A Curriculum Perspective of Robust Loss Functions

Anonymous Author(s)
Affiliation
Address
email

Abstract

Learning with noisy labels is a fundamental problem in machine learning. A large body of work aims to design loss functions robust against label noise. However, it remains open questions why robust loss functions can underfit and why loss functions deviating from theoretical robustness conditions can appear robust. To tackle these questions, we show that each loss function implicitly induces a sample-weighting curriculum. We then adopt the resulting curriculum perspective to analyze how robust losses interact with various training dynamics, which helps elucidate the above questions. Based on our findings, we propose simple fixes to make robust losses that severely underfit competitive to state-of-the-art losses. Notably, our curriculum perspective connects robust losses to the seemingly distinct approaches based on curriculum learning.[1]

1 Introduction

Labeling errors are non-negligible [1] in datasets from expert annotations [2, 3], crowd-sourcing [4] and automatic annotations [5, 6]. The resulting noisy labels can hamper generalization, as over-parameterized neural networks can memorize all training samples [7]. To combat the impact of noisy labels, a large body of research aims to design loss functions robust against label noise [8–13], which is the main focus of this work. Theoretical results show that losses satisfying robustness conditions [9, 11] will lead to the same optimum with clean or noisy labels.

Existing robustness conditions are agnostic to training dynamics. Although theoretically appealing, they may fail to fully characterize the performance of robust losses with noisy labels. In particular, it has been shown that (1) robust losses can underfit difficult tasks [1, 10, 12, 13], while (2) losses failing to satisfy theoretical robustness conditions [12–14] can exhibit robustness. The reasons behind these observations remain open questions. For (1), existing explanations [10, 15] can be limited, which we discuss in §2.3. For (2), to our knowledge, there is no work directly addressing it.

To tackle the above questions, we consider training dynamics in our analysis, which complements existing robustness conditions. By rewriting loss functions into a standard form, we show that each loss function implicitly induces a sample-weighting curriculum (§3). Such a perspective connects robust losses to the seemingly distinct [1] approaches with curriculum learning [16–20] for noise-robust training. The original definition [21] of curriculum learning aims to present training samples with gradually increasing difficulty and diversity to ease learning. We adopt a generalized definition of curriculum [22], i.e., a *curriculum* specifies a sequence of *re-weighting* of training sample distributions, which can manifest as sample weighting [16–18] or sample selection [19, 20].

The curriculum perspective helps elucidate underfitting and noise robustness from the interaction between the sample-weighting curriculums and various training dynamics. We first attribute the underfitting problem to the marginal average sample weights from the implicit curriculums (§4.1) We then show that an increased number of classes leads to low initial sample weights with fixed curriculums (§4.2). By adapting the curriculums accordingly, we make robust losses that severely

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

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

underfit perform competitively to state-of-the-art losses (§4.2). Finally, we attribute the noise robustness of losses to higher average sample weights for clean samples compared to noisy ones (§4.3). We hypothesize that with sample-weighting curriculums that magnify the learning speed differences and neglect unlearnt samples, clean samples will receive more weights, explaining the evolution of sample performance during training (§4.3). Such an explanation leads to two unexpected results when viewed from existing robustness conditions: by simply changing the learning rate schedule, robust losses can be vulnerable to label noise and cross entropy can appear robust (§4.3).

2 Background

We review classification, label noise, sufficient conditions for noise robustness and several typical robust loss functions before summarizing the open questions that we aim to address.

2.1 Classification with Label Noise and Noise Robustness

The k-ary classification problem with input $\boldsymbol{x} \in \mathbb{R}^d$ and ground truth label $y \in \{1, \dots, k\}$ can be solved with $\hat{y} = \arg\max_j s_j$, where s_j corresponds to the j-th class in the class scoring function $\boldsymbol{s}_\theta : \mathbb{R}^d \to \mathbb{R}^k$ with parameter θ. Given a loss function $L(\boldsymbol{s}_\theta(\boldsymbol{x}), y)$, we can estimate the optimal parameter with $\theta^* = \arg\min_\theta \mathbb{E}_{\boldsymbol{x},y} L(\boldsymbol{s}_\theta(\boldsymbol{x}), y)$. The un-normalized class scores $\boldsymbol{s}_\theta(\boldsymbol{x})$ can be transformed to class probabilities with the softmax function, where the i-th element of the output $\boldsymbol{p}$ is $p_i = e^{s_i}/(\sum_j e^{s_j})$. For notation simplicity, we omit the dependence on θ and $\boldsymbol{x}$ if possible.

The annotation process may introduce errors, resulting in a potentially corrupted label $\tilde{y}$ following

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

Noise is symmetric (or uniform) if $P(\tilde{y} = j|\boldsymbol{x}, y) = \eta/k$, with $\eta = P(\tilde{y} \neq y)$ the noise rate constant. Noise is asymmetric (or class-conditional) if $P(\tilde{y} = j|\boldsymbol{x}, y) = P(\tilde{y} = j|y)$. A loss function L is robust against label noise if the optimal parameter $\tilde{\theta}^* = \arg\min_\theta \mathbb{E}_{\mathbf{x},\tilde{y}} L(\boldsymbol{s}_\theta(\boldsymbol{x}), \tilde{y})$ estimated using dataset with noisy labels $\tilde{y}$ equals to θ^*.

We review two existing sufficient conditions for noise robustness to demonstrate how they are *agnostic to models and training dynamics*. Loss L is called *symmetric* [9] if

$$\sum_j L(\boldsymbol{s}_\theta(\boldsymbol{x}), j) = C, \forall \boldsymbol{x}, \boldsymbol{s}_\theta \tag{1}$$

where C is a constant. With $\eta < (k-1)/k$, L is robust against symmetric (uniform) label noise. Such stringent condition is later extended by Zhou et al. [11]. Suppose a loss can be written as $L(\boldsymbol{s}_\theta(\boldsymbol{x}), j) = l(p_j)$. As an equivalent rephrase of the sufficient condition, L is called *asymmetric* if

$$\max_{j \neq y} \frac{P(\tilde{y} = j|\boldsymbol{x}, y)}{P(\tilde{y} = y|\boldsymbol{x}, y)} = \tilde{r} \leq r = \inf_{\substack{0 \leq p_j, \Delta p \leq 1 \\ p_j + \Delta p \leq 1}} \frac{l(p_j) - l(p_j + \Delta p)}{l(0) - l(\Delta p)} \tag{2}$$

With $\tilde{r} < 1$, i.e., the dataset is dominated by clean labels, L is robust under *generic* label noise.

2.2 Review of Selected Losses

In addition to cross entropy (CE) that is vulnerable to label noise [9, 11], we review several typical robust losses for our analysis. We *ignore the difference in constant scaling factors and constant additive bias in the loss functions*. They are either equivalent to learning rate scaling in SGD or irrelevant in gradient computation. See Table 1 for the exact formulas and Appendix A for derivations.

Symmetric Losses Mean absolute error (MAE) [9] and the subsequent reverse cross entropy (RCE) [13] are essentially equivalent, both satisfying Eq. (1). Ma et al. [10] normalize generic losses satisfying $L(\boldsymbol{s}, j) > 0, \forall j \in \{1, \dots, K\}$ into symmetric losses with $L_{\mathrm{N}}(\boldsymbol{s}, y) = L(\boldsymbol{s}, y)/(\sum_j L(\boldsymbol{s}, j))$. We include normalized cross entropy (NCE) as an example.

Asymmetric Losses We include two asymmetric losses [11]: asymmetric generalized cross entropy (AGCE) and asymmetric unhinged loss (AUL). Notably, with $q \geq 1$ for AGCE and $q \leq 1$ in AUL, both losses are complete asymmetric [11], i.e., Eq. (2) always holds when $\tilde{r} < 1$.

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	$-\log p_y / \sum_j -\log p_j$	$\gamma_{\mathrm{NCE}}\left(w_{\mathrm{CE}} + \frac{k-1}{k}\epsilon_{\mathrm{NCE}}\right)$	
Asym.	AUL	$[(a-p_y)^q - (a-1)^q]/q$	$p_y(1-p_y)(a-p_y)^{q-1}$	$a>1, q>0$
	AGCE	$[(a+1)-(a+p_y)^q]/q$	$p_y(a+p_y)^{q-1}(1-p_y)$	$a>0, q>0$
Comb.	GCE	$(1-p_y^q)/q$	$p_y^q(1-p_y)$	$0<q\leq 1$
	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	$(1-q)\cdot L_{\mathrm{NCE}} + q\cdot L_{\mathrm{MAE}}$	$(1-q)\cdot w_{\mathrm{NCE}} + q\cdot w_{\mathrm{MAE}}$	$0<q<1$

Table 1: Loss functions reviewed in §2.2, constraints of their hyperparameters and their sample-weighting curriculums derived in §3.1. Note that w_{NCE} is an approximation as discussed in §3.2.

Combined Losses Losses can be combined to enjoy both robustness and sufficient learning. For example, generalized cross entropy (GCE) [12] can be viewed as a smooth interpolation between CE and MAE. Alternatively, symmetric cross entropy (SCE) [13] uses a weighted average of CE and RCE (MAE). Finally, Ma et al. [10] propose the active-passive-loss framework and argue that robust and sufficient learning requires a balanced combination of active and passive losses. Suppose loss L can be rewritten into $L(\boldsymbol{s}, y) = \sum_j l(\boldsymbol{s}, j)$. An active loss requires $\forall j \neq y, l(\boldsymbol{s}, j) = 0$, while a passive one satisfies $\exists j \neq y, l(\boldsymbol{s}, j) \neq 0$. Accordingly, CE and NCE are active losses while MAE (RCE) is a passive loss. Note that passive losses can be viewed as a combination of active and passive losses. We include NCE+MAE as an example.

2.3 Open Questions Addressed in this Work

Why do robust losses underfit? Ma et al. [10] attribute underfitting to failure in balancing active-passive components. However, different specifications of $L(\boldsymbol{s}, y) = \sum_j l(\boldsymbol{s}, j)$ can lead to ambiguities in the active-passive dichotomy. For example, with $L_{\mathrm{MAE}}(\boldsymbol{s}, y) \propto \sum_j |\mathbb{I}(j=y) - p_y|$ where $\mathbb{I}(\cdot)$ is the indicator function, MAE is a passive loss; yet the equivalent $L_{\mathrm{MAE}}(\boldsymbol{s}, y) \propto \sum_j \mathbb{I}(j=y)(1-p_y)$ makes MAE an active loss. Wang et al. [15] view $\|\nabla_{\boldsymbol{s}} L(\boldsymbol{s}, y)\|_1$ as weights for sample gradients and attribute underfitting to their low variance, making clean and noisy samples less distinguishable. However, as we show in §4.1, MAE also underfits on clean datasets. Therefore, why robust losses underfit remains an open question.

What affects the robustness of a loss? Although combined losses such as GCE and SCE fail to satisfy existing robustness conditions in Eq. (1) and (2), it is unclear why they also exhibit robustness against label noise [12, 13]. Furthermore, it is unclear how training dynamics, which are irrelevant in existing robustness conditions, affect the noise robustness of a loss.

3 Implicit Curriculums of Robust Losses

We derive the standard form of losses and show that each implicitly induces a sample-weighting curriculum, which helps examine how robust losses interact with various training dynamics.

3.1 The Implicit Sample-Weighting Curriculum

Losses in Table 1 generally follow the form $L(\boldsymbol{s}, y) = l(p_y)$, where p_y is the softmax output of class y. Note that p_y can be rewritten as

$$p_y = \frac{e^{s_y}}{\sum_j e^{s_j}} = \frac{1}{e^{\log \sum_{j\neq y} e^{s_j - s_y}} + 1} = \frac{1}{e^{-\Delta_y} + 1} \tag{3}$$

where

$$\Delta_y = s_y - \log \sum_{j \neq y} e^{s_j} \leq s_y - \max_{j \neq y} s_j = \Delta_y^* \tag{4}$$

is the *soft margin* between s_y and the maximum of other scores, a smooth approximation of the *hard margin* Δ_y^*. With classifier $\hat{y} = \arg\max_j s_j$, Δ_y indicates how well a sampled is learnt, as $\Delta_y \geq 0$ leads to $\Delta_y^* \geq 0$, which ensures successful classification using scores $\boldsymbol{s}$. Since $\nabla_{\boldsymbol{s}} l(p_y) = l'(p_y) \cdot p_y'(\Delta_y) \cdot \nabla_{\boldsymbol{s}} \Delta_y$, losses can be rewritten into a standard form with *equivalent gradients*:

$$L(\boldsymbol{s}, y) = \text{stop_grad}[w(\Delta_y)] \cdot \Delta_y \tag{5}$$

where $\text{stop_grad}(\cdot)$ avoids backpropagating through $w(\Delta_y) = l'(p_y) \cdot p_y'(\Delta_y)$. Each loss in the form of Eq. (5) thus implicitly induces a sample-weighting curriculum with $w(\Delta_y)$ as the *sample weight* and Δ_y as the *implicit loss*. Such a view is only valid with first-order derivatives. By examining the interaction between $w(\Delta_y)$ and various training dynamics, we can elucidate the reasons behind underfitting and noise robustness. Table 1 shows the derived sample weights for losses in §2.2.

By analyzing $\nabla_{\boldsymbol{s}} L(\boldsymbol{s}, y)$, Wang et al. [14, 15] extract $\|\nabla_{\boldsymbol{s}} L(\boldsymbol{s}, y)\|_1$ as weights for sample gradients, which share similar forms in Table 1. Instead of directly weighting sample gradients, our derivation identifies the implicit loss Δ_y, making the resulting sample-weighting scheme compatible with the standard definition of curriculum learning [22]. In addition, the extracted Δ_y and Δ_y^* can serve as direct metrics for sample performance in curriculum learning, compared to loss [23, 24] and gradient magnitude [25] that are affected by preference specified with sample weight $w(\Delta_y)$ of a loss function. Finally, the Δ_y distribution is essential in analyzing the interaction between losses and training dynamics as shown in §4.

3.2 The Additional Entropy-Reducing Curriculum of NCE

As shown in Table 1, due to normalization, $L_{\text{NCE}}(\boldsymbol{s}, y)$ additionally depends on p_j where $j \neq y$, which cannot be trivially rewritten into Eq. (5). A derivation of the gradient gives

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

Thus NCE can be rewritten as

$$L_{\text{NCE}}(\boldsymbol{s}, y) = \gamma_{\text{NCE}} \cdot L_{\text{CE}}(\boldsymbol{s}, y) + \gamma_{\text{NCE}} \cdot \epsilon_{\text{NCE}} \cdot R_{\text{CE}}(\boldsymbol{s}) \tag{6}$$

The first term results in a similar sample-weighting curriculum as CE with an additional weight $\gamma_{\text{NCE}} = 1/(\sum_j -\log p_j) \leq 1/(k \log k)$ which has marginal magnitude when k is large. The second term is a regularizer

$$R_{\text{CE}}(\boldsymbol{s}) = -\frac{1}{k} \sum_j L_{\text{CE}}(\boldsymbol{s}, j) \tag{7}$$

which reduces the entropy of the softmax output $\boldsymbol{p}$. The regularizer has per-sample weights $\epsilon_{\text{NCE}} = k(-\log p_y)/(\sum_j -\log p_j)$. It can thus be interpreted as a regularization curriculum. Notably, the two curriculums work synergically in reducing the entropy of the softmax output. As Δ_y induces gradients with constant L1 norm $\|\nabla_{\boldsymbol{s}} \Delta_y\|_1 = 2$ as shown in Appendix B, we *approximate* the effective sample-weighting curriculum with w_{NCE} for analysis of underfitting (§4.1):

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

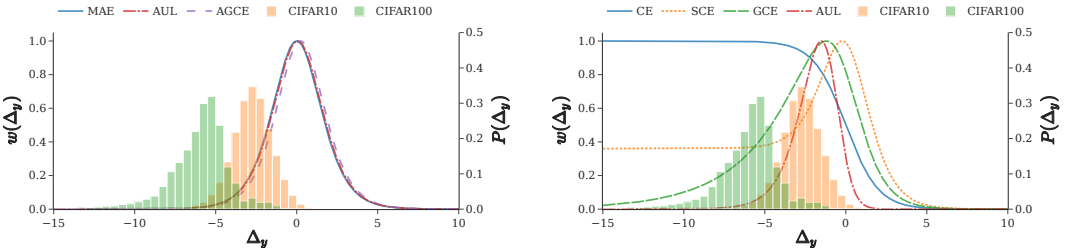
with derivations in Appendix B. Note that $\nabla_{\boldsymbol{s}} L_{\text{NCE}}(\boldsymbol{s}, y)$ and $\nabla_{\boldsymbol{s}} \Delta_y$ may have different directions.

4 Understanding Robust Losses with Their Implicit Curriculums

We empirically investigate the interaction between sample-weighting curriculum $w(\Delta_y)$ and various training dynamics for questions in §2.3, following the standard settings [10, 11]. Experiments are based on the standard MNIST [26] and CIFAR10/100 [27] benchmarks with synthetic symmetric and asymmetric label noises. We use a 4-layer CNN for MNIST, an 8-layer CNN for CIFAR10 and a ResNet-34 [28] for CIFAR100, all with batch normalization [29]. Without further specifications, all models are trained with a fixed number of epochs using SGD with momentum, weight decay and *cosine learning rate annealing*. Different from the standard settings, as hyperparameters of losses can change the maximum of $w(\Delta_y)$, which essentially adjust the learning rate of SGD, we normalize the maximum of $w(\Delta_y)$ to 1 to avoid complications. We leave hyperparameters, data augmentation and noise generation to Appendix C. We state explicitly when deviating from the standard settings.

Underfitting	Loss	Hyperparams	CIFAR100 Acc.	CIFAR100 $\bar{\alpha}_t^*$	CIFAR10 Acc.	CIFAR10 $\bar{\alpha}_t^*$
No	CE		71.33 ± 0.23	8.183	92.76 ± 0.30	5.541
No	GCE	$q = 0.3$	69.95 ± 0.40	8.861	92.96 ± 0.13	6.151
	SCE	$q = 0.7$	71.36 ± 0.39	9.541	93.17 ± 0.06	7.018
	NCE+MAE	$q = 0.3$	68.89 ± 0.23	2.971	92.37 ± 0.33	2.414
Moderate	NCE		43.18 ± 1.55	1.769	91.28 ± 0.22	1.072
	AUL	$a = 1.1, q = 5.0$	58.75 ± 1.07	5.278	92.43 ± 0.19	5.171
	AGCE	$a = 0.1, q = 0.1$	49.27 ± 1.03	4.537	92.61 ± 0.18	5.225
Severe	MAE		3.69 ± 0.59	0.035	91.56 ± 0.11	2.492
	AUL	$a = 3.0, q = 0.7$	3.13 ± 0.43	0.033	91.13 ± 0.06	2.308
	AGCE	$a = 1.6, q = 2.0$	1.62 ± 0.69	0.009	87.14 ± 4.96	1.701

Table 2: Robust losses can underfit on CIFAR100 but CIFAR10. We report test accuracy and average effective learning rate $\bar{\alpha}_t^*$ (scaled by 10^3) of 3 different runs at the final step without label noise, using learning rate $\alpha = 0.1$. See Table 5 in Appendix D for similar results with $\alpha = 0.01$. Hyperparameters of losses are tuned for the best test accuracy on CIFAR100. AGCE and AUL with inferior hyperparameters are included for comparison.



(a) Severe underfitting

(b) Moderate/no underfitting

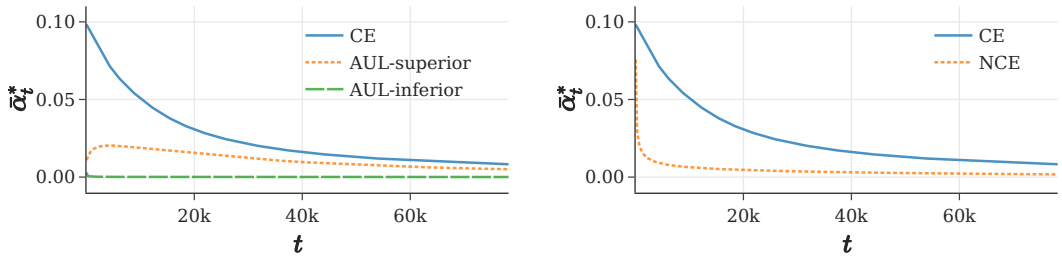
Figure 1: Sample weights $w(\Delta_y)$ of selected losses and hyperparameters in Table 2. We include the Δ_y distributions at initialization for CIFAR10 and CIFAR100.

4.1 Underfitting of Robust Losses from a Sample-Weighting Curriculum Perspective

Robust losses do underfit. We confirm that on difficult tasks like CIFAR100 [10, 12, 13], underfiting results from robust losses themselves rather than inferior hyperparameters. We tune hyperparameters of each loss on CIFAR100 and report results on CIFAR100 and CIFAR10 without label noise in Table 2. As shown in Table 2, the performance of NCE, AGCE and AUL lag behind CE by a nontrivial margin on CIFAR100. Notably, MAE performs much worse compared to CE, similar to AGCE and AUL with inferior hyperparameters. In contrast, there is no underfitting in CIFAR10. See Table 5 in Appendix D for similar results with a smaller learning rate.

Marginal effective learning rate α_t^* explains underfitting. We attribute underfitting of a loss to the diminishing effective learning rate $\alpha_t^* = \bar{w}_t \cdot \alpha_t$, where $\bar{w}_t$ is the average sample weights of the batch and α_t the learning rate at step t. Let $\bar{\alpha}_t^* = \sum_{i=1}^{t} \alpha_i^*/t$ be the average effective learning rate up to step t. In Table 2 we report $\bar{\alpha}_t^*$ at the final step. For losses that heavily underfit on CIFAR100, their final $\bar{\alpha}_t^*$ is marginal compare to CE.

Marginal α_t^* due to marginal initial sample weights. To examine why robust losses result in marginal α_t^*, in Fig. 1 we compare their sample-weighting functions $w(\Delta_y)$ to the Δ_y distribution of CIFAR10 and CIFAR100 at initialization. For robust losses that severely underfit (Fig. 1a), the Δ_y distribution of CIFAR100 concentrates at regions with marginal sample weights, resulting in small effective learning rate α_t^*. It can be hard for these samples to escape the region with marginal weights before the learning rate attenuates. In contrast, losses with non-trivial initial sample weights (Fig. 1b) result in moderate or no underfitting in Table 2. As a corroboration, we plot the average effective learning rate $\bar{\alpha}_t^*$ of AUL with different hyperparameters in Fig. 2a. With superior hyperparameters ($a = 1.1, q = 5.0$), $\bar{\alpha}_t^*$ quickly increase to a non-negligible value before annealing. In contrast, $\bar{\alpha}_t^*$ stays marginal with inferior hyperparameters ($a = 3.0, q = 0.7$).



(a) AUL with inferior/superior hyperparameters.

(b) NCE with estimated weight upperbound.

Figure 2: Different causes of underfitting: (a) marginal initial sample weights; (b) fast diminishing sample weights. We plot the average effective learning rate $\bar{\alpha}_t^*$ at different training steps t with selected losses on CIFAR100.

Marginal α_t^* due to fast diminishing sample weights. In Fig. 2b, different from other robust losses but similar to CE, α_t^* of NCE peaks at initialization. However, it decreases substantially faster compared to CE, which can be attributed to the synergy between the two implicit curriculums of NCE in reducing w_{NCE}. As Δ_y improves, γ_{NCE}, ϵ_{NCE} and w_{CE} all decreases. In addition, the regularizer $R_{\mathrm{CE}}(\boldsymbol{s})$ further decreases the entropy of softmax output and thus γ_{NCE}. As a result, w_{NCE} decreases much faster compared to w_{CE}, leading to faster attenuating α_t^*.

Loss combination mitigates marginal initial sample weights. As w_{CE} and w_{NCE} peak at initialization, they compensate the marginal initial sample weights when combined with other robust losses, helping initial learning and thus avoiding underfitting. In Table 2, the effective learning rate on CIFAR100 is substantially increased when combining MAE with CE and NCE. Interestingly, CE and NCE are both "active" as their sample weights peak at initialization, while other robust losses are "passive" due to their marginal initial sample weights. Such dichotomy based on sample-weighting curriculums complements the active-passive dichotomy of losses [10] from a distinct perspective.

4.2 Addressing Underfitting by Adapting the Sample-Weighting Curriculums

Although robust losses can underfit CIFAR100, the problem is absent on CIFAR10 in Table 2. Such difference has been vaguely attributed to the increased difficulty of CIFAR100 [1, 12]. We further show that the issue lies specifically in the increased number of classes k when combined with fixed sample-weighting curriculums. By adapting the curriculums accordingly, robust losses that severely underfit can become competitive with the state-of-the-art losses. Given findings in §4.1, we use MAE as a typical example.

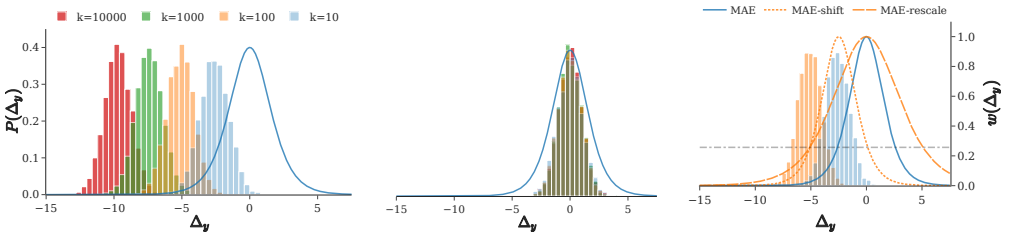
Intuitively, the larger number of classes, the more subtile differences to be distinguished among the classes, thus the harder the task is. In addition, the number of classes k determines the Δ_y distribution at initialization. Assuming that class scores s_j at initialization are i.i.d. variables following the normal distribution $s_j \sim \mathcal{N}(\mu, \sigma)$. In particular, $\mu = 0$ and $\sigma = 1$ for most neural networks with standard initializations [30] and normalization layers [29, 31]. $\Delta_y = s_y - \log \sum_{j \neq y} e^{s_j}$ is then a combination of a normal random variable and the logarithm of sum of log-normal random variables. We simulate the distribution of Δ_y with different number of classes k in Fig. 3a. The simulation fits the real distribution in geneal as shown in Fig. 6 of Appendix E. With more output class, the Δ_y distribution has similar variance but shifts further left, corresponding to diminishing sample weights with the fixed MAE curriculum. The expected Δ_y can be approximated with

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

See Appendix E for detailed derivations. Subtracting $\mathbb{E}(\Delta_y)$ from Δ_y centers the distribution at 0 where $w_{\mathrm{MAE}}(\Delta_y)$ peaks. As shown in Fig. 3b, the centered distributions virtually overlaps.

Shifting or rescaling Δ_y mitigates underfitting from increased number of classes. To assign nontrivial sample weights at initialization, the sample-weighting curriculum of robust losses should be adapted according to the number of classes k. A simple strategy is to make the expected initial sample weights agnostic to k. Given a sample-weighting function $w(\Delta_y)$, we can either shift

$$w^{\mathrm{shift}}(\Delta_y) = w\left(\Delta_y - \mathbb{E}[\Delta_y] + a\right) \tag{10}$$



(a) Simulated distributions (b) Shifted by $-\mathbb{E}[\Delta_y]$ (c) Expected initial $w(\Delta_y)$

Figure 3: Underfitting from increased number of classes k and how to fix it. The plot of standard $w_{\mathrm{MAE}}(\Delta_y)$ are included for reference. (a): the simulated distributions with different k; (b): shifting the distributions by $-\mathbb{E}[\Delta_y]$ centers them at zero; (c) illustration of our proposed modifications (shift and rescale) to $w_{\mathrm{MAE}}(\Delta_y)$, resulting in the same expected initial $w_{\mathrm{MAE}}(\Delta_y)$ for $k = 100$ and $k = 10$, which is indicated by the horizontal dash line.

Loss	Clean $\eta = 0$	Symmetric Noise (Noise Rate η) $\eta = 0.2$	$\eta = 0.4$	$\eta = 0.6$	$\eta = 0.8$
CE [11]	71.33 ± 0.43	56.51 ± 0.39	39.92 ± 0.10	21.39 ± 1.17	7.59 ± 0.20
GCE [11]	63.09 ± 1.39	61.57 ± 1.06	56.11 ± 1.35	45.28 ± 0.61	17.42 ± 0.06
NCE [11]	29.96 ± 0.73	25.27 ± 0.32	19.54 ± 0.52	13.51 ± 0.65	8.55 ± 0.37
NCE+AUL [11]	68.96 ± 0.16	65.36 ± 0.20	59.25 ± 0.23	46.34 ± 0.21	23.03 ± 0.64
MAE	3.56 ± 0.65	2.92 ± 0.46	1.29 ± 0.50	2.27 ± 1.24	1.00 ± 0.00
MAE-shift	68.13 ± 1.32	60.41 ± 0.99	45.06 ± 0.73	24.55 ± 0.53	7.58 ± 0.37
MAE-rescale	69.95 ± 1.21	66.42 ± 0.71	60.70 ± 0.30	45.17 ± 2.37	10.79 ± 0.97

Table 3: Shifting or rescaling Δ_y with $a = 2.6$ mitigates underfitting of MAE. Test accuracy with 3 different runs are reported on CIFAR100 with symmetric label noise. For comparison we also include results from [11]. Results for asymmetric label noise are reported in Table 6 of Appendix E.

or rescale

$$w^{\mathrm{rescale}}(\Delta_y) = w\left(\Delta_y / \mathbb{E}[\Delta_y] \cdot a\right) \tag{11}$$

it, where a is a hyperparameter. As shown in Table 3, we find that $a = 2.6$ makes MAE comparable to the previous best (NCE+AUL) [11] on CIFAR100 with symmetric label noise, which substantially outperforms the vanilla MAE that underfits. Notably, since $\mathbb{E}[\Delta_y] \approx 2.6$ when $k = 10$, $\mu = 0$ and $\sigma = 1$, we essentially match the expected initial sample weights of CIFAR100 to that of CIFAR10, as illustrated in Fig. 3c. For similar results with asymmetric label noise, see Table 6 in Appendix E.

4.3 Noise Robustness from a Sample-Weighting Curriculum Perspective

Intuitively, losses exhibiting noise robustness should weight clean samples more than noisy ones. We provide an explanation based on the interaction between $w(\Delta_y)$ and two different training dynamics. As NCE additionally involves a regularization curriculum, we leave its discussions for future work.

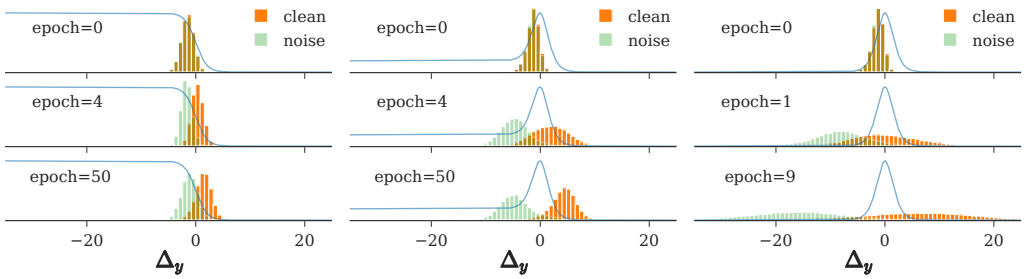
Robust losses assign larger weights to clean samples. The average weight assigned to noisy samples during training can be characterized with $\bar{w}_{\mathrm{noise}} = \sum_{i,t} \mathbb{I}(\tilde{y}_{i,t} = y_{i,t}) w_{i,t} / (\sum_{i,t} \mathbb{I}(\tilde{y}_{i,t} \neq y_{i,t}) \alpha_t)$, where $w_{i,t}$ denotes the weight of i-th sample of the batch at t-th training step. $\bar{w}_{\mathrm{clean}}$ for clean samples can be defined similarly. As their relative difference can be indicated by the ratio $\mathrm{snr} = \bar{w}_{\mathrm{clean}} / \bar{w}_{\mathrm{noise}}$, we report the drop in test accuracy and snr on CIFAR10 under symmetric label noise in Table 4. Losses with less performance drop under label noise has higher snr in general.

To explain what leads to a high snr, we first examine how Δ_y distributions of noisy and clean samples evolve during training on CIFAR10 in Fig. 4. When trained with robust losses (Fig. 4b and 4c), the noisy and clean distributions of Δ_y gets better separated and more spread. In addition, Δ_y's of some noisy samples gets decreased, suggesting that noisy samples can be *unlearnt*. In contrast, with CE (Fig. 4a), the noisy and clean distributions of Δ_y are less separated and more compact.

We now give a possible explanation for Fig. 4 with the following two training dynamics: **(D1) clean samples are learnt faster than noisy samples**; **(D2) noisy samples can be unlearnt when trained on clean samples.** D1 is identified in [7, 32] and later manifests in curriculum-based robust training

Loss	Hyperparams	Clean Acc	$\eta = 0.2$ Δ_{acc}	snr	$\eta = 0.4$ Δ_{acc}	snr	$\eta = 0.6$ Δ_{acc}	snr	$\eta = 0.8$ Δ_{acc}	snr
CE		90.49	-15.85	0.39	-32.34	0.58	-51.57	0.77	-71.14	0.95
SCE	$q = 0.95$	91.06	-8.10	0.76	-21.55	1.03	-43.86	1.29	-71.10	1.32
GCE	$q = 0.9$	90.85	-2.02	3.25	-5.59	3.16	-14.16	2.95	-50.10	2.29
MAE		90.56	-1.96	3.46	-8.25	3.15	-12.31	2.88	-38.11	2.53
AUL	$a = 7, q = 0.5$	90.79	-1.90	3.51	-5.06	3.40	-13.43	3.01	-50.99	1.79
AGCE	$a = 3, q = 1.2$	90.56	-4.28	3.11	-4.47	3.29	-17.76	2.69	-44.87	2.04

Table 4: Robust losses assign larger weights to clean samples. We report snr and drop in test accuracy with symmetric label noise on CIFAR10 at the final epoch with 3 different runs. We omit standard deviation due to space limitation. Hyperparameters of losses are tuned with noise rate $\eta = 0.6$.



(a) CE (b) SCE: $a = 0.05, b = 0.95$ (c) MAE

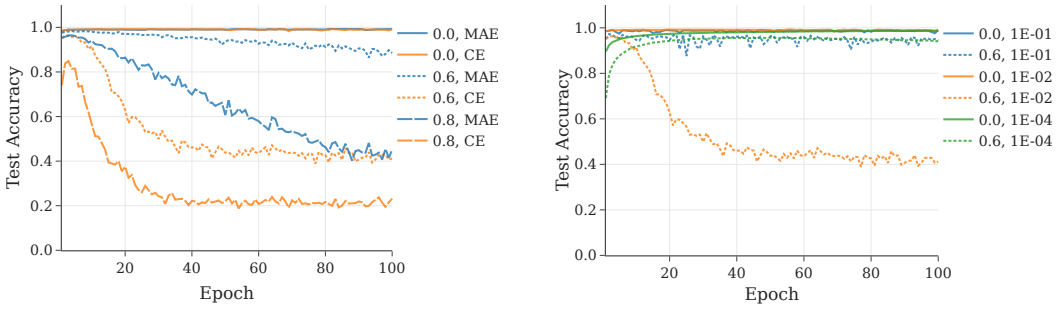
Figure 4: How Δ_y distribution of clean and noisy samples evolve during training with 40% symmetric label noise on CIFAR10. The $w(\Delta_y)$ curve for each loss is included. The same y-axises denoting probability density are omitted for brevity. Note that SCE is a weighted average of CE and MAE.

[33, 34]. It can result from the dominance of clean samples ($\tilde{r} < 1$) in the expected gradient. In addition, gradients of clean samples are more correlated than those of noisy samples [35]. Thus performance on clean samples can be improved when training on one another, leading to D1. D2 only become apparent when examining Fig. 4b and 4c, which can result from generalization with clean samples. Suppose in MNIST, a sample of 0 is erroneously labeled as 9. Then a model well-trained with clean samples of class 9 and 0 can result in a low Δ_y for this noisy sample. D1 and D2 can act in synergy to separate the clean and noisy distributions of Δ_y, as shown in Fig. 4.

We hypothesis that robust losses enhance the synergy of D1 and D2. In Table 1, $w(\Delta_y)$ of losses can be decomposed into $f(\Delta_y) \cdot g(\Delta_y)$, where $f(\Delta_y)$ is a monotonically increasing function and $g(\Delta_y)$ a decreasing one. For example, $f_{\text{CE}}(\Delta_y)$ degenerates to constant 1 and $g_{\text{CE}}(\Delta_y) = 1 - p_y$, while $f_{\text{MAE}}(\Delta_y) = p_y$ and $g_{\text{MAE}}(\Delta_y) = 1 - p_y$. Notably, $g(\Delta_y)$ shared by all losses converge to 0 as Δ_y increases, preventing Δ_y from growing infinitely large. In addition, **a non-degenerated $f(\Delta_y)$ can enhance the synergy between D1 and D2**. Since the initial Δ_y distribution generally lies on the monotonically increasing part of $w(\Delta_y)$ determined by $f(\Delta_y)$, faster learning of samples results in their larger weights during training. Thus robust losses **magnify the difference in learning speed** between clean and noisy samples, which can also account for the substantially spread Δ_y distributions in Fig. 4b and 4c. As $w(\Delta_y)$ can assign negligible sample weights with low Δ_y due to the monotonically increasing $f(\Delta_y)$, **unlearnt noisy samples are neglected** with diminishing weights, which can account for the decrease of Δ_y's for noisy samples in Fig. 4b and 4c. In contrast, as $w_{\text{CE}}(\Delta_y)$ assign high sample weights for small Δ_y's, it compensates the synergy of D1 and D2, thus results in compact Δ_y distribution, larger Δ_y's for noisy samples, and less separated Δ_y distributions in Fig. 4a.

With sufficient training, clean samples will eventually have high Δ_y's with diminishing sample weights thanks to $g(\Delta_y)$. Noisy samples then dominate the expected gradient and can lead to overfitting, leading to two unexpected results when viewed from robustness conditions [9, 11]:

Robust losses are vulnerable to label noise with extended training. In Fig. 5a we show the learning curve of CE and MAE using *constant* learning rate under different symmetric noises on MNIST.



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

(b) CE with different (η, α).

Figure 5: Learning curves with fixed learning rate and extended training epochs ($\times 2$ compared to the standard setting) on MNIST, where α is the learning rate and η is the noise rate.

Although enjoying theoretically guaranteed noise robustness [9, 11], similar to CE, MAE eventually overfits to noisy samples, becoming vulnerable to label noise as weights of clean samples diminish.

Losses can become robust by adjusting the learning rate schedule. Interestingly, in Fig. 4a, despite the compensation of $w_{\mathrm{CE}}(\Delta_y)$, the synergy between D1 and D2 still results in partially-separated Δ_y distributions of noisy and clean samples. We can thus improve the noise robustness of CE by preventing the weights of clean samples from diminishing due to $g(\Delta_y)$, which can be achieve by slowing down the convergence or early stopping [36]. In Fig. 5b we show the learning curve of CE using fixed learning rates under symmetric noise on MNIST. By simply increasing or decreasing the learning rate, which strengthens the implicit regularization of SGD [37] 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 [8–13] for robust training with noisy labels [1]. Theoretical results [9, 11] derive sufficient conditions for robustness against label noise without considering the training dynamics. We complement these results by considering the interaction between robust losses and various training dynamics. The underfitting of robust losses has been heuristically mitigated with loss combination [10, 12, 13]. We further elucidate the cause of underfitting from a curriculum perspective, based on which we provide an effective solution.

Curriculum-based approaches combat label noise with either sample selection [19, 20] or sample-weighting [16–18]. In particular, sample weights are designed [14–16] or predicted by a model trained on a separated dataset [17, 18]. In contrast, the sample-weighting curriculums considered in this work are implicitly induced by robust loss functions. Most related to our work, Wang et al. [14] identifies gradient norms as weights for sample gradients of each robust loss. In contrast, as discussed in §3.1, we explicitly extract the implicit loss, which helps draw the connection to standard curriculum learning [22] and facilitates analysis of training dynamics.

Our work is also related to the ongoing debate [22, 38] on strategies for selecting or weighting samples in curriculum learning: whether easier first [21, 23] or harder first [24, 39] is better. The implicit curriculums of robust losses in this work differ in two important ways. First, the implicit loss identified in §3.1 more directly measures sample difficulty than loss value [23, 24] and gradient magnitude [25]. Second, the implicit sample-weighting curriculums can be viewed as a combination of both weighting strategies by emphasizing moderately difficult samples, as discussed in §4.3.

6 Conclusion

We identify the implicit sample-weighting curriculums of selected loss functions. The resulting implicit loss as a direct sample performance metric and sample weight specifying the implicit preference of each loss function can help analyze and design robust loss functions and curriculums. Following such a curriculum perspective, we elucidate the reasons behind underfitting and robustness against label noise for existing robust loss functions, and design a simple approach to address the underfitting issue. Notably, our curriculum perspective connects robust loss functions to the seemingly distinct approaches based on curriculum learning.

References

[1] 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.

[2] 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.

[3] 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*, pages 74–79, Uppsala, Sweden, 2010. Association for Computational Linguistics. URL https://aclanthology.org/P10-2014.

[4] 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.

[5] 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*, pages 3437–3445, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.269. URL https://aclanthology.org/2021.naacl-main.269.

[6] 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*, pages 74–83, Melbourne, Australia, 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-2709. URL https://aclanthology.org/W18-2709.

[7] 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.

[8] N. Manwani and P. S. Sastry. Noise tolerance under risk minimization. *IEEE Transactions on Cybernetics*, 43(3):1146–1151, 2013. doi: 10.1109/tsmcb.2012.2223460. URL https://doi.org/10.1109%2Ftsmcb.2012.2223460.

[9] 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, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 1919–1925. AAAI Press, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14759.

[10] 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*, pages 6543–6553. PMLR, 2020. URL http://proceedings.mlr.press/v119/ma20c.html.

[11] 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, editors, *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*, pages 12846–12856. PMLR, 2021. URL http://proceedings.mlr.press/v139/zhou21f.html.

[12] 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, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*,

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

[13] 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*, pages 322–330. IEEE, 2019. doi: 10.1109/ICCV.2019.00041. URL https://doi.org/10.1109/ICCV.2019.00041.

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

[15] 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, 2019. URL https://arxiv.org/abs/1903.12141.

[16] 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, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1002–1012, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/2f37d10131f2a483a8dd005b3d14b0d9-Abstract.html.

[17] 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, editors, *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*, pages 2309–2318. PMLR, 2018. URL http://proceedings.mlr.press/v80/jiang18c.html.

[18] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In Jennifer G. Dy and Andreas Krause, editors, *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*, pages 4331–4340. PMLR, 2018. URL http://proceedings.mlr.press/v80/ren18a.html.

[19] 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, 2021. URL https://openreview.net/forum?id=lmTWnm3coJJ.

[20] 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, editors, *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*, pages 1062–1070. PMLR, 2019. URL http://proceedings.mlr.press/v97/chen19g.html.

[21] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman, editors, *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*, pages 41–48. ACM, 2009. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

[22] 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.

[23] 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, editors, *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*, pages 1189–1197. Curran Associates, Inc., 2010. URL https://proceedings.neurips.cc/paper/2010/hash/e57c6b956a6521b28495f2886ca0977a-Abstract.html.

[24] 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.

[25] Siddharth Gopal. Adaptive sampling for SGD by exploiting side information. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *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*, pages 364–372. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/gopal16.html.

[26] 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.

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

[28] 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*, pages 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

[29] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis R. Bach and David M. Blei, editors, *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*, pages 448–456. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/ioffe15.html.

[30] 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*, page 249–256. JMLR Workshop and Conference Proceedings, 2010. URL https://proceedings.mlr.press/v9/glorot10a.html.

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

[32] 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, editors, *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*, pages 233–242. PMLR, 2017. URL http://proceedings.mlr.press/v70/arpit17a.html.

[33] 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.

[34] 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, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8536–8546, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/a19744e268754fb0148b017647355b7b-Abstract.html.

[35] 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.

[36] 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.

[37] 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.

[38] Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *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*, pages 2535–2544. PMLR, 2019. URL http://proceedings.mlr.press/v97/hacohen19a.html.

[39] 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.

[40] 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*, pages 2233–2241. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.240. URL https://doi.org/10.1109/CVPR.2017.240.

[41] 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, editors, *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*, pages 1353–1360. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2006/hash/532b7cbe070a3579f424988a040752f2-Abstract.html.

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

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] In §3.1 we state that the curriculum view is valid when considering the first order derivatives. We also analyze the exception with NCE in §3.2.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] In §4.2 we explicitly state the assumed distributions of s_j when deriving $\mathbb{E}[\Delta_y]$.
   (b) Did you include complete proofs of all theoretical results? [Yes] In Appendix B we include the detailed derivations of $\|\nabla_{\boldsymbol{s}}\Delta_y\|_1$ and w_{NCE}; in Appendix E we include the detailed derivation of $\mathbb{E}[\Delta_y]$.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] All default settings are in Appendix C, specific hyperparameters deviation from default settings are stated near each result.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Tables 2 to 4, 5 and 6. We omit error bars for figures to improve readability.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Appendix C

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] At the beginning of §4, we cite MNIST, CIFAR10/100.

(b) Did you mention the license of the assets? [N/A] MNIST and CIFAR10/100 are classic benchmarks

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Detailed Review of Selected Losses in §2.2

The standard **Cross Entropy (CE)**

$$L_{\mathrm{CE}}(\boldsymbol{s}, y) = -\log p_y \tag{12}$$

is neither symmetric [9] or asymmetric [11], thus can be vulnerable to label noise.

A.1 Symmetric Losses

The classic **Mean Absolute Error (MAE)** proposed by Ghosh et al. [9] is

$$L_{\mathrm{MAE}}(\boldsymbol{s}, y) = \frac{1}{K}\sum_j |\mathbb{I}(j = y) - p_j| = 2 - 2p_y \propto 1 - p_y$$

where $\mathbb{I}(j = y)$ is the indicator function.

Reverse Cross Entropy (RCE) [13] is

$$L_{\mathrm{RCE}}(\boldsymbol{s}, y) = \sum_j p_j \log \mathbf{1}(j = y) = \sum_{j \neq y} A p_j = A(1 - p_y) \propto 1 - p_y = L_{\mathrm{MAE}}(\boldsymbol{s}, y)$$

where $\log 0$ is truncated to a constant A to avoid numerical overflow. Thus implementation-wise, RCE is equivalent to MAE.

Ma et al. [10] argued that any generic losses with $L(\boldsymbol{s}, j) > 0, \forall j \in \{1, \ldots, K\}$ can become symmetric by simply normalizing them. As an example, the **Normalized Cross Entropy (NCE)** is

$$L_{\mathrm{NCE}}(\boldsymbol{s}, y) = \frac{L_{\mathrm{CE}}(\boldsymbol{s}, y)}{\sum_j L_{\mathrm{CE}}(\boldsymbol{s}, j)} = \frac{-\log p_y}{\sum_j -\log p_j} \tag{13}$$

A.2 Asymmetric Losses

Asymmetric Generalized Cross Entropy (AGCE) is

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

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) is

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

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

A.3 Combined Losses

Losses can be combined in different ways. For example, **Generalized Cross Entropy (GCE)** [12]

$$L_{\mathrm{GCE}}(\boldsymbol{s}, y) = \frac{1 - p_i^q}{q} \tag{14}$$

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 $q = 1$.

Alternatively, **Symmetric Cross Entropy (SCE)** [13] is a weighted average of CE and RCE (MAE):

$$L_{\mathrm{SCE}}(\boldsymbol{s}, y) = a \cdot L_{\mathrm{CE}}(\boldsymbol{s}, y) + b \cdot L_{\mathrm{RCE}}(\boldsymbol{s}, y) \propto (1-q) \cdot (-\log p_i) + q \cdot (1 - p_i) \tag{15}$$

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

Finally, Ma et al. [10] propose to combine active and passive robust losses. Here we review NCE+MAE as an example:

$$L_{\mathrm{NCE+MAE}}(\boldsymbol{s}, y) = a \cdot L_{\mathrm{NCE}}(\boldsymbol{s}, y) + b \cdot L_{\mathrm{MAE}}(\boldsymbol{s}, y) \propto (1-q) \cdot \frac{-\log p_y}{\sum_j -\log p_j} + q \cdot (1 - p_y)$$

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

B Additional Derivations for §3.2

Constant Norm of $\|\nabla_{\boldsymbol{s}} \Delta_y\|_1$: Since

$$\frac{\partial \Delta_y}{\partial s_j} = \begin{cases} 1, & j = y \\ -\frac{e^{s_j}}{\sum_{k \neq y} e^{s_k}} = \frac{p_j}{1 - p_y}, & j \neq y \end{cases}$$

then

$$\|\nabla_{\boldsymbol{s}} \Delta_y\|_1 = \sum_j |\frac{\partial \Delta_y}{\partial s_j}| = 1 + \sum_{j \neq y} \frac{e^{s_j}}{\sum_{k \neq y} e^{s_k}} = 1 + 1 = 2$$

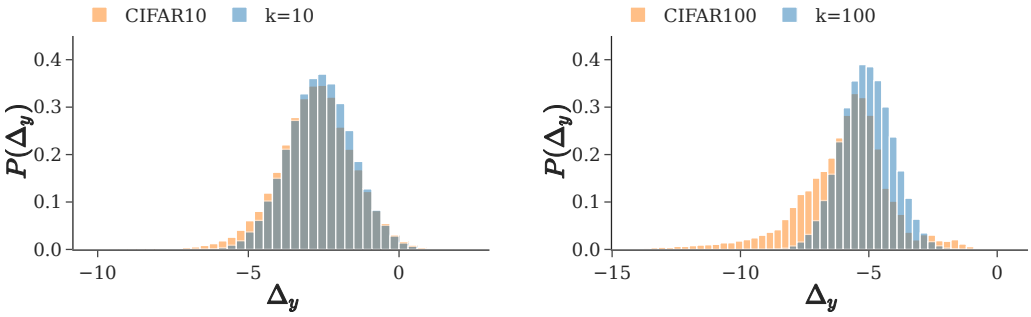
Approximating w_{NCE} in Eq. (8):

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

The derivation is based on the inequality $|x \pm y| \leq |x| + |y|$, following the intuition [14, 15] that $\|\nabla_{\boldsymbol{s}} L_{\mathrm{NCE}}(\boldsymbol{s}, y)\|_1$ can be regarded as sample weights.

Underfitting	Loss	Hyperparams	CIFAR100 Acc.	CIFAR100 $\bar{\alpha}_t^*$	CIFAR10 Acc.	CIFAR10 $\bar{\alpha}_t^*$
No	CE		68.76 ± 0.21	0.962	90.24 ± 0.14	0.624
No	GCE	$q = 0.3$	69.00 ± 0.24	0.956	90.83 ± 0.20	0.644
	SCE	$q = 0.7$	68.89 ± 0.05	1.165	91.07 ± 0.09	0.726
	NCE+MAE	$q = 0.3$	68.21 ± 0.51	0.520	90.14 ± 0.09	0.344
Moderate	NCE		57.95 ± 0.26	0.330	85.96 ± 0.21	0.206
	AUL	$a = 1.1, q = 5.0$	47.98 ± 3.48	0.485	88.94 ± 0.29	0.604
	AGCE	$a = 0.1, q = 0.1$	43.51 ± 2.58	0.406	90.71 ± 0.19	0.549
Severe	MAE		9.11 ± 0.83	0.025	90.65 ± 0.10	0.355
	AUL	$a = 3.0, q = 0.7$	10.04 ± 2.33	0.023	90.77 ± 0.04	0.337
	AGCE	$a = 1.6, q = 2.0$	5.34 ± 0.67	0.008	81.59 ± 8.55	0.243

Table 5: Similar results as Table 2 except with learning rate $\alpha = 0.01$. Robust losses can underfit regardless of hyperparameters of training.



(a) CIFAR10 (b) CIFAR100

Figure 6: Comparing simulated and real Δ_y distributions at initialization. (a) CIFAR10 and $k = 10$; (b) CIFAR100 and $k = 100$.

C Detailed Experimental Settings

Our settings are mainly identical to [10, 11], with differences explicitly stated in the main text. Here we describe the default settings. All experiments are trained on NVIDIA 2080ti gpus with FP32.

Noise generation The noisy labels are generated following [10, 11, 40]. 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 CIFAR-100, 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 $\eta \in \{0.1, 0.2, 0.3, 0.4\}$.

Models and Training We use a 4-layer CNN for MNIST, an 8-layer CNN for CIFAR10 and a ResNet-34 [28] for CIFAR100, all with batch normalization [29]. Data augmentation including random width/height shift and horizontal flip are applied to CIFAR10/100. Without further specifications, all models are trained using SGD with momentum 0.9 for 50, 120 and 200 epochs on MNIST, CIFAR10 and CIFAR100, respectively. Learning rates with cosine annealing are 0.01 on MNIST and CIFAR10, and 0.1 on CIFAR100. Weight decays are 10^{-3} on MNIST, 10^{-4} on CIFAR10 and 10^{-5} on CIFAR100. Notably, all losses are normalized to have unit maximum in sample weights, which is different from [10].

D Additional Results for §4.1

Underfitting of robust losses with smaller learning rate. We report similar results as Table 2 with smaller learning rates in Table 5. Although settings that severe underfits slightly improve, the performance gap compared to CE is still substantial. Such results further confirms that underfitting results from robust losses themselves.

Loss	Clean $\eta = 0$	Symmetric Noise (Noise Rate η) $\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$	$\eta = 0.4$
CE [11]	71.33 ± 0.43	64.85 ± 0.37	58.11 ± 0.32	50.68 ± 0.55	40.17 ± 1.31
GCE [11]	63.09 ± 1.39	63.01 ± 1.01	59.35 ± 1.10	53.83 ± 0.64	40.91 ± 0.57
NCE [11]	29.96 ± 0.73	27.59 ± 0.54	25.75 ± 0.50	24.28 ± 0.80	20.64 ± 0.40
NCE+AUL [11]	68.96 ± 0.16	66.62 ± 0.09	63.86 ± 0.18	50.38 ± 0.32	38.59 ± 0.48
MAE	4.25 ± 0.41	3.59 ± 0.56	3.19 ± 0.98	2.11 ± 1.93	2.53 ± 1.34
MAE-shift	68.65 ± 0.46	64.70 ± 0.65	54.36 ± 1.20	48.38 ± 1.06	41.26 ± 0.12
MAE-rescale	69.03 ± 0.67	68.01 ± 1.08	65.71 ± 0.47	57.40 ± 0.35	39.22 ± 1.54

Table 6: Results of shifted and rescaled MAE with asymmetric label noise on CIFAR100. For comparison we include results reproduced by Zhou et al. [11].

E Additional Results and Derivation for §4.2

Simulated initial Δ_y approximate real dataset well. As shown in Fig. 6 simulated distributions approximate CIFAR10 and CIFAR100 well, which supports the analysis with Fig. 3.

Deriving $\mathbb{E}(\Delta_y)$ in Eq. (9) :

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

where $\approx_1$ follows $\mathbb{E}[\log X] \approx \log \mathbb{E}[X] - \frac{\mathbb{V}[X]}{2\mathbb{E}[X]^2}$ [41], $=_2$ utilize properties of sum of log-normal variables [42], and $=_3$ substitutes the expression of $\mathbb{E}[e^{s_y}]$ and $\mathbb{V}[e^{s_y}]$ for log-normal distributions.

Results of shifted and rescaled MAE with asymmetric label noise. In addition to Table 3, we also report results with asymmetric label noise in Table 6. Shift and rescale Δ_y also mitigate underfitting of MAE with asymmetric label noise, resulting in performance similar to the state-of-the-arts.