NOISE AGAINST NOISE: STOCHASTIC LABEL NOISE HELPS COMBAT INHERENT LABEL NOISE

Anonymous authors
Paper under double-blind review

ABSTRACT

The noise in stochastic gradient descent (SGD) provides a crucial implicit regularization effect, previously studied in optimization by analyzing the dynamics of parameter updates. In this paper, we are interested in learning with noisy labels, where we have a collection of samples with potential mislabeling. We show that a previously rarely discussed SGD noise, induced by stochastic label noise (SLN), mitigates the effects of inherent label noise. In contrast, the common SGD noise directly applied to model parameters does not. We formalize the differences and connections of SGD noise variants, showing that SLN induces SGD noise dependent on the sensitivity of model parameters and the confidence of model outputs. SLN not only improves generalization in its simplest form but also boosts popular robust training methods, including sample-selection and label-correction. Specifically, we present an enhanced algorithm by applying SLN to label-correction.

1 INTRODUCTION

The existence of label noise is a common issue in classification since real-world samples unavoidably contain some noisy labels, resulting from annotation platforms such as crowdsourcing systems (Yan et al., 2014). In the canonical setting of learning with noisy labels, we collect samples with potential mislabeling, but we do not know which samples are mislabeled since true labels are unobservable. It is troubling that overparameterized Deep Neural Networks (DNNs) have sufficient capacity to memorize noise (Zhang et al., 2017), leading to poor generalization performance. Thus, we are urgent for robust training methods that can mitigate the effects of label noise.

The noise in stochastic gradient descent (SGD) (HaoChen et al., 2020) provides a crucial implicit regularization effect for training overparameterized models. SGD noise is previously studied in optimization by analyzing the dynamics of parameter updates, whereas its utility in learning with noisy labels has not been explored to the best of our knowledge. In this paper, we find that the common SGD noise directly applied to model parameters does not endow much robustness, whereas a variant induced by controllable label noise does. *Interestingly, inherent label noise is harmful to generalization, while we can mitigate its effects using additional controllable label noise.* To prevent confusion, we use stochastic label noise (SLN) to indicate the label noise we introduce. Inherent label noise is biased and unknown, fixed when the data is given. SLN is mean-zero and independently sampled for each instance in each training step. Our main contributions are as follows.

- We formalize the differences and connections of three SGD noise variants (Proposition 1-3), and show that SLN induces SGD noise that is dependent on the sensitivity of model parameters and the confidence of model outputs.
- We approximate the perturbation on the output induced by the SGD noise (Theorem 1), and analyze the effects. We empirically show that SLN is superior than label smoothing in terms of preventing overconfidence.
- We empirically show that SLN not only improves generalization in its simplest form but also boosts popular robust training methods, including sample-selection and label-correction. We present an enhanced algorithm by applying SLN to label-correction.

In Fig. 1, we present a quick comparison between models trained with/without SLN on CIFAR-10 with symmetric/asymmetric/instance-dependent/open-set label noise. Throughout this paper, we use

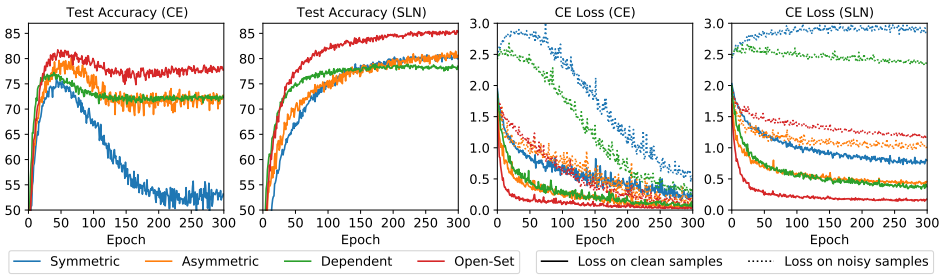

Figure 1: Test accuracy and training loss, averaged in 5 runs. For the loss, we separately show the average loss on clean and noisy samples.

CE to indicate a model trained with cross-entropy (CE) loss without any robust learning techniques, and the canonical CE loss is used by default for methods like SLN. In Section 4, we will provide more experimental details and more results that comprehensively verify the robustness of SLN on different synthetic noise and real-world noise. Here, the test curves in Fig. 1 show that SLN avoids the drop of test accuracy, with converged test accuracy even higher than the peak accuracy of the model trained with CE. The right two subplots in Fig. 1 show the average loss on clean and noisy samples. When trained with CE, the model eventually memorizes noise, indicated by the drop of average loss on noisy samples. In contrast, SLN largely avoids fitting noisy labels, and is useful for different types of noise.

2 Related Works

2.1 SGD Noise in Optimization

The SGD noise (HaoChen et al., 2020; Wen et al., 2019; Keskar et al., 2016) has long been studied in optimization. It is believed to provide a crucial implicit regularization effect (HaoChen et al., 2020; Arora et al., 2019; Soudry et al., 2018) for training overparameterized models. The most studied SGD noise is the spherical Gaussian noise directly added to model parameters in each training step (Ge et al., 2015; Neelakantan et al., 2015; Mou et al., 2018). However, empirical studies (Wen et al., 2019; Shallue et al., 2019) demonstrate that parameter-dependent SGD noise induced by label perturbations is far more effective. HaoChen et al. (2020) theoretically investigate the regularization effect on a quadratically-parameterized model introduced by Vaskevicius et al. (2019) and Woodworth et al. (2020). They prove that in an over-parameterized regression setting, SGD with label perturbations recovers the sparse groundtruth, whereas SGD with Gaussian noise directly added on gradient descent overfits to dense solutions. In the deep learning scenario, HaoChen et al. (2020) present primary empirical results for a VGG19 model (Simonyan & Zisserman, 2014) on CIFAR-100, showing that SGD noise - induced by Gaussian noise on the partial gradient of the loss to the model's output - avoids performance degeneration of large-batch training. To the best of our knowledge, the utility of SGD noise in learning with noisy labels has not been studied.

2.2 Robust Training Methods

Mitigating the effects of label noise is a vital topic in classification, which has a long history (Ekholm & Palmgren, 1982; Natarajan et al., 2013) and attracts much recent interest with several directions explored. **1)** Malach & Shalev-Shwartz (2017); Han et al. (2018b); Yu et al. (2019); Chen et al. (2019a); Wei et al. (2020) propose *sample-selection* methods that train on trusted samples, identified according to training loss, cross-validation or (dis)agreement between two models. **2)** Liu & Tao (2015); Jiang et al. (2018); Ren et al. (2018); Shu et al. (2019); Li et al. (2019a) develop *sample-weighting* schemes that aim to add higher weights on clean samples. **3)** Reed et al. (2014); Sukhbaatar et al. (2015); Patrini et al. (2017); Han et al. (2018a) apply *loss-correction* based on an estimated noise transition matrix. **4)** Tanaka et al. (2018); Arazo et al. (2019); Zheng et al. (2020) propose *label-correction* based on the model's predictions. **5)** Ghosh et al. (2017); Zhang & Sabuncu (2018); Xu et al. (2019); Charoenphakdee et al. (2019); Lyu & Tsang (2020); Ma et al. (2020) study *robust loss* functions that have a theoretical guarantee for noisy risk minimization,

typically with the assumption that the noise is class-conditional (Scott et al., 2013; Natarajan et al., 2013). **6)** Chen et al. (2019b); Menon et al. (2020); Hu et al. (2020); Harutyunyan et al. (2020); Lukasik et al. (2020) apply *regularization* techniques to improve generalization under label noise, including explicit regularizations such as manifold regularization (Belkin et al., 2006) and virtual adversarial training (Miyato et al., 2018), and implicit regularizations such as dropout (Srivastava et al., 2014), temporal ensembling (Laine & Aila, 2017), gradient clipping (Pascanu et al., 2012; Zhang et al., 2019; Menon et al., 2020) and label smoothing (Szegedy et al., 2016). **7)** One can combat label noise with *refined training strategies* (Li et al., 2019b; 2020; Nguyen et al., 2020) that potentially incorporate several techniques, including sample-selection/weighting, label-correction, meta-learning (Li et al., 2019b) and semi-supervised learning (Tarvainen & Valpola, 2017; Berthelot et al., 2019). Among these methods, regularization techniques are closely related to the essence of training networks, and studying robustness under label noise provides a new lens of understanding the regularization apart from the optimization lens.

3 METHOD

3.1 THE DIFFERENCES AND CONNECTIONS OF SGD NOISE VARIANTS

In this section, we present notations and definitions of three three SGD noise variants, and analyze the differences and connections between them. Notably, we show that SLN induces SGD noise dependent on the sensitivity of model parameters and the confidence of model predictions.

Notations. Let $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$ be a dataset with noisy labels. For each sample (x, y), its label y may be incorrect and the true label is unobservable. Let $f(x;\theta)$ be the neural network model with trainable parameter $\theta \in \mathbb{R}^p$. For a c-class classification problem, we have $f(x;\theta) \in \mathbb{R}^c$. We use a softmax function $S(f(x;\theta)) \in \mathbb{R}^c$ to obtain the probability of each class. The loss on a sample is denoted as $\ell(f, y)$. For classification, we use the cross-entropy (CE) loss by default. In parameter updates, a sample contributes $\nabla_\theta \ell(f, y)$ to the gradient descent. With SGD noise, the model is trained with a noisy gradient $\tilde{\nabla}_\theta \ell(f, y)$. Following the standard notation of Jacobian matrix, we have $\nabla_\theta \ell \in \mathbb{R}^{1\times p}$, $\nabla_f \ell \in \mathbb{R}^{1\times c}$, $\nabla_\theta f \in \mathbb{R}^{c\times p}$, $\nabla_{\theta_i} f \in \mathbb{R}^c$ and $\nabla_\theta f_i \in \mathbb{R}^{1\times p}$.

Noise on the full gradient. The most common SGD noise is the spherical Gaussian noise directly added to the full gradient (Neelakantan et al., 2015) as follow,

$$\tilde{\nabla}_\theta \ell(f, y) = \nabla_\theta \ell(f, y) + \sigma_\theta z_\theta, \tag{1}$$

where $\sigma_\theta > 0$, $z_\theta \in \mathbb{R}^{1\times p}$ and $z_\theta \sim \mathcal{N}(0, I_{p\times p})$.

Noise on the partial gradient. Taking a step further, HaoChen et al. (2020) study SGD noise induced by label noise on a quadratically-parameterized regression model, whereas for classification, they add mean-zero noise to $\nabla_f \ell(f, y)$ as follow

$$\tilde{\nabla}_f \ell(f, y) = \nabla_f \ell(f, y) + \sigma_f z_f, \quad \text{and,} \quad \tilde{\nabla}_\theta \ell(f, y) = \tilde{\nabla}_f \ell(f, y) \cdot \nabla_\theta f, \tag{2}$$

where $\sigma_f > 0$, $z_f \in \mathbb{R}^{1\times c}$ and $z_f \sim \mathcal{N}(0, I_{c\times c})$.

Noise induced by SLN. We induce SGD noise by label perturbations, namely the stochastic label noise (SLN), which is independently sampled for each instance in each training step.

$$\tilde{\nabla}_\theta \ell(f, y) = \nabla_\theta \ell(f, y + \sigma_y z_y), \tag{3}$$

where $\sigma_y > 0$, $z_y \in \mathbb{R}^c$ and $z_y \sim \mathcal{N}(0, I_{c\times c})$. Here, $\sigma_y z_y$ is the SLN on the label y, where y itself may be inherently incorrect.

In above Eq. (1-3), the standard deviation σ is a hyper-parameter. Since the SGD noise we introduce is i.i.d for each sample, we can equivalently consider each single sample (x, y) separately in theoretical analysis. For convenience, we use f, S to indicate the model output $f(x;\theta)$ and the softmax output $S(f(x;\theta))$ on a sample. In the following propositions, we formalize the differences and connections of the SGD noise. The proofs are presented in Appendix A.

Proposition 1. *Compared with Eq. (1), Eq. (2) induces noise $z \sim \mathcal{N}(0, \sigma_f^2 M)$ on $\nabla_\theta \ell(f, y)$, s.t., $M \in \mathbb{R}^{p\times p}$ and $M_{i,j} = (\nabla_{\theta_i} f)^T \nabla_{\theta_j} f$, $\forall i, j \in \{1, \cdots, p\}$. Note that the standard deviation of noise on the i-th parameter θ_i is $\sigma_f \|\nabla_{\theta_i} f\|_2$, where $\|\cdot\|_2$ denotes the L_2 norm.*

Proposition 2. *For the cross-entropy loss, compared with Eq. (2), Eq. (3) induces noise* $z \sim \mathcal{N}(0, \sigma_y^2 M)$ *on* $\nabla_f \ell(f, y)$*, s.t.,* $M \in \mathbb{R}^{c \times c}$*,* $M_{i,i} = c(S_i - 1/c)^2 + (c-1)/c$*, and* $M_{i,j} = c \cdot S_i S_j - S_i - S_j$*, if* $i \neq j$*. Note that the standard deviation of noise on the* i*-th entry is* $\sigma_y \sqrt{c(S_i - 1/c)^2 + (c-1)/c}$*.*

Proposition 3. *For the cross-entropy loss, compared with Eq. (1), Eq. (3) induces noise* $z \sim \mathcal{N}(0, \sigma_y^2 M)$ *on* $\nabla_\theta \ell(f, y)$*, s.t.,* $M \in \mathbb{R}^{p \times p}$ *and* $M_{i,j} = (\frac{\nabla_{\theta_i} S}{S})^T \frac{\nabla_{\theta_j} S}{S}$*,* $\forall i, j \in \{1, \cdots, p\}$*, where* $\frac{\cdot}{S}$ *denotes element-wise division. Note that the standard deviation of noise on the* i*-th parameter* θ_i *is* $\sigma_y \left\| \frac{\nabla_{\theta_i} S}{S} \right\|_2$*.*

We provide intuitions on the propositions in the following. The term $\|\nabla_{\theta_i} f\|_2$ in Proposition 1 can be interpreted as the *sensitivity* of f to θ_i. Hence, Eq. (2) induces high noise on a parameter if it has a high impact on the model output. For Eq. (3) (SLN), Proposition 2 implies that the equivalent noise on the partial gradient is minimized if $S_i = 1/c$, and is maximized if $S_i = 1$. Proposition 3 shows that the equivalent noise on θ_i is dependent on the *sensitivity* of S to θ_i and the *confidence* of S. If S is *confident*, i.e., an entry approaches 1 and other entries approach 0, then $\sigma_y \|\nabla_{\theta_i} S / S\|_2$ will be very large since there are small numbers in the denominator.

3.2 The effects of SGD noise

Based on Proposition 1-3, we can approximate the perturbation on the output induced by the SGD noise, and analyze the effects. We empirically show that in terms of preventing overconfidence, SLN reduces confidence mostly on noisy samples. Considering a training step $\theta \to \theta + \delta \cdot \nabla_\theta \ell(f, y)$, let δ be the learning rate and z be the noise on $\nabla_\theta \ell(f, y)$. Since we typically take a small step, using the first-order Taylor approximation, we obtain the perturbation on f induced by the SGD noise,

$$\Delta f = f(x; \theta + \delta \cdot \nabla_\theta \ell(f, y) + \delta \cdot z) - f(x; \theta + \delta \cdot \nabla_\theta \ell(f, y)) \approx \delta \cdot \nabla_\theta f \cdot z^T. \tag{4}$$

Theorem 1. *Considering a training step on a single sample with cross-entropy loss, we quantify the perturbation on* f *resulting from SGD noise using the first-order Taylor approximation. The perturbation on the* i*-th entry of* f *is denoted as* $|\Delta f_i|$*,* $i \in \{1, 2, \cdots, c\}$*, then*

- *for Eq. (1), we have* $\mathbb{E}[|\Delta f_i|^2] = \delta^2 \sigma_\theta^2 \cdot \nabla_\theta f_i \cdot (\nabla_\theta f_i)^T$*;*
- *for Eq. (2), we have* $\mathbb{E}[|\Delta f_i|^2] = \delta^2 \sigma_f^2 \cdot \sum_{j=1}^c (\nabla_\theta f_i \cdot (\nabla_\theta f_j)^T)^2$*;*
- *for Eq. (3), we have* $\mathbb{E}[|\Delta f_i|^2] = \delta^2 \sigma_y^2 \cdot \sum_{j=1}^c (\nabla_\theta f_i \cdot (\overline{\nabla_\theta f_j})^T)^2$*.*

where $\overline{\nabla_\theta f_j} = \nabla_\theta f_j - \sum_{k=1}^c S_k \cdot \nabla_\theta f_k$ *is normalized by the weighted average of* $\{\nabla_\theta f_k\}_{k=1}^c$*.*

The proof is presented in Appendix A. Note that the training can not converge if $|\Delta f_i|$ is large. In this sense, all SGD noise variants encourage converging to a small $\nabla_\theta f_i$, i.e., to a point that the output is *robust* to the parameters. Eq. (2) further encourages $\{\nabla_\theta f_k\}_{k=1}^c$ to be *orthogonal*, otherwise there will be a high perturbation. Compared with Eq. (2), our SLN similarly encourages orthogonality. Additionally, there is a normalization in $\nabla_\theta f_j$. When the orthogonal condition has been satisfied, we have $\mathbb{E}[|\Delta f_i|^2] = \delta^2 \sigma_y^2 (c \cdot S_i^2 - 2S_i + 1) \cdot \nabla_\theta f_i \cdot (\nabla_\theta f_i)^T$, which is minimized if $S_i = 1/c$, maximized if $S_i = 1$, consistent with the finding of Proposition 2. Therefore, once the prediction is *overconfident*, there will be a high perturbation on the output.

In terms of preventing overconfidence, we shall discuss label smoothing (LS) (Lukasik et al., 2020). It smooths the given one-hot label y into a soft one $\tilde{y}$, s.t., $\tilde{y} = (1 - \alpha y) + \alpha e$, where e is an all-one vector and $\alpha > 0$ is a hyperparameter. In this way, LS introduces a fixed and biased perturbation on the label, whereas our SLN introduces dynamic and mean-zero perturbations. SLN adaptively perturbs the output once it is overconfident. In Fig. 2, we plot the sample density w.r.t. predictions on the labeled class, using CIFAR-10 with 40% symmetric noise as an example. It shows that SLN does reduce overconfidence, and it mostly affects noisy samples.

3.3 A detailed dissection on predictions of DNNs trained with SLN

In this section, we show that SLN boosts popular robust training methods, including sample-selection and label-correction. Many methods, demonstrated to work well, select or add higher

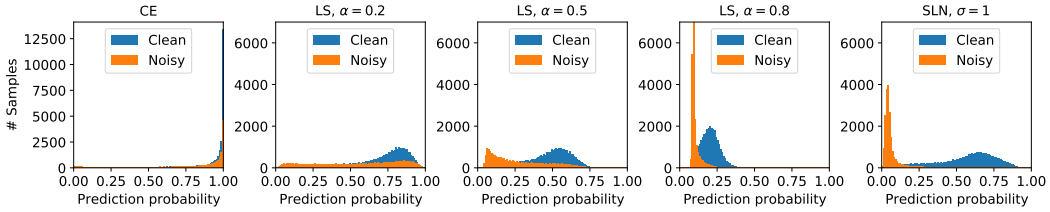

Figure 2: Samples density w.r.t. predictions (the softmax outputs on the labeled class). SLN reduces overconfidence mostly on noisy samples, while clean samples are less affected.

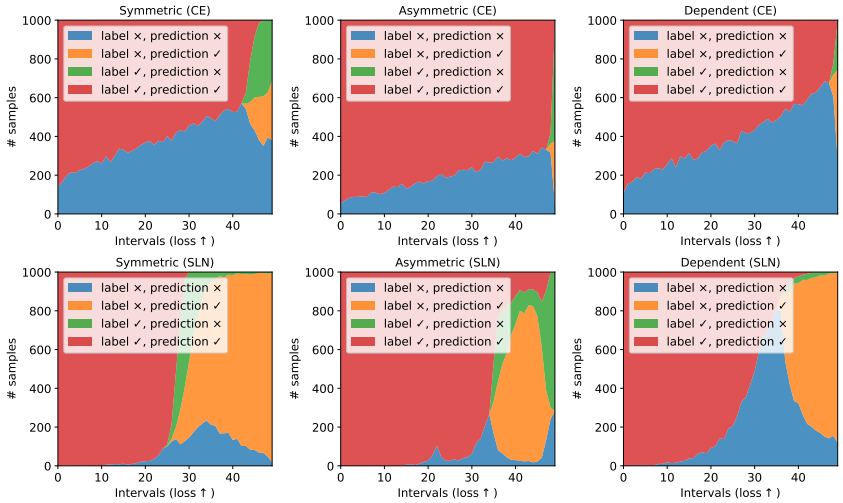

Figure 3: Training samples are sorted in ascending order of loss, uniformly divided into 1000 samples per interval, and dissected according to the correctness of the given label and the prediction.

weight on small-loss samples (Han et al., 2018b; Jiang et al., 2018; Li et al., 2020), or use the model's predictions to correct noisy labels (Tanaka et al., 2018; Arazo et al., 2019). A warm-up phase is usually required to initialize the model before sample-selection or label-correction, yet the model will memorize noise if the warm-up phase is too long. With SLN, we can simply train the model until convergence. In Fig. 3, we compare converged models on CIFAR-10, where the model is trained with SLN or merely CE loss. The detailed noise setting can be found in Section 4. We first sort training samples in ascending order of loss, then uniformly divide them into 1000 samples per interval, finally obtain the number of four types of samples in each interval based on the correctness of the given label and the prediction. Fig. 3 verifies the utility of SLN. For sample-selection, when trained with CE, there are many small-loss samples with incorrect labels (the blue region), while the model trained with SLN is much more reliable. For label-correction, the model trained with SLN yields correct predictions for many originally incorrect samples (the orange region).

As a concrete example, we present an algorithm by applying SLN to label-correction. With SLN, we train the model for sufficient epochs until convergence, without the need of carefully tuning a warm-up phase. Then we start label-correction using $y_{correction} = \omega \cdot y + (1 - \omega) \cdot S$, where S is the softmax prediction, $\omega \in [0, 1]$ is the weight obtained by normalizing the training loss, s.t., for the i-th training sample, $\omega_i = (\ell_i - \ell_{min})/(\ell_{max} - \ell_{min})$. More discussions on the label-correction strategies are presented in Appendix C.1.

4 Experiments

4.1 The overview of experimental setup

We comprehensively verify the utility of SLN on different types of label noise, including symmetric noise, asymmetric noise, instance-dependent noise and open-set noise synthesized on CIFAR-10 and

CIFAR-100, and real-world noise on Clothing1M (Xiao et al., 2015). **1)** Symmetric noise assumes each label has the same probability of flipping to any other class. We uniformly flip the label to other classes with an overall probability 40%. **2)** Asymmetric noise contains noisy labels flipped between similar classes. Following Zhang & Sabuncu (2018), on CIFAR-10, we flip labels between TRUCK→AUTOMOBILE, BIRD→AIRPLANE, DEER→HORSE, and CAT↔DOG, with a probability 40%; on CIFAR-100, we flip each class into the next class circularly with a probability 40%. **3)** Instance-dependent noise assumes that the mislabeling probability should be dependent on each instance's input features (Xia et al., 2020). We first train a model on clean data, then flip labels of 40% samples to their second-confident class in the predictions. **4)** Open-set noise contains samples that do not belong to any class considered in the classification task. Following Wang et al. (2018), we yield CIFAR-10 with open-set noise by randomly replacing 40% of its training images with images from CIFAR-100. **5)** For real-world noise, we use the large-scale benchmark Clothing1M, which contains $1M$ training images with noisy labels from online shops.

We use the backbone wide ResNet-28-2 (Zagoruyko & Komodakis, 2016) on CIFAR-10 and CIFAR-100, and ResNet-50 on Clothing1M. More training details can be found in Appendix B. Apart from SLN, we introduce a momentum (MO) model (Tarvainen & Valpola, 2017) and label-correction (LC), with which we get SLN-MO and SLN-MO-LC. Parameters of the momentum model (θ') is updated as a moving average of the the training model (θ). At t-th training step, $\theta'^{(t)} = \alpha\theta'^{(t-1)} + (1-\alpha)\theta^{(t)}$. We use $\alpha = 0.999$ in all experiments. For SLN-MO-LC, we have discussed in Section 3.3 that SLN is reliable when applied to label-correction.

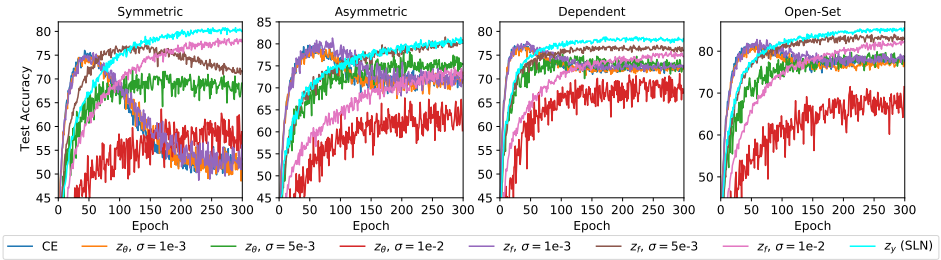

Figure 4: Performance of SGD noise variants on CIFAR-10. The accuracy is averaged in 5 runs.

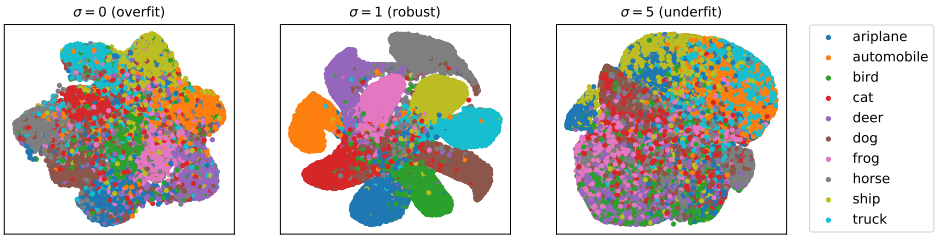

Figure 5: The t-SNE visualization of features on the model's penultimate layer.

4.2 THE EFFECTS OF SGD NOISE VARIANTS

We first show that compared with other variants, SLN stands out in mitigating the effects of label noise. The SGD noise variants have been formalized in Section 3.1, including noise on the full gradient z_θ, noise on the partial gradient z_f and noise induced by SLN (z_y). For SLN, the standard deviation is $\sigma = 1$ under symmetric label noise and $\sigma = 0.5$ in all other cases. For other SGD noise, the standard deviation is equally tuned, and the performance under three different σ is separately shown in Fig. 4. SLN significant improves the generalization, while other SGD noise variants can not achieve such impressive performance even if σ is heavily tuned. It is worth noting that for all these variants, when σ is too small, the model overfits noise since the training is similar to merely using CE loss; when σ is too large, the model fails to fit the training data since the SGD noise is too high. This finding provides a guideline for tuning σ. In Fig. 5, we show the t-SNE visualization of features on the model's penultimate layer, taking all training samples as input. We visualize the embedding under symmetric noise since it yields the most severe damage to the generalization and

Table 1: Test accuracy on the last epoch (mean±std in 5 runs) on CIFAR-10. The Open-Set noise is generated by randomly replacing 40% images of CIFAR-10 with images from CIFAR-100.

Method	Symmetric	Asymmetric	Dependent	Open-Set	*Training time*
CE	53.16±1.64	72.72±1.35	72.33±0.50	77.74±0.84	*1.18±0.03h*
GCE	79.27±0.54	70.90±2.98	73.05±0.24	81.56±0.41	*1.19±0.03h*
Co-Teaching	**82.37±0.32**	79.61±1.06	76.69±0.66	85.21±0.35	*2.22±0.06h*
PHuber-CE	80.58±0.34	73.01±1.28	72.70±0.24	81.62±0.52	*1.20±0.04h*
LS	66.59±1.25	66.96±2.02	72.79±0.27	77.06±0.37	*1.20±0.02h*
SLN	80.00±0.61	**80.63±0.61**	**78.48±0.28**	**85.33±0.52**	*1.24±0.05h*
SLN-MO	**84.71±0.43**	**84.80±0.27**	**80.56±0.16**	**89.16±0.31**	*1.30±0.03h*
SLN-MO-LC	**87.00±0.27**	**87.85±0.41**	**81.76±0.16**	**88.47±0.22**	*1.31±0.04h*

Table 2: Test accuracy on the last epoch (mean±std in 5 runs) on CIFAR-100.

Method	Symmetric	Asymmetric	Dependent	*Training time*
CE	31.63±0.83	36.26±0.65	55.59±0.43	*1.22±0.04h*
GCE	44.07±0.56	36.75±1.07	54.51±0.26	*1.23±0.05h*
Co-Teaching	**53.05±1.11**	39.78±0.52	47.06±0.44	*2.23±0.07h*
PHuber-CE	48.76±1.28	35.13±0.38	55.11±0.29	*1.29±0.04h*
LS	40.57±0.68	43.51±0.48	53.89±0.39	*1.23±0.03h*
SLN	50.24±0.41	**44.43±0.19**	**57.47±0.30**	*1.28±0.04h*
SLN-MO	**56.57±0.38**	**50.59±0.35**	**61.24±0.27**	*1.32±0.02h*
SLN-MO-LC	**58.64±0.37**	**63.90±0.46**	**61.14±0.39**	*1.35±0.03h*

SLN provides a significant improvement. Fig. 5 shows that the model trained with SLN can yield a better embedding. It also demonstrates overfitting and underfitting when σ is too small or too large.

4.3 CIFAR-10 AND CIFAR-100

On CIFAR-10 and CIFAR-100, we compare with the following baselines: 1) only cross-entropy (CE) loss, 2) Generalized Cross-Entropy (GCE) (Zhang & Sabuncu, 2018) loss, 3) Co-Teaching (Han et al., 2018b) that uses co-training and sample-selection, 4) PHuber-CE (Menon et al., 2020) that uses gradient clipping and 5) label-smoothing (LS) Lukasik et al. (2020) that clips the label to be less confident before training. We use $5k$ noisy samples as the validation to tune hyper-parameters, then train the model on the full training set and report the test accuracy at the last epoch. SLN simply requires tuning the standard deviation σ, which is tuned in $\{0.1, 0.2, 0.5, 1\}$. On CIFAR-10, the best σ is 1 under symmetric noise and 0.5 otherwise; On CIFAR-100, it is 0.1 under instance-dependent noise and 0.2 otherwise. The label-correction in SLN-MO-LC is applied in the last 50 epochs. The softmax prediction is converted into one-hot label in correction. We repeat each experiment 5 times. The average test accuracy at the last epoch is reported in Table 1 and Table 2,. We mark the top-3 results in bold. To compare the efficiency, we present the average training time of each method, evaluated on a single V100 GPU. Without the momentum model and label-correction, SLN itself achieves impressive test performance, which is consistent with results in Fig. 1. In Section 3.3, we have analyzed that SLN can boosts popular robust training methods, based on a detailed dissection on DNNs' predictions. As expected, we obtain further improvement with SLN-MO and SLN-MO-LC. Notably, SLN, SLN-MO and SLN-MO-LC sweep the top-3 results in almost all cases with low computational cost. In Fig. 6, we visualize the training efficiency and test accuracy. The figure clearly illustrates the superior generalization performance and high efficiency of our methods. To further verify our discussions that SLN can enhance existing methods, we present additional results in Appendix C.2. Specifically, we apply SLN to a practical baseline that combines many robust training strategies, and show that SLN not only eliminates the need of tuning warm-up phase but also improves the test performance.

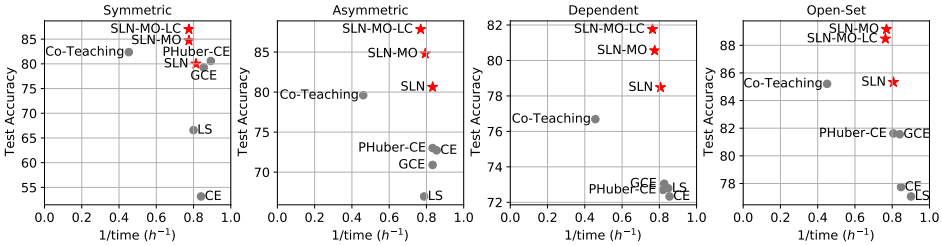

Figure 6: Test accuracy w.r.t. training efficiency (1/time) on CIFAR-10.

Table 3: Test accuracy (mean±std in 3 runs) on Clothing1M. The star* marks results copied from Patrini et al. (2017). The result of DivideMix (Li et al., 2020) is reproduced from its official implementation, which uses class-balanced training sampling.

Training sampling	Standard		Noisy-class-balanced	
Method	Test accuracy	*Training time*	Test accuracy	*Training time*
CE	68.94*	-	71.12±0.32	*2.61±0.08h*
Forward	69.84*	-	71.28±0.27	*2.74±0.05h*
Backward	69.13*	-	71.03±0.33	*2.72±0.09h*
Co-Teaching	70.19±0.28	*15.06±0.33h*	72.14±0.28	*4.51±0.17h*
DivideMix	-	-	73.81±0.41	*18.78±0.32h*
SLN	70.42±0.34	*9.25±0.12h*	72.95±0.31	*2.73±0.05h*
SLN-MO	71.15±0.21	*9.31±0.09h*	72.98±0.15	*2.78±0.07h*
SLN-MO-LC	**72.61±0.23**	*11.59±0.23h*	**74.08±0.18**	*3.31±0.03h*

4.4 CLOTHING1M

Clothing1M (Xiao et al., 2015) is a large-scale benchmark of clothing images from online shops with 14 classes, containing real-world label noise. It has 1 million noisy samples for training, $14k$ and $10k$ clean samples respectively for validation and test. We find that the number of images labeled as each class is unbalanced, ranging from 18976 to 88588 in the noisy training set. In previous works, some experiments are conducted by sampling a class-balanced training subset in each epoch (Li et al., 2020), while other benchmark experiments directly train on the full training set (Patrini et al., 2017). However, the balanced training sampling itself affects the test performance, which makes it difficult to compare results across papers. Therefore, we make it clear here and conduct experiments in both setting, including the standard sampling and the noisy-class-balanced sampling. For the latter, in each epoch, 18976 instances are randomly sampled from each labeled class. Other training details strictly follow the standard benchmark setting (Patrini et al., 2017), presented in Appendix B. We set the standard deviation of SLN as $\sigma = 0.2$. For SLN-MO-LC, the label-correction is applied since the first epoch. Results are listed in Table 3, with the best result in bold and previous published results marked by a star. The result of DivideMix is averaged in each run on the two model it trained, rather than using additional ensemble of two models. Our SLN outperforms many baselines. The variants SLN-MO and SLN-MO-LC further achieve higher test accuracy. Specifically, SLN-MO-LC achieves the best test accuracy in both settings. Our methods also stand out for training efficiency.

5 CONCLUSION

In this paper, we establish that SLN induces SGD noise dependent on the sensitivity of model parameters and the confidence of model outputs. We empirically verify the robustness of SLN under various synthetic label noise and real-world noise. Moreover, we show that SLN not only improves generalization in its simplest form but also boosts popular robust training methods, including sample-selection and label-correction. In particular, we justify that SLN can enhance existing methods based on a detailed dissection on DNNs' predictions, then present a practical algorithm by applying SLN to label-correction.

REFERENCES

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *ICML*, 2019.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In *NeurIPS*, 2019.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 2006.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019.

Nontawat Charoenphakdee, Jongyeong Lee, and Masashi Sugiyama. On symmetric losses for learning from corrupted labels. In *ICML*, 2019.

Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *ICML*, 2019a.

Pengfei Chen, Benben Liao, Guangyong Chen, and Shengyu Zhang. A meta approach to defend noisy labels by the manifold regularizer psdr. *arXiv preprint arXiv:1906.05509*, 2019b.

Anders Ekholm and Juni Palmgren. A model for a binary response with misclassifications. In *GLIM 82: Proceedings of the international conference on generalised linear models*, pp. 128–143. Springer, 1982.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *COLT*, 2015.

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

Bo Han, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Ivor Tsang, Ya Zhang, and Masashi Sugiyama. Masking: A new perspective of noisy supervision. In *NeurIPS*, 2018a.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018b.

Jeff Z HaoChen, Colin Wei, Jason D Lee, and Tengyu Ma. Shape matters: Understanding the implicit bias of the noise covariance. *arXiv preprint arXiv:2006.08680*, 2020.

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

Wei Hu, Zhiyuan Li, and Dingli Yu. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. In *ICLR*, 2020.

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 *ICML*, 2018.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017.

Jia Li, Yafei Song, Jianfeng Zhu, Lele Cheng, Ying Su, Lin Ye, Pengcheng Yuan, and Shumin Han. Learning from large-scale noisy web data with ubiquitous reweighting for image classification. *TPAMI*, 2019a.

Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Learning to learn from noisy labeled data. In *CVPR*, 2019b.

Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *ICLR*, 2020.

Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *TPAMI*, 2015.

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

Yueming Lyu and Ivor W Tsang. Curriculum loss: Robust learning and generalization against label corruption. In *ICLR*, 2020.

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

Eran Malach and Shai Shalev-Shwartz. Decoupling" when to update" from" how to update". In *NeurIPS*, 2017.

Aditya Krishna Menon, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Can gradient clipping mitigate label noise? In *ICLR*, 2020.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *TPAMI*, 2018.

Wenlong Mou, Liwei Wang, Xiyu Zhai, and Kai Zheng. Generalization bounds of sgld for non-convex learning: Two theoretical viewpoints. In *COLT*, 2018.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *NeurIPS*, 2013.

Arvind Neelakantan, Luke Vilnis, Quoc V Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach, and James Martens. Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*, 2015.

Tam Nguyen, C Mummadi, T Ngo, L Beggel, and Thomas Brox. Self: learning to filter noisy labels with self-ensembling. In *ICLR*, 2020.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. *CoRR, abs/1211.5063*, 2012.

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 *CVPR*, 2017.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, 2018.

Clayton Scott, Gilles Blanchard, and Gregory Handy. Classification with asymmetric label noise: Consistency and maximal denoising. In *COLT*, 2013.

Christopher J Shallue, Jaehoon Lee, Joseph Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E Dahl. Measuring the effects of data parallelism on neural network training. *JMLR*, 20: 1–49, 2019.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *NeurIPS*, 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *JMLR*, 19(1):2822–2878, 2018.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 2014.

Sainbayar Sukhbaatar, Joan Bruna Estrach, Manohar Paluri, Lubomir Bourdev, and Robert Fergus. Training convolutional networks with noisy labels. In *ICLR*, 2015.

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

Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, 2018.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017.

Tomas Vaskevicius, Varun Kanade, and Patrick Rebeschini. Implicit regularization for optimal sparse recovery. In *NeurIPS*, 2019.

Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. Iterative learning with open-set noisy labels. In *CVPR*, 2018.

Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *CVPR*, 2020.

Yeming Wen, Kevin Luk, Maxime Gazeau, Guodong Zhang, Harris Chan, and Jimmy Ba. Interplay between optimization and generalization of stochastic gradient descent with covariance noise. *arXiv preprint arXiv:1902.08234*, 2019.

Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. *arXiv preprint arXiv:2002.09277*, 2020.

Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Parts-dependent label noise: Towards instance-dependent label noise. *arXiv preprint arXiv:2006.07836*, 2020.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, 2015.

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

Yan Yan, Rómer Rosales, Glenn Fung, Ramanathan Subramanian, and Jennifer Dy. Learning from multiple annotators with varying expertise. *Machine learning*, 2014.

Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *ICML*, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference*, 2016.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Analysis of gradient clipping and adaptive scaling with a relaxed smoothness condition. *arXiv preprint arXiv:1905.11881*, 2019.

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

Songzhu Zheng, Pengxiang Wu, Aman Goswami, Mayank Goswami, Dimitris Metaxas, and Chao Chen. Error-bounded correction of noisy labels. In *ICML*, 2020.

A PROOFS

Proposition 1. *Compared with Eq. (1), Eq. (2) induces noise $z \sim \mathcal{N}(0, \sigma_f^2 M)$ on $\nabla_\theta \ell(f, y)$, s.t., $M \in \mathbb{R}^{p\times p}$ and $M_{i,j} = (\nabla_{\theta_i} f)^T \nabla_{\theta_j} f$, $\forall i, j \in \{1, \cdots, p\}$. Note that the standard deviation of noise on the i-th parameter θ_i is $\sigma_f \|\nabla_{\theta_i} f\|_2$, where $\|\cdot\|_2$ denotes the L_2 norm.*

Proof. Dimension: $z_f \in \mathbb{R}^{1\times c}$, $\nabla_\theta \ell \in \mathbb{R}^{1\times p}$, $\nabla_f \ell \in \mathbb{R}^{1\times c}$, $\nabla_\theta f \in \mathbb{R}^{c\times p}$, $\nabla_{\theta_i} f \in \mathbb{R}^c$.

For Eq. (2), the noisy gradient is

$$\tilde{\nabla}_\theta \ell(f, y) = \tilde{\nabla}_f \ell(f, y) \cdot \nabla_\theta f = (\nabla_f \ell(f, y) + \sigma_f z_f) \cdot \nabla_\theta f = \nabla_\theta \ell(f, y) + \sigma_f z_f \cdot \nabla_\theta f \tag{5}$$

The noise on $\nabla_\theta \ell(f, y)$ is $z = \sigma_f z_f \cdot \nabla_\theta f \in \mathbb{R}^{1\times p}$. Note that $z_f \sim \mathcal{N}(0, I_{c\times c})$, let z_i be the i-th entry of z, we have

$$z_i = \sigma_f \sum_{k=1}^{c} \frac{\partial f_k}{\partial \theta_i} z_{f_k}. \tag{6}$$

Hence,

$$\mathbb{E}[z_i^2] = \sigma_f^2 \|\nabla_{\theta_i} f\|_2^2, \quad \mathbb{E}[z_i z_j] = \sigma_f^2 (\nabla_{\theta_i} f)^T \nabla_{\theta_j} f. \tag{7}$$

□

Proposition 2. *For the cross-entropy loss, compared with Eq. (2), Eq. (3) induces noise $z \sim \mathcal{N}(0, \sigma_y^2 M)$ on $\nabla_f \ell(f, y)$, s.t., $M \in \mathbb{R}^{c\times c}$, $M_{i,i} = c(S_i - 1/c)^2 + (c-1)/c$, and $M_{i,j} = c \cdot S_i S_j - S_i - S_j$, if $i \neq j$. Note that the standard deviation of noise on the i-th entry is $\sigma_y \sqrt{c(S_i - 1/c)^2 + (c-1)/c}$.*

Proof. Dimension: $y \in \mathbb{R}^c$, $z_y \in \mathbb{R}^c$, $S \in \mathbb{R}^c$, $\nabla_f \ell \in \mathbb{R}^{1\times c}$, $\nabla_S \ell \in \mathbb{R}^{1\times c}$, $\nabla_f S \in \mathbb{R}^{c\times c}$.

Firstly, for the softmax function $S = S(f(x))$, we have the derivative matrix

$$\nabla_f S = \mathrm{Diag}(S) - S \cdot S^T, \tag{8}$$

where $\mathrm{Diag}(S)$ is a diagonal matrix with S_i on its i-th diagonal element and 0 otherwise.

For the cross-entropy loss, $\ell(f, y) = -\sum_{k=1}^{c} y_k \log S_k$. Let $\frac{\cdot}{S}$ denote element-wise division, we have

$$\begin{aligned}
\nabla_f \ell(f, y + \sigma_y z_y) &= \nabla_S \ell(f, y + \sigma_y z_y) \cdot \nabla_f S = -\left(\frac{y + \sigma_y z_y}{S}\right)^T \cdot \nabla_f S \\
&= -\left(\frac{y}{S}\right)^T \cdot \nabla_f S - \left(\frac{\sigma_y z_y}{S}\right)^T \cdot \nabla_f S \\
&= \nabla_f \ell(f, y) - \left(\frac{\sigma_y z_y}{S}\right)^T \cdot (\mathrm{Diag}(S) - S \cdot S^T) \\
&= \nabla_f \ell(f, y) - \sigma_y \left(z_y - \sum_{k=1}^{c} z_{y_k} \cdot S\right)^T
\end{aligned} \tag{9}$$

Then it is equivalent to induce noise $z = -\sigma_y(z_y - \sum_{k=1}^{c} z_{y_k} \cdot S)$ on $\nabla_f \ell(f, y)$, whose i-th entry is $z_i = -\sigma_y(z_{y_i} - \sum_{k=1}^{c} z_{y_k} \cdot S_i)$. Note that $z_y \sim \mathcal{N}(0, I_{c\times c})$, then we can derive for $i = j$,

$$\mathbb{E}[z_i^2] = \sigma_y^2 (1 - 2S_i + c \cdot S_i^2) = \sigma_y^2 (c(S_i - 1/c)^2 + (c-1)/c), \tag{10}$$

and for $i \neq j$,

$$\mathbb{E}[z_i z_j] = \sigma_y^2 (c \cdot S_i S_j - S_i - S_j). \tag{11}$$

□

Proposition 3. *For the cross-entropy loss, compared with Eq. (1), Eq. (3) induces noise $z \sim \mathcal{N}(0, \sigma_y^2 M)$ on $\nabla_\theta \ell(f, y)$, s.t., $M \in \mathbb{R}^{p\times p}$ and $M_{i,j} = (\frac{\nabla_{\theta_i} S}{S})^T \frac{\nabla_{\theta_j} S}{S}$, $\forall i, j \in \{1, \cdots, p\}$, where $\frac{\cdot}{S}$ denotes element-wise division. Note that the standard deviation of noise on the i-th parameter θ_i is $\sigma_y \left\| \frac{\nabla_{\theta_i} S}{S} \right\|_2$.*

Proof. Dimension: $z_y \in \mathbb{R}^c$, $\nabla_\theta \ell \in \mathbb{R}^{1\times p}$, $\nabla_S \ell \in \mathbb{R}^{1\times c}$, $\nabla_\theta S \in \mathbb{R}^{c\times p}$, $\nabla_{\theta_i} S \in \mathbb{R}^c$.

For Eq. (3), the noisy gradient is

$$\begin{aligned}\tilde{\nabla}_\theta \ell(f,y) = \nabla_\theta \ell(f, y+\sigma_y z_y) = \nabla_S \ell(f, y+\sigma_y z_y)\cdot \nabla_\theta S = -\left(\frac{y+\sigma_y z_y}{S}\right)^T \cdot \nabla_\theta S \\ = -\left(\frac{y}{S}\right)^T \cdot \nabla_\theta S - \left(\frac{\sigma_y z_y}{S}\right)^T \cdot \nabla_\theta S = \nabla_\theta \ell(f,y) - \left(\frac{\sigma_y z_y}{S}\right)^T \cdot \nabla_\theta S.\end{aligned} \tag{12}$$

The noise on $\nabla_\theta \ell(f,y)$ is $z = -\left(\frac{\sigma_y z_y}{S}\right)^T \cdot \nabla_\theta S$. Note that $z_y \sim \mathcal{N}(0, I_{c\times c})$, let z_i be the i-th entry of z, we have

$$z_i = -\sigma_y \sum_{k=1}^{c} \frac{\partial S_k}{\partial \theta_i}\frac{z_{y_k}}{S_k} = -\sigma_y \sum_{k=1}^{c} \frac{\nabla_{\theta_i} S_k}{S_k} z_{y_k}. \tag{13}$$

Hence,

$$\mathbb{E}[z_i^2] = \sigma_y^2 \left\| \frac{\nabla_{\theta_i} S}{S} \right\|_2^2, \quad \mathbb{E}[z_i z_j] = \sigma_y^2 \left(\frac{\nabla_{\theta_i} S}{S}\right)^T \frac{\nabla_{\theta_j} S}{S}. \tag{14}$$

□

Theorem 1. *Considering a training step on a single sample with cross-entropy loss, we quantify the perturbation on f resulting from SGD noise using the first-order Taylor approximation. The perturbation on the i-th entry of f is denoted as $|\Delta f_i|$, $i \in \{1, 2, \cdots, c\}$, then*

- *for Eq. (1), we have $\mathbb{E}[|\Delta f_i|^2] = \delta^2 \sigma_\theta^2 \cdot \nabla_\theta f_i \cdot (\nabla_\theta f_i)^T$;*
- *for Eq. (2), we have $\mathbb{E}[|\Delta f_i|^2] = \delta^2 \sigma_f^2 \cdot \sum_{j=1}^c (\nabla_\theta f_i \cdot (\nabla_\theta f_j)^T)^2$;*
- *for Eq. (3), we have $\mathbb{E}[|\Delta f_i|^2] = \delta^2 \sigma_y^2 \cdot \sum_{j=1}^c (\nabla_\theta f_i \cdot (\overline{\nabla_\theta f_j})^T)^2$.*

where $\overline{\nabla_\theta f_j} = \nabla_\theta f_j - \sum_{k=1}^c S_k \cdot \nabla_\theta f_k$ is normalized by the weighted average of $\{\nabla_\theta f_k\}_{k=1}^c$.

Proof. Dimension: $\nabla_\theta f \in \mathbb{R}^{c\times p}$, $\nabla_\theta f_i \in \mathbb{R}^{1\times p}$.

Let δ be the learning rate and $z \in \mathbb{R}^{1\times p}$ be the noise on $\nabla_\theta \ell(f,y)$. Since we typically take a small step, then the first-order Taylor approximation implies the perturbation on f resulting from the SGD noise is

$$\begin{aligned}\Delta f &= f(x; \theta + \delta\cdot\nabla_\theta \ell(f,y) + \delta \cdot z) - f(x; \theta + \delta \cdot \nabla_\theta \ell(f,y)) \\ &\approx [f(x;\theta) + \delta\cdot\nabla_\theta f \cdot (\nabla_\theta \ell(f,y) + z)^T] - [f(x;\theta) + \delta \cdot \nabla_\theta f \cdot (\nabla_\theta \ell(f,y))^T] \\ &= \delta \cdot \nabla_\theta f \cdot z^T.\end{aligned} \tag{15}$$

The i-th entry is

$$\Delta f_i = \delta \cdot \nabla_\theta f_i \cdot z^T = \delta \cdot z \cdot (\nabla_\theta f_i)^T. \tag{16}$$

Then we have

$$\mathbb{E}[|\Delta f_i|^2] = \mathbb{E}[\delta \cdot \nabla_\theta f_i \cdot z^T \cdot \delta \cdot z \cdot (\nabla_\theta f_i)^T] = \delta^2 \nabla_\theta f_i \cdot \mathbb{E}[z^T \cdot z] \cdot (\nabla_\theta f_i)^T. \tag{17}$$

The conclusion follows by substituting the covariance matrix $\mathbb{E}[z^T \cdot z]$ in the above equation.

For Eq. (1), $z \sim \mathcal{N}(0, \sigma_\theta^2 I)$, i.e., $\mathbb{E}[z^T \cdot z] = \sigma_\theta^2 I$.

For Eq. (2), $z \sim \mathcal{N}(0, \sigma_f^2 M)$ according to Proposition 1. We have $\mathbb{E}[z^T \cdot z] = \sigma_f^2 M$, where $M_{i,j} = (\nabla_{\theta_i} f)^T \nabla_{\theta_j} f$, i.e.,

$$M = \sum_{j=1}^{c} (\nabla_\theta f_j)^T \nabla_\theta f_j. \tag{18}$$

For Eq. (3), $z \sim \mathcal{N}(0, \sigma_y^2 M)$ according to Proposition 3. We have $\mathbb{E}[z^T \cdot z] = \sigma_y^2 M$, where $M_{i,j} = (\frac{\nabla_{\theta_i} S}{S})^T \frac{\nabla_{\theta_j} S}{S}$, i.e.,

$$M = \sum_{j=1}^{c} \frac{(\nabla_\theta S_j)^T \nabla_\theta S_j}{S_j^2}. \tag{19}$$

Note that $\nabla_\theta S_j = \nabla_f S_j \cdot \nabla_\theta f$, where $\nabla_f S_j$ is the j-th row of $\nabla_f S$ shown in Eq. (8). We have $\nabla_f S_j = S_j(e_j - S)^T$, where $e_j \in \mathbb{R}^c$ is a one-hot vector with 1 at the j-th entry and 0 otherwise. Therefore,

$$\nabla_\theta S_j = \nabla_f S_j \cdot \nabla_\theta f = S_j(e_j - S)^T \cdot \nabla_\theta f = S_j \left(\nabla_\theta f_j - \sum\nolimits_{k=1}^{c} S_k \cdot \nabla_\theta f_k \right). \tag{20}$$

Let $\overline{\nabla_\theta f_j} = \nabla_\theta f_j - \sum_{k=1}^{c} S_k \cdot \nabla_\theta f_k$. Substituting the above equation to M, we get

$$M = \sum_{j=1}^{c} \frac{(\nabla_\theta S_j)^T \nabla_\theta S_j}{S_j^2} = \sum_{j=1}^{c} (\overline{\nabla_\theta f_j})^T \overline{\nabla_\theta f_j}. \tag{21}$$

Finally the result follows by substituting $\mathbb{E}[z^T \cdot z] = \sigma_y^2 M$ in Eq. (17). □

B TRAINING DETAILS

In all experiments on CIFAR-10 and CIFAR-100, we train wide ResNet-28-2 (Zagoruyko & Komodakis, 2016) for 300 epochs using the SGD optimizer with learning rate 0.001, momentum 0.9, weight decay 5×10^{-4}, and a batchsize of 128. This is sufficient for all methods to converge. Standard data augmentation is applied, including per-pixel normalization, horizontal random flip and 32×32 random crop after padding with 4 pixels on each side. On Clothing1M, following the common setting (Patrini et al., 2017), we train an Imagenet-pretrained ResNet-50 using the SGD optimizer with momentum 0.9, weight decay 10^{-3} and a batchsize of 32. The initial learning rate is 10^{-3} and decreased to 10^{-4} after 5 epochs. We use standard data augmentation with per-pixel normalization, horizontal random flip and 224×224 random crop.

C ADDITIONAL RESULTS AND DISCUSSIONS

C.1 THE WEIGHT IN LABEL-CORRECTION

When applying our SLN to label-correction, we use $y_{correction} = \omega \cdot y + (1-\omega) \cdot S$, which is different to the correction strategy in Arazo et al. (2019). Firstly, the common idea is that both Arazo et al. (2019) and our method tend to yield higher ω for large-loss samples. Arazo et al. (2019) have shown that assigning weight according to the loss is superior than using a fixed weight (Reed et al., 2014). Secondly, the key difference is that Arazo et al. (2019) use $y_{correction} = (1-\omega) \cdot y + \omega \cdot S$. We find that our correction strategy helps mitigate the convergence issue. The problem of unstable convergence is reported in Arazo et al. (2019). It may be a common issue for label-correction methods, but has not been widely discussed, nor do we have an explanation on the issue. Here, we provide a hypothesis, with which we justify our label-correction strategy. When we use the model's predictions to correct noisy labels, since the predictions unavoidably contain errors, the model easily accumulates its own errors in each label-correction iteration if the weight on predictions is too high. This should be more clear by analysing how much the label is modified:

$$\begin{aligned} y_{correction} - y &= \omega \cdot (S - y), \quad \text{[Arazoet al. (2019)]}. \\ y_{correction} - y &= (1-\omega) \cdot (S - y), \quad \text{[ours]}. \end{aligned} \tag{22}$$

For small-loss samples, S is close to y, hence labels of these samples will not be modified much no matter how we apply ω. The key is how we correct large-loss samples, for which S is very different from y, and ω is high. These large-loss samples cover three cases in Fig. 7, 1) label $\times$, prediction $\times$, 2) label $\times$, prediction ✓, 3) label ✓, prediction $\times$. Samples of case 2) can be improved by label-correction. Compared with CE, SLN yields more samples of case 2) (the orange region). Still, we can not guarantee that the prediction S is 100% correct, i.e., there are case 1) and case 3) . If we correct the label with large ω on S for these samples, it may result in accumulating prediction errors, because we will train on the modified label. Worse still, the weight ω will be higher in the next label-correction iteration, since the loss between S and y shall be higher after training on the modified label. Such positive feedback may account for the unstable convergence observed in existing label-correction methods. To address this issue, we apply a mild correction with weight $1-\omega$ on S. Therefore, for small-loss samples, the correction does modify the label much because S is close to y; for large-loss samples, $1-\omega$ is small, avoiding high weight on S. As shown in Fig. 8, our strategy works well in terms of convergence.

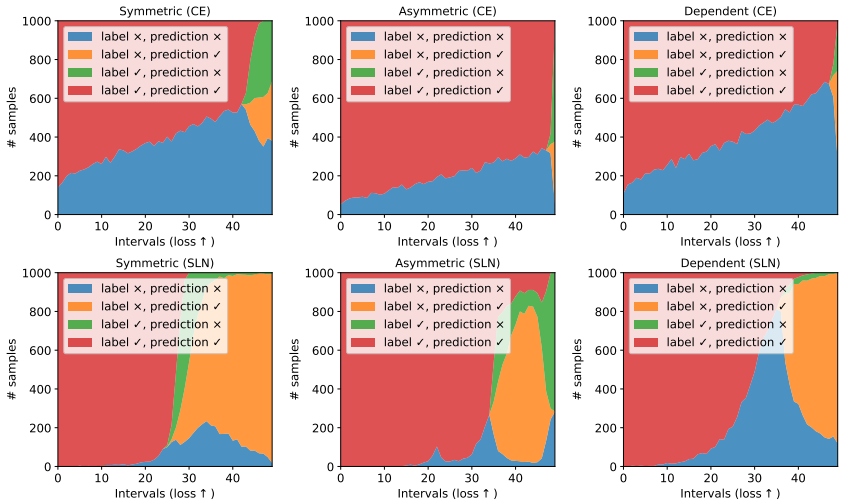

Figure 7: Training samples are sorted in ascending order of loss, uniformly divided into 1000 samples per interval, and dissected according to the correctness of the given label and the prediction. This figure is *exactly the same* as Fig. 3 in the main paper. We present it here for your convenience since we refer to the figure in Section C.1.

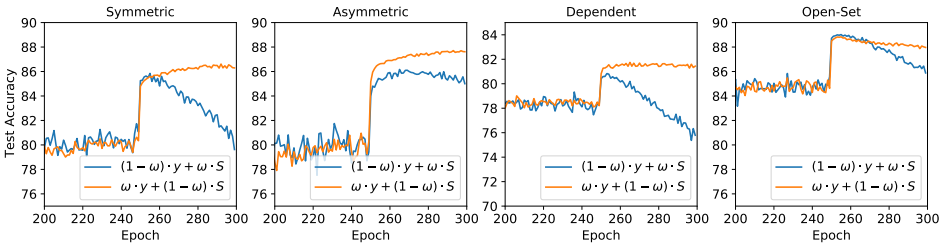

Figure 8: The test accuracy on CIFAR-10, averaged in 5 runs. We zoom in to focus on the effects of label-correction applied in the last 50 epochs. The open-set noise involves samples that do not belong to any class considered in the task, for which label-correction can not address.

C.2 SLN ENHANCES EXISTING METHODS.

We have analyzed that SLN can boosts popular robust training methods such as label-correction and sample-selection. Here we use the model trained with SLN as the initialization of DivideMix (Li et al., 2020). DivideMix combines co-training of two models, sample-selection based on the loss, label-correction/guessing based on semi-supervised learning MixMatch (Berthelot et al., 2019), and other techniques including regularization, augmenting each image twice. It takes more than 4 times of computational cost compared with SLN, and has extensive hyper-parameters. Still, it works well after we heavily tuning the hyper-parameters. With SLN as initialization, we do not need to tune the warm-up phase, because we can train with SLN until convergence without the concern of overfitting. In contrast, without SLN, we need to carefully warm up the model so that it learns enough correct patterns and does not memorizes too much noise. Moreover, as shown in Table 4, the generalization performance of DivideMix is improved with SLN.

Table 4: Testing accuracy on CIFAR-10. Using SLN as the initialization of DivideMix not only eliminates the need of tuning warm-up phase but also improves the test accuracy.

Method	Symmetric	Asymmetric	Dependent	Open-Set	*Training time* (h)
DivideMix	90.38±0.34	87.88±0.45	82.21±0.37	90.49±0.62	*9.61±0.18*
SLN-DivideMix	**90.87±0.28**	**89.31±0.39**	**82.86±0.41**	**91.65±0.59**	*11.76±0.23*