$-(1-p_y)^q \log p_y$	$(1-p_y)^q(1-p_y-qp_y\log p_y)$	$q>0$
MAE/RCE	$1-p_y$	$p_y(1-p_y)$	
AUL	$\frac{(a+1)-(a+p_y)^q}{q}$	$p_y(1-p_y)(a-p_y)^{q-1}$	$a>1, q>0$
AGCE	$\frac{(a-p_y)^q-(a-1)^q}{q}$	$p_y(a+p_y)^{q-1}(1-p_y)$	$a>0, q>0$
AEL	$e^{-p_y/q}$	$\frac{1}{q}p_y(1-p_y)e^{-p_y/q}$	$q>0$
GCE	$(1-p_y^q)/q$	$p_y^q(1-p_y)$	$0<q\leq 1$
SCE	$-(1-q)\log p_y + q(1-p_y)$	$(1-q+q\cdot p_y)(1-p_y)$	$0<q<1$
TCE	$\sum_{i=1}^{q}(1-p_y)^i/i$	$p_y\sum_{i=1}^{q}(1-p_y)^i$	$q\geq 1$

Table 6: Expressions, constraints of hyperparameters and sample-weighting functions of loss functions reviewed in Appendix A that follows the standard form Eq. (6).



(a) FL (b) AUL, $a = 2.0$ (c) AUL, $q = 0.1$

(d) AGCE, $a = 1.0$ (e) AGCE, $q = 0.5$ (f) AEL

(g) GCE (h) SCE (i) TCE

Figure 5: How hyperparameters affect the sample-weighting functions of loss functions in Table 6. The initial Δ_y distribution of CIFAR100 are included as reference.

Ma et al. (2020) propose weighted combinations of active and passive loss functions. We include NCE+MAE as an example:

$$L_{\text{NCE+MAE}}(\boldsymbol{s}, y) = a \cdot L_{\text{NCE}}(\boldsymbol{s}, y) + b \cdot L_{\text{MAE}}(\boldsymbol{s}, y) \propto (1-q) \cdot \frac{-\log p_y}{\sum_{i=1}^{k} -\log p_i} + q \cdot (1-p_y)$$

where $a > 0$, $b > 0$, and $0 < q < 1$.

Name	Original	Primary Loss	Regularizer
MSE	$1-2p_y+\sum_{i=1}^k p_i^2$	$1-p_y$	$\sum_{i=1}^k p_i^2$
PL(CR)	$-\log p_y+\log p_{y_n\|\boldsymbol{x}_m}$	$-\log p_y$	$\sum_{i=1}^k P(\tilde{y}=i)\log p_i$
CE+GLS	$-\sum_{i=1}^k[\mathbb{I}(i=y)(1-\alpha)+\frac{\alpha}{k}]\log p_i$	$-\log p_y$	$\pm\sum_{i=1}^k \frac{1}{k}\log p_i$
NCE	$\frac{-\log p_y}{\sum_{i=1}^k -\log p_i}$	$\mathrm{sg}\left(\frac{1}{\sum_{i=1}^k \log p_i}\right)\log p_i$	$\sum_{i=1}^k \frac{1}{k}\log p_i$

Table 7: Original expressions, primary loss functions in the standard form Eq. (6) and regularizers for loss functions reviewed in Appendix A.5. We view PL in its expectation to derive its regularizer. $p_{y_n|\boldsymbol{x}_m}$ is the softmax probability of a random label y_n with a random input $\boldsymbol{x}_m$ sampled from the noisy data. $\mathrm{sg}(\cdot)$ stands for the stop-gradient operator.

A.5 Loss Functions with Additional Regularizers

We additionally review loss functions that implicitly involve a regularizer and a primary loss function that fits the standard form Eq. (6). See Table 7 for a summary. We leave investigation on how these additional regularizers affect noise robustness for future work.

Mean Square Error (MSE; Ghosh et al. 2017)

$$\begin{aligned} L_{\mathrm{MSE}}(\boldsymbol{s},y) &= \sum_{i=1}^k(\mathbb{I}(i=y)-p_i)^2 = 1-2p_y+\sum_{i=1}^k p_i^2 \\ &\propto 1-p_y+\frac{1}{2}\cdot\sum_{i=1}^k p_i^2 = L_{\mathrm{MAE}}(\boldsymbol{s},y)+\alpha\cdot R_{\mathrm{MSE}}(\boldsymbol{s}) \end{aligned}$$

is more robust than CE, where $\alpha=0.5$ and the regularizer

$$R_{\mathrm{MSE}}(\boldsymbol{s})=\sum_{i=1}^k p_i^2 \tag{10}$$

increases the entropy of the softmax output. We can generalize α as a hyperparamter, making MSE a combination of MAE and an entropy regularizer R_{MSE}.

Peer Loss (PL; Liu & Guo 2020)

$$L_{\mathrm{PL}}(\boldsymbol{s},y)=L(\boldsymbol{s},y)-L(\boldsymbol{s}_n,y_m)$$

makes a generic loss function $L(\boldsymbol{s},y)$ robust against label noise, where $\boldsymbol{s}_n$ and y_m denote scores (of input $\boldsymbol{x}_n$) and labels randomly sampled from the noisy data. Its noise robustness is theoretically established for binary classification and extended to multi-class setting (Liu & Guo, 2020). Cheng et al. (2020) later show that PL in its expectation is equivalent to the original loss plus a **Confidence Regularizer** (CR; Cheng et al. 2020):

$$R_{\mathrm{CR}}(\boldsymbol{s})=-\mathbb{E}_{\tilde{y}}[L(\boldsymbol{s},\tilde{y})]$$

Substituting L with the standard L_{CE}, $R_{\mathrm{CR}}(\boldsymbol{s})$ becomes

$$R_{\mathrm{CR}}(\boldsymbol{s})=-\mathbb{E}_{\tilde{y}}[-\log p_{\tilde{y}}]=\sum_{i=1}^k P(\tilde{y}=i)\log p_i \tag{11}$$

Minimizing $R_{\mathrm{CR}}(\boldsymbol{s})$ thus makes the softmax output distribution $\boldsymbol{p}$ deviate from the prior label distribution of the noisy dataset $P(\tilde{y}=i)$, reducing the entropy of the softmax output.

Lukasik et al. (2020) show that label smoothing (Szegedy et al., 2015) can mitigate overfitting with label noise. Cross entropy with **Generalized Label Smoothing** (GLS; Wei et al. 2022a)

$$\begin{aligned} L_{\text{CE+GLS}}(\boldsymbol{s}, y) &= \sum_{i=1}^{k} -[\mathbb{I}(i=y)(1-\alpha) + \frac{\alpha}{k}] \log p_i \\ &= -(1-\alpha)\log p_y - \alpha \cdot \frac{1}{k}\sum_{i=1}^{k} \log p_i \\ &\propto -\log p_y - \frac{\alpha}{1-\alpha} \cdot \frac{1}{k}\sum_{i=1}^{k} \log p_i = L_{\text{CE}}(\boldsymbol{s}, y) + \alpha' \cdot R_{\text{GLS}}(\boldsymbol{s}) \end{aligned}$$

where $\alpha' = \alpha/(1-\alpha)$, has regularizer R_{GLS}

$$R_{\text{GLS}}(\boldsymbol{s}) = -\sum_{i=1}^{k} \frac{1}{k} \log p_i \tag{12}$$

With $\alpha' > 0$, R_{GLS} corresponds to the original label smoothing, which increases the entropy of softmax outputs. In contrast, $\alpha' < 0$ corresponding to negative label smoothing (Wei et al., 2022a), which decreases the output entropy similar to R_{CR}.

Finally, with equivalent derivatives, **NCE** discussed in §3 can be decomposed into

$$\begin{aligned} L_{\text{NCE}}(\boldsymbol{s}, y) &= \frac{1}{\sum_{i=1}^{k} -\log p_i} \left\{ -\log p_y + \frac{k \log p_y}{\sum_{i=1}^{k} \log p_i} \cdot \left[\frac{1}{k} \sum_{i=1}^{k} \log p_i \right] \right\} \\ &= \gamma_{\text{NCE}} \cdot [L_{\text{CE}}(\boldsymbol{s}, y) + \epsilon_{\text{NCE}} \cdot R_{\text{NCE}}(\boldsymbol{s})] \end{aligned}$$

where γ_{NCE} and ϵ_{NCE} are scalar weights wrapped with the stop-gradient operator, and

$$R_{\text{NCE}}(\boldsymbol{s}) = \sum_{i=1}^{k} \frac{1}{k} \log p_i \tag{13}$$

is the same regularizer as R_{GLS} with a negative weight $-\epsilon_{\text{NCE}}$.

B Detailed Experimental Settings

Label noise The synthetic noisy labels are generated following (Ma et al., 2020; Zhou et al., 2021b; Patrini et al., 2017). For symmetric label noise, the training labels are randomly flipped to a different class with with probabilities $\eta \in \{0.2, 0.4, 0.6, 0.8\}$. Asymmetric label noise are generated by a class-dependent flipping pattern. On CIFAR100, the 100 classes are grouped into 20 super-classes each having 5 sub-classes. Each class are flipped within the same super-class into the next in a circular fashion. The flip probabilities are $\eta \in \{0.1, 0.2, 0.3, 0.4\}$. Human label noise for CIFAR10/100 are adopted from Wei et al. (2022b). We use the "worst" labels of CIFAR10-N and the "fine" labels of CIFAR100-N, both leading to $\eta = 0.4$.

Models and hyperparameters We use a 4-layer CNN for MNIST, an 8-layer CNN for CIFAR10, a ResNet-34 (He et al., 2016) for CIFAR100, and a ResNet-50 (He et al., 2016) for WebVision, all with batch normalization (Ioffe & Szegedy, 2015). Data augmentation on CIFAR10/100 includes random width/height shift and horizontal flip. On WebVision, we additionally include random cropping and color jittering. Without further specifications, all models are trained using SGD with momentum 0.9 and batch size 128 for 50, 120, 200 and 250 epochs on MNIST, CIFAR10, CIFAR100 and WebVision, respectively. Learning rates with cosine annealing are 0.01 on MNIST and CIFAR10, 0.1 on CIFAR100, and 0.2 on WebVision. Weight decays are 10^{-3} on MNIST, 10^{-4} on CIFAR10, 10^{-5} on CIFAR100 and 3×10^{-5} on WebVision. All loss functions are normalized to have unit maximum in sample weights, which is different from (Ma et al., 2020). Hyperparameters of loss functions are listed in Tables 8 and 10 for different experiments.

C Additional Results to Understand Robust Loss Functions

We complement §4 in the main text with detailed derivations and additional results. Appendix C.1 extends discussions in §4.1 while Appendix C.2 extends §4.2

	SCE	GCE	NCE+MAE	AUL	AGCE	AUL†	AGCE†
a	/	/	/	1.1	0.1	3.0	1.6
q	0.7	0.3	0.3	5.0	0.1	0.7	2.0

Table 8: Hyperparameters of different loss functions for results in §4.1 and Appendix C.1. They are tuned on CIFAR100 without label noise. Settings with inferior hyperparameters are denoted with †.

Underfitting	Loss	CIFAR100 Acc.	CIFAR100 $\bar{\alpha}_t^*$	CIFAR10 Acc.	CIFAR10 $\bar{\alpha}_t^*$
/	CE	68.76 ± 0.21	0.962	90.24 ± 0.14	0.624
No	SCE	68.89 ± 0.05	1.165	91.07 ± 0.09	0.726
	GCE	69.00 ± 0.24	0.956	90.83 ± 0.20	0.644
	NCE+MAE	68.21 ± 0.51	/	90.14 ± 0.09	/
Moderate	AUL	47.98 ± 3.48	0.485	88.94 ± 0.29	0.604
	AGCE	43.51 ± 2.58	0.406	90.71 ± 0.19	0.549
	NCE	57.95 ± 0.26	/	85.96 ± 0.21	/
Severe	MAE	9.11 ± 0.83	0.025	90.65 ± 0.10	0.355
	AUL†	10.04 ± 2.33	0.023	90.77 ± 0.04	0.337
	AGCE†	5.34 ± 0.67	0.008	81.59 ± 8.55	0.243

Table 9: Similar results as Table 2 with learning rate $\alpha = 0.01$.

C.1 Understanding Underfitting of Robust Loss Functions

Hyperparameters We list the hyperparameters tuned on CIFAR100 without label noise for experiments in §4.1 and Appendix C.1.

Robust loss functions can underfit. In Table 9 we report similar results as Table 2 with learning rate $\alpha = 0.01$. Although settings that severe underfit slightly improve, they still perform much worse than CE, which further confirms that underfitting results from robust loss functions themselves.

Simulated Δ_y approximate real settings. We compare the simulated Δ_y distributions based on our assumptions in §4.1 to that of real datasets at initialization in Fig. 6. The expectations of simulated Δ_y mostly follow real settings, which supports the analysis in §4.1.

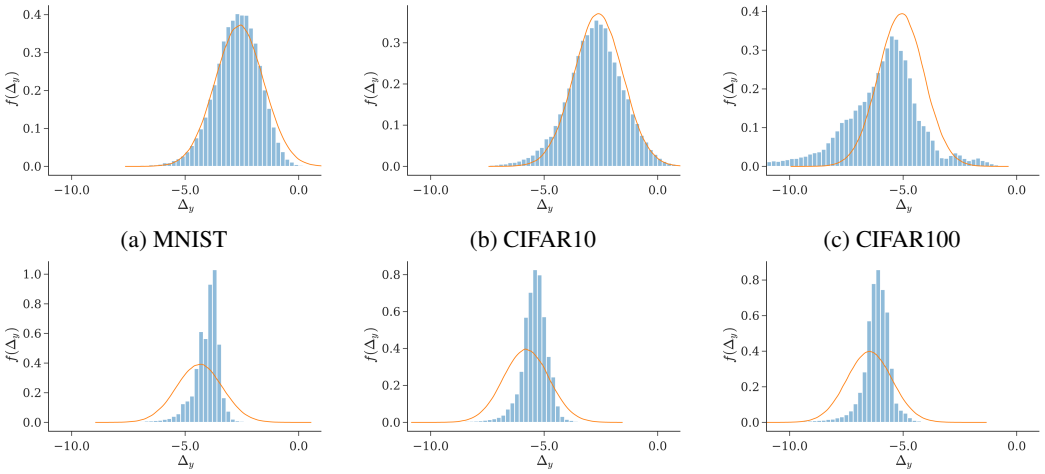
Derivation for $\mathbb{E}(\Delta_y)$ at initialization in Eq. (9):

$$
\begin{aligned}
\mathbb{E}(\Delta_y) &= \mathbb{E}[s_y - \log \sum_{i \neq y} e^{s_i}] = \mu - \mathbb{E}[\log \sum_{i \neq y} e^{s_i}] \\
&\approx_1 \mu - \log \mathbb{E}[\sum_{i \neq y} e^{s_i}] + \frac{\mathbb{V}[\sum_{i \neq y} e^{s_i}]}{2\mathbb{E}[\sum_{i \neq y} e^{s_i}]^2} \\
&=_2 \mu - \log\{(k-1)\mathbb{E}[e^{s_y}]\} + \frac{(k-1)\mathbb{V}[e^{s_y}]}{2\{(k-1)\mathbb{E}[e^{s_y}]\}^2} \\
&=_3 \mu - \log[(k-1)e^{\mu+\sigma^2/2}] + \frac{(k-1)(e^{\sigma^2}-1)e^{2\mu+\sigma^2}}{2[(k-1)e^{\mu+\sigma^2/2}]^2} \\
&= -\log(k-1) - \sigma^2/2 + \frac{e^{\sigma^2}-1}{2(k-1)}
\end{aligned}
$$

where $\approx_1$ follows $\mathbb{E}[\log X] \approx \log \mathbb{E}[X] - \mathbb{V}[X]/(2\mathbb{E}[X]^2)$ (Teh et al., 2006), $=_2$ utilize properties of sum of log-normal variables (Cobb et al., 2012), and $=_3$ substitutes the expression of $\mathbb{E}[e^{s_y}]$ and $\mathbb{V}[e^{s_y}]$ for log-normal distributions.

C.1.1 Addressing Underfitting from Marginal Initial Sample Weights

Hyperparamter τ for different settings The hyperparameter τ controlling the shape of modified sample-weighting function $w^+(\Delta_y)$ and $w^*(\Delta_y)$ can affect the noise robustness. Thus we tune τ



(a) MNIST (b) CIFAR10 (c) CIFAR100

(d) WebVision50 (e) WebVision200 (f) WebVision400

Figure 6: Comparison between simulated and real Δ_y distributions at initialization. The simulation is based on our assumption that class scores follow normal distribution $s_i \sim \mathcal{N}(0, 1)$ at initialization. Histograms are real distributions while the curves are simulations, with the vertical axis denoting probability density $f(\Delta_y)$.

for the best performance under different noise types and noise rates as listed in Table 11. A larger noise rate η requires a larger τ to avoid overfitting noisy samples.

Additional results with $w^*(\Delta_y)$ and $w^+(\Delta_y)$. We report additional results under symmetric and asymmetric label noise with diverse noise rates η in Table 12 and Table 13, respectively. Performance of MAE and AGCE gets substantially improved with $w^*(\Delta_y)$ and $w^+(\Delta_y)$.

C.2 Noise Robustness of Loss Functions

Computation of $\bar{w}_{\text{clean}}$ and $\bar{w}_{\text{noise}}$ for snr in Table 5 The average weight for clean samples, adiusted by the learning rate at each step α_t, can be

$$\bar{w}_{\text{clean}} = \frac{\sum_{i,t} \alpha_t \cdot \mathbb{I}(\tilde{y}_{i,t} = y_{i,t}) w_{i,t}}{\sum_{i,t} \alpha_t \cdot \mathbb{I}(\tilde{y}_{i,t} = y_{i,t})}$$

where $w_{i,t}$ denotes the weight of i-th sample of the batch at step t, $\tilde{y}_{i,t}$ is the potentially corrupted label and $y_{i,t}$ the uncorrupted label. Similarly, for noisy samples,

$$\bar{w}_{\text{noise}} = \frac{\sum_{i,t} \alpha_t \cdot \mathbb{I}(\tilde{y}_{i,t} \neq y_{i,t}) w_{i,t}}{\sum_{i,t} \alpha_t \cdot \mathbb{I}(\tilde{y}_{i,t} \neq y_{i,t})}$$

Hyperparameters We list the hyperparameters for different loss functions in Table 10 for results in §4.2 and Appendix C.2. In Fig. 7, we plot the sample-weighting functions of different loss functions.

Evolution of Δ_y distribution under different label noise and loss functions Complementing Fig. 3, in Fig. 8 we plot how distribution of Δ_y evolve during training on CIFAR10 with additional types of label noise using hyperparameters in Table 10. They follow similar trends as in Fig. 3, thus supporting analysis in §4.2. As MAE is not robust against asymmetric label noise with high η (Ghosh et al., 2017), it results in worse performance. We also include results with additional loss functions in Fig. 9. Since optimal hyperparameters will results in very similar sample weighting functions, we choose hyperparameters for a broad coverage of $w(\Delta_y)$ to better understand how they affects robustness.

	AUL	AGCE	GCE	SCE
a	2.0	3.0	/	/
q	2.0	4.0	0.4	0.95

Table 10: Hyperparameters of different loss functions for results in §4.2 and Appendix C.2. They are selected for broad coverage of shapes, scales and horizontal locations of sample-weighting functions instead of optimal performance on CIFAR10.

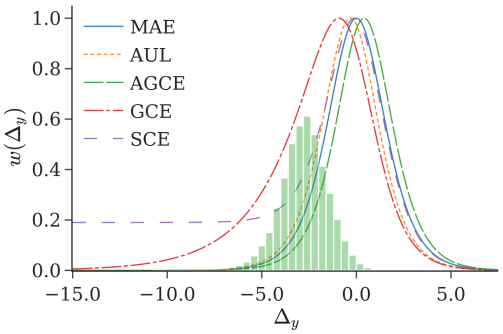


Figure 7: Plots of sample-weighting functions for results in Table 5 with hyperparameters in Table 8.

Settings	Clean 0.0	Symmetric 0.2	0.4	0.6	0.8	Asymmetric 0.1	0.2	0.3	0.4	Human 0.4
MAE shift	2.0	2.6	3.0	4.0	4.0	2.6	2.6	3.0	3.0	2.6
MAE scale	2.0	2.6	2.6	3.0	4.0	2.6	2.6	3.0	3.0	2.6
AGCE shift	4.0	4.0	4.0	5.0	5.0	4.0	4.0	4.0	4.0	4.0
AGCE scale	2.6	3.0	4.0	5.0	5.0	3.0	3.0	3.0	3.0	4.0

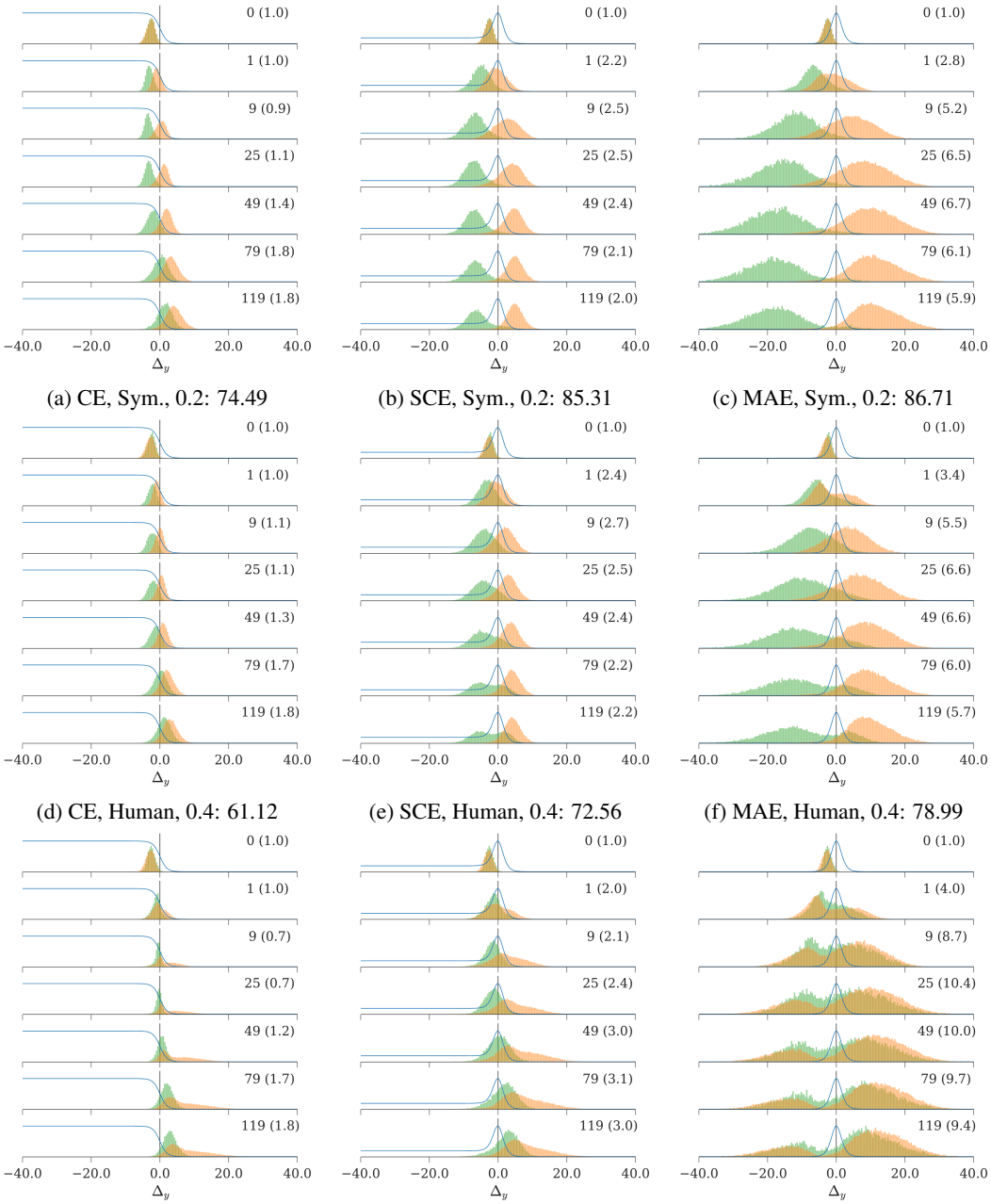
Table 11: Hyperparameter τ of $w^*(\Delta_y)$ and $w^+(\Delta_y)$ for different settings on CIFAR100. For reference, $\mathbb{E}[\Delta_y] = 5.097$ on CIFAR100 and $\mathbb{E}[\Delta_y] = 2.717$ on CIFAR10. Better performance can be achieved with a more thorough hyperparameter search.

Loss	Clean $\eta = 0$	Symmetric Noise (Noise Rate η) $\eta = 0.2$	$\eta = 0.4$	$\eta = 0.6$	$\eta = 0.8$
CE‡	71.33 ± 0.43	56.51 ± 0.39	39.92 ± 0.10	21.39 ± 1.17	7.59 ± 0.20
GCE‡	63.09 ± 1.39	61.57 ± 1.06	56.11 ± 1.35	45.28 ± 0.61	17.42 ± 0.06
NCE‡	29.96 ± 0.73	25.27 ± 0.32	19.54 ± 0.52	13.51 ± 0.65	8.55 ± 0.37
NCE+AUL‡	68.96 ± 0.16	65.36 ± 0.20	59.25 ± 0.23	46.34 ± 0.21	23.03 ± 0.64
AGCE	49.27 ± 1.03	49.17 ± 2.15	47.76 ± 1.75	38.17 ± 1.43	16.03 ± 0.59
AGCE shift	69.39 ± 0.84	62.81 ± 0.42	48.21 ± 1.06	36.70 ± 2.89	14.49 ± 0.17
AGCE scale	70.57 ± 0.62	64.73 ± 0.98	56.69 ± 0.33	39.02 ± 1.20	14.64 ± 0.79
MAE	3.69 ± 0.59	2.92 ± 0.46	1.29 ± 0.50	2.27 ± 1.24	1.00 ± 0.00
MAE shift	69.02 ± 0.78	59.75 ± 0.84	44.60 ± 0.24	24.27 ± 0.26	8.08 ± 0.26
MAE scale	**70.97 ± 0.41**	**66.83 ± 0.84**	**60.57 ± 1.04**	**49.23 ± 1.22**	**24.44 ± 0.73**

Table 12: Addition results to Table 3 with more symmetric label noise rates on CIFAR100.

Loss	Clean $\eta = 0$	Asymmetric Noise (Noise Rate η) $\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$	$\eta = 0.4$
CE‡	71.33 ± 0.43	64.85 ± 0.37	58.11 ± 0.32	50.68 ± 0.55	40.17 ± 1.31
GCE‡	63.09 ± 1.39	63.01 ± 1.01	59.35 ± 1.10	53.83 ± 0.64	40.91 ± 0.57
NCE‡	29.96 ± 0.73	27.59 ± 0.54	25.75 ± 0.50	24.28 ± 0.80	20.64 ± 0.40
NCE+AUL‡	68.96 ± 0.16	66.62 ± 0.09	63.86 ± 0.18	50.38 ± 0.32	38.59 ± 0.48
AGCE	49.27 ± 1.03	47.53 ± 0.73	46.77 ± 2.37	39.82 ± 2.70	33.40 ± 1.57
AGCE shift	69.39 ± 0.84	63.03 ± 0.42	55.84 ± 0.78	49.05 ± 0.81	40.76 ± 0.74
AGCE scale	70.57 ± 0.62	67.13 ± 0.60	59.71 ± 0.10	48.23 ± 0.29	39.71 ± 0.17
MAE	3.69 ± 0.59	3.59 ± 0.56	3.19 ± 0.98	2.11 ± 1.93	2.53 ± 1.34
MAE shift	68.57 ± 0.54	63.44 ± 0.32	56.47 ± 0.48	48.79 ± 1.22	39.83 ± 0.18
MAE scale	**70.97 ± 0.41**	**69.50 ± 0.24**	**64.80 ± 0.49**	**59.04 ± 1.52**	**44.48 ± 1.05**

Table 13: Addition results to Table 3 with more asymmetric label noise rates on CIFAR100.

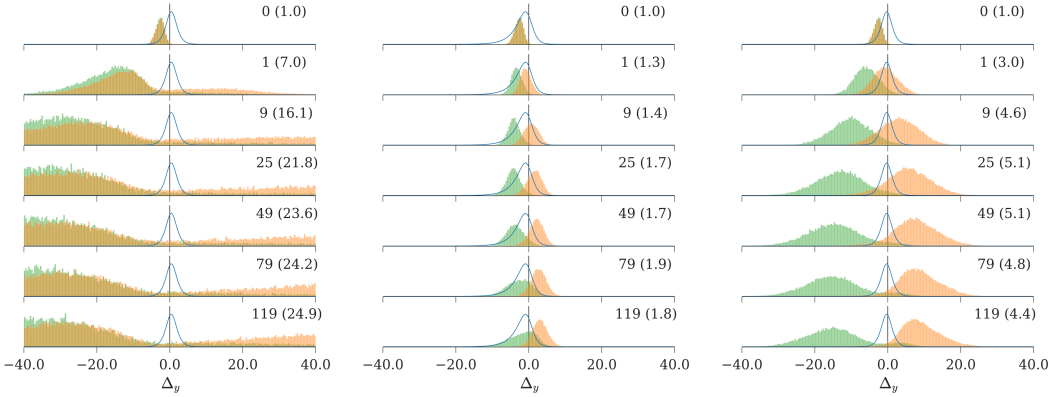


(a) CE, Sym., 0.2: 74.49 (b) SCE, Sym., 0.2: 85.31 (c) MAE, Sym., 0.2: 86.71

(d) CE, Human, 0.4: 61.12 (e) SCE, Human, 0.4: 72.56 (f) MAE, Human, 0.4: 78.99

(g) CE, Asym., 0.4: 73.75 (h) SCE, Asym., 0.4: 72.44 (i) MAE, Asym., 0.4: 61.11

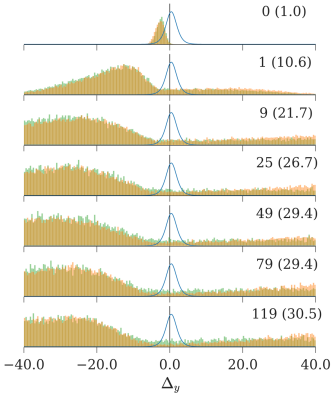
Figure 8: Additional results to Fig. 3 with different label noise: (a-c) symmetric label noise with $\eta = 0.2$; (d-f) human label noise with $\eta = 0.4$; (g-i) asymmetric label noise with $\eta = 0.4$. Noisy samples are colored green (on the left) and clean samples are orange (on the right). Test accuracies are included in the caption for reference.



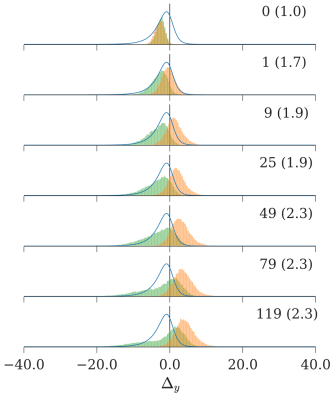
(a) AGCE, Sym., 0.4: 44.52

(b) GCE, Sym., 0.4: 66.60

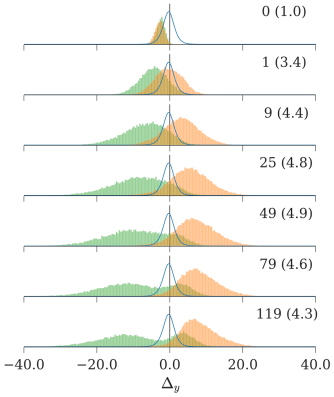
(c) AUL, Sym., 0.4: 84.12



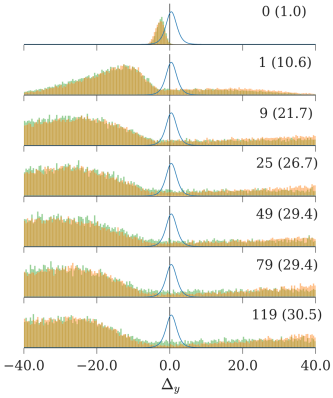
(d) AGCE, Human, 0.4: 35.73



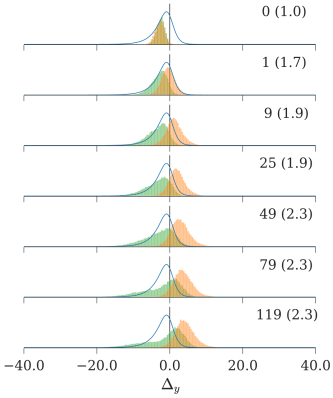
(e) GCE, Human, 0.4: 68.22



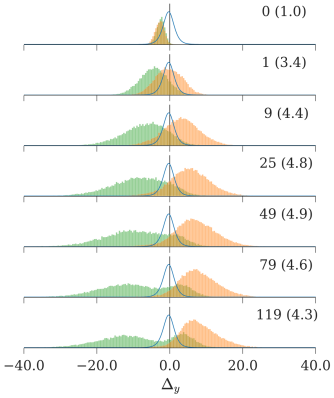
(f) AUL, Human, 0.4: 77.22



(g) AGCE, Asym., 0.4: 46.75



(h) GCE, Asym., 0.4: 73.23



(i) AUL, Asym., 0.4: 67.72

Figure 9: Additional results to Fig. 3 with more robust loss functions under different label noise: (a-c) symmetric label noise with $\eta = 0.4$; (d-f) human label noise with $\eta = 0.4$; (g-i) asymmetric label noise with $\eta = 0.4$. Test accuracies are included in the caption for reference. Noisy samples are colored green (on the left) and clean samples are orange (on the right). Hyperparameters of these loss functions are selected for broad coverage rather than optimal performance.