Journal of Machine Learning Research 27 (2026) 1-38 Submitted 8/23; Revised 3/26; Published 3/26

Do We Need to Penalize Variance of Losses for Learning with Label Noise?

Yexiong Lin YLIN6547@UNI.SYDNEY.EDU.AU
Sydney AI Centre
The University of Sydney
Sydney, Australia

Yu Yao YU.YAO@SYDNEY.EDU.AU
Sydney AI Centre
The University of Sydney
Sydney, Australia

Yuxuan Du DUYUXUAN123@GMAIL.COM
College of Computing and Data Science
Nanyang Technological University
Singapore

Jun Yu HARRYJUN@USTC.EDU.CN
School of Information Science and Technology
University of Science and Technology of China
Hefei, China

Bo Han BHANML@COMP.HKBU.EDU.HK
Department of Computer Science
Hong Kong Baptist University
Hong Kong, China

Mingming Gong MINGMING.GONG@UNIMELB.EDU.AU
School of Mathematics and Statistics
The University of Melbourne
Melbourne, Australia
Department of Machine Learning
Mohamed bin Zayed University of Artificial Intelligence
United Arab Emirates

Tongliang Liu* TONGLIANG.LIU@SYDNEY.EDU.AU
Sydney AI Centre
The University of Sydney
Sydney, Australia
Department of Machine Learning
Mohamed bin Zayed University of Artificial Intelligence
United Arab Emirates

Editor: Samy Bengio

*. Corresponding author

Abstract

Statistically consistent algorithms have been widely employed for dealing with noisy labels. Their objective functions are designed so that minimizing the expected risk on noisy data leads to the same minimizer as minimizing the expected risk on clean data. From the weak law of large numbers, penalizing the variance of losses would reduce the discrepancy between the average loss and the expected risk on the clean data when there is a finite training sample, and the estimation error in the model's parameters can be reduced. Interestingly, we found that the variance of losses needs to be encouraged for label-noise learning. Specifically, encouraging a large variance of losses would boost the memorization effect and reduce the harmfulness of incorrect labels. Regularizers can be easily designed to encourage a large variance of losses and be plugged into many existing algorithms. Empirically, the proposed method by encouraging a large variance of losses could improve the generalization ability of baselines on both synthetic and real-world datasets.

Keywords: Variance of losses, the memorization effect, label-noise learning

1. Introduction

Learning with noisy labels can be dated back to the 1980s (Angluin and Laird, 1988). It has recently drawn a lot of attention (Li et al., 2019, 2021; Garg et al., 2023; Wang et al., 2024c; Englesson and Azizpour, 2024; Garg et al., 2025) because large-scale datasets used in training modern deep learning models can easily contain label noise, *e.g.*, ImageNet (Deng et al., 2009) and Clothing1M (Xiao et al., 2015). The reason is that it is expensive and sometimes infeasible to accurately annotate large-scale datasets. Meanwhile, many cheap but imperfect surrogates such as crowdsourcing and web crawling are widely used to build large-scale datasets (Vijayanarasimhan and Grauman, 2014; Welinder and Perona, 2010). Training with such data can lead to poor generalization abilities of modern deep learning models because they will overfit noisy labels (Han et al., 2018b; Zhang et al., 2021).

Generally, the algorithms of learning with noisy labels can be divided into two major categories: *heuristic-based* algorithms and *statistically consistent* algorithms. Methods in the first category improve models' robustness by designing heuristics, such as selecting reliable examples to train model (Han et al., 2018b; Malach and Shalev-Shwartz, 2017; Jiang et al., 2018; Li et al., 2019; Zheltonozhskii et al., 2022; Zhang et al., 2024; Xu et al., 2025), correcting labels (Ma et al., 2018; Kremer et al., 2018; Tanaka et al., 2018; Reed et al., 2014; Guo et al., 2023; Wang et al., 2024a; Sheng et al., 2024; Su et al., 2026), and adding regularization (Han et al., 2018a; Guo et al., 2018; Veit et al., 2017; Liu et al., 2020). Those methods empirically perform well. However, it is not guaranteed that the classifiers learned from noisy data are *statistically consistent*, *i.e.*, the classifiers learned from the noisy data will converge to the optimal ones learned from clean data.

To address this problem, many researchers explore algorithms in the second category. Those algorithms aim to learn *statistically consistent classifiers* (Liu and Tao, 2015; Patrini et al., 2017; Liu et al., 2020; Xia et al., 2020; Bae et al., 2024; Nguyen et al., 2024). Specifically, these algorithms mostly first model the statistical relationship between clean class posterior and noisy class posterior using the *transition matrix* (Patrini et al., 2017; Xia et al., 2019; Li et al., 2021). Then, by leveraging the transition matrix, the noisy data can be linked to clean data. In such a way, their objective functions are specially designed to ensure that minimizing their expected risks on noisy data is equivalent to minimizing

the expected risk on clean data. Thereby, the minimizer of the expected risk on noisy data obtained by using statistically consistent algorithms is identical to that on clean data[1].

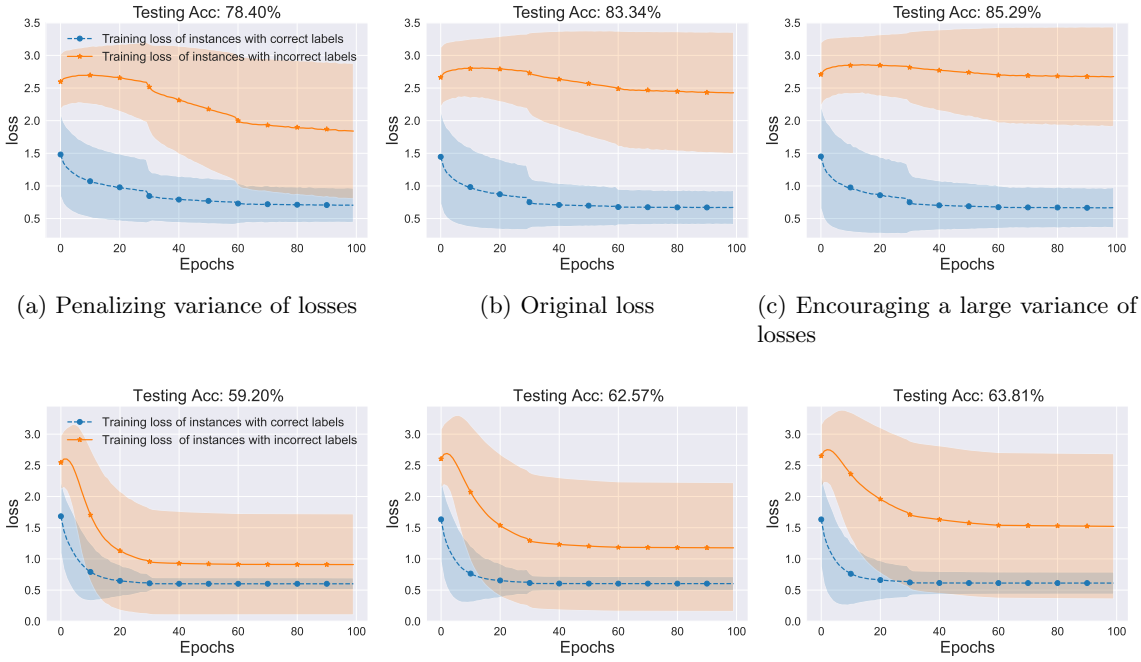
However, in real-world settings, it is infeasible to calculate the expected risk on noisy data. Instead, the empirical risk, calculated as the average of sample losses, is employed as an estimator. This step introduces an estimation error, defined as the discrepancy between the empirical and expected risks on noisy data. Importantly, the estimation error is influenced by the variance of losses. Specifically, according to the Weak Law of Large Numbers, a larger variance of losses leads to a looser upper bound on the estimation error (Mohri et al., 2018), potentially resulting in a large estimation error. Consequently, when the estimation error is large, the minimizer of the empirical risk on noisy data obtained using statistically consistent methods may dramatically deviate from the minimizer of the expected risk on clean data. This implies that the classifiers learned from limited noisy data using statistically consistent methods can significantly differ from the optimal classifiers on clean data. This suggests that high variance in training losses can be harmful.

In this paper, we propose that maintaining high variance is crucial for statistically consistent algorithms in practice. This implies that, during a model's training process, we should encourage a large variance of training losses to prevent it from becoming too small. The reason is that encouraging a large variance of training losses boosts the *memorization effect* that we have found to be beneficial for statistically consistent algorithms. Specifically, the memorization effect refers to the tendency of neural networks to learn simple patterns first before gradually memorizing more complex ones (Arpit et al., 2017; Zhang et al., 2017). When training examples contain noisy labels, neural networks typically memorize correctly labeled examples before incorrectly labeled ones. This occurs because incorrectly labeled examples usually exhibit more complex relationships and are harder to fit compared to correctly labeled examples (Han et al., 2018b). As a result, the loss for incorrectly labeled examples is generally larger than that for correctly labeled examples. By encouraging a large variance in training losses, we can prevent the loss for incorrectly labeled examples from decreasing, which boosts the memorization effect.

We note that boosting the memorization effect is important for statistically consistent algorithms in practice. According to statistical consistency, an optimal classifier can be learned when the number of training noisy examples is infinite. In this setting, the noisy examples can be treated as clean examples by leveraging the transition matrix, which eliminates the effect of label noise. However, in practice, the number of training examples is finite, meaning that noisy examples cannot be fully treated as clean, and the side effects of noisy labels still persist. By boosting the memorization effect for statistically consistent algorithms in practice, we can prevent the loss of incorrectly labeled examples from decreasing, thereby reducing the risk of the model memorizing these incorrectly labeled examples. Consequently, encouraging a large variance of losses can improve the classification performance of statistically consistent algorithms.

Empirically, as illustrated in Fig. 1, the change of the variance of losses does not have much influence on the average training loss of instances with correct labels but makes the average training loss of instances with incorrect labels very different. Specifically, penalizing the variance of losses leads to the average training loss of instances with incorrect labels

1. We assume uniqueness up to risk equivalence. That is, any two classifiers achieving the same minimal expected risk on clean data are considered identical in our analysis.



(a) Penalizing variance of losses (b) Original loss (c) Encouraging a large variance of losses

(d) Penalizing variance of losses (e) Original loss (f) Encouraging a large variance of losses

Figure 1: To analyze the effect of different strategies in controlling the variance of losses, we visualized the average training loss of instances with correct labels (represented by blue dashed lines) and instances with incorrect labels (represented by orange solid lines). The standard deviation for the training losses in each figure is shaded. We employed the Forward algorithm (Patrini et al., 2017) to train a classifier with and without data augmentations, specifically random crop and random horizontal flip. Sub-Fig. (a)-(c) present the results with data augmentations, while Sub-Fig. (d)-(f) show the results without data augmentations. In (a) and (d), the variance of losses was penalized; in (b) and (e), the original loss was employed; and in (c) and (f), the variance of losses was encouraged. The experiments were conducted on the CIFAR-10 dataset with symmetry-flipping label noise, where the noise rate was set to 0.5. A ResNet-18 is a backbone, and the transition matrix $\boldsymbol{T}$ was provided and did not require estimation. The experimental results demonstrate that encouraging a large variance of losses effectively prevent the model from memorizing incorrectly labeled instances.

decreasing fast, which will encourage the model to memorize instances with incorrect labels. Therefore, the classification accuracy decreases. On the contrary, encouraging a large variance of losses can prevent the average training loss of instances with incorrect labels from decreasing, as shown in Fig. 1c and 1f. Therefore, the memorization effect is boosted. As a result, the test accuracy is improved significantly by encouraging a large variance of losses.

The remaining sections of this paper are organized as follows. In Section 2, we provide an overview of the related work in the field. Section 3 presents our proposed method

and highlights its advantages. In Section 4, we evaluate the classification performance of the proposed method using both synthetic and real-world datasets. We also analyze the impact of the proposed method on clean and noisy class posterior distributions and examine the robustness of our method. Additionally, we conduct the sensitivity analysis of the hyperparameter. We also include an ablation study that investigates the effect of the hyperparameters schedule strategy. In Section 5, we discuss the observed memorization effect and explore the influence of the proposed method on clean examples. Finally, we conclude our findings and contributions in Section 6.

2. Related Work

Heuristic-based algorithms. Some methods are proposed to reduce the side-effect of noisy labels using heuristics, *e.g.*, using the *small-loss trick* to select the confident examples (Han et al., 2020; Yao et al., 2020a; Yu et al., 2019; Jiang et al., 2018; Huang et al., 2025). Specifically, when the dataset contains label noise, the network will memorize clean labels first and memorize noisy labels gradually (Han et al., 2018b); thus, the losses with respect to incorrectly labeled examples are usually larger than the losses with respect to correctly labeled examples. The networks are only trained on the confident examples. The learning of neural networks and the selection of confident examples are processed alternatively. The performance of the networks improves over iterations, and then the quality of confident examples is improved. More recently, Semi-Supervised Learning is employed to learn noise-robust classifier (Li et al., 2019; Zheltonozhskii et al., 2022; Lin et al., 2023; Huang et al., 2023). The first step in these methods is also selecting confident examples as the labeled set and the remaining examples as the unlabeled set. By employing the Semi-Supervised Learning techniques, the unlabeled set can be used to train the classifier and lead to the improvement of classification performance (Berthelot et al., 2019; Li et al., 2019; Garg et al., 2023). Zheltonozhskii et al. (2022) further introduces Self-Supervised Learning to initialize the weights of the network, which could help the model identify confident examples better. To better separate clean examples and noisy examples, Kim et al. (2025) propose SplitNet, which is a method that automatically learns to distinguish clean examples based on prediction history and observed labels. Alternatively, to mitigate the bias of loss-based metrics in long-tailed distributions, Wei et al. (2026) propose a *small-distance* criterion, which identifies confident examples by measuring the distance between sample features and their corresponding class centroids. Beyond selecting clean examples, Shu et al. (2019) proposed Meta-Weight-Net to learn an explicit weighting function via meta-learning. While effective, such approaches typically rely on a clean validation set to guide the bi-level optimization, which limits their applicability in scenarios where clean labels are unavailable. Those methods empirically perform well. However, most of them do not provide statistical guarantees for the learned classifiers on noisy data.

Transition matrix-based algorithms. Some methods try to model the underlying noise structure to guarantee the consistency of classifiers. Intuitively, the clean distributions can be inferred through the noisy distributions when the underlying noise model is known. Thus, the consistency of classifiers can be guaranteed. The label noise transition matrix $\boldsymbol{T}(x) \in [0,1]^{C\times C}$, where C is the number of classes, has been widely employed to model the underlying noise structure (Liu and Tao, 2015; Patrini et al., 2017; Xia et al., 2020; Li

et al., 2021; Nguyen et al., 2024). Let the clean class posterior $P(\boldsymbol{Y}|X=\boldsymbol{x}) := [P(Y=1|X=\boldsymbol{x}),\ldots,P(Y=C|X=\boldsymbol{x})]^\top$. It can be inferred by employing the noisy class posterior $P(\tilde{\boldsymbol{Y}}|X=\boldsymbol{x})$ and the transition matrix $\boldsymbol{T}(\boldsymbol{x})$, where $\boldsymbol{T}_{ij}(\boldsymbol{x}) = P(\tilde{Y}=i|Y=j,X)$, *i.e.*, $P(\boldsymbol{Y}|X=\boldsymbol{x}) = [\boldsymbol{T}(\boldsymbol{x})]^{-1}P(\tilde{\boldsymbol{Y}}|X=\boldsymbol{x})$. Then, the expected risk of a function $f(X,Y)$ modeling $P(\boldsymbol{Y}|X)$ can be formulated as the expected risk of a function $g(X,\tilde{Y})$ modeling $P(\tilde{\boldsymbol{Y}}|X)$ multiplied by $\boldsymbol{T}(X)$, *i.e.*, $R(f(X,Y)) = R([\boldsymbol{T}(X)]^{-1}g(X,\tilde{Y}))$. In practice, the expected risk $R([\boldsymbol{T}(X)]^{-1}g(X,\tilde{Y}))$ can not be calculated. Existing methods approximate the expected risk with the average loss over the noisy training examples. Patrini et al. (2017) uses the transition matrix to correct the predicted clean labels to the noisy labels. Natarajan et al. (2013) designs a proxy loss whose sample average can converge to the expected risk on the clean data. Van Rooyen and Williamson (2018) further generalize this method. They use the invertibility of the transfer matrix to construct the unbiased estimator. Importance reweighting (Liu and Tao, 2015; Xia et al., 2019) uses the transition matrix to calculate the weights of loss with regard to each example.

Graphical model-based algorithms. Beyond transition matrix-based approaches, another important line of research leverages graphical models to represent dependencies between observed variables and latent variables. Classical approaches adopt the Expectation–Maximization (EM) framework to jointly infer the latent clean labels and estimate the parameters of the graphical model (Vahdat, 2017). More recent work uses graphical models with structured probabilistic or causal assumptions to explicitly model the generative process of noisy data. CausalNL (Yao et al., 2021) employs a structural causal model to help the learning of classifiers by leveraging anti-causal learning (Schölkopf, 2022). InstanceGM (Garg et al., 2023) further improves CausalNL by introducing the semi-supervised learning technique MixMatch (Berthelot et al., 2019). Bae et al. (2022) introduce a generative calibration model to recover true labels from noisy predictions given by trained classifiers. Lin et al. (2024) propose to learn a latent causal graph to model the generative process of noisy labels. This approach allows the model to estimate instance-dependent noise transition matrices without relying on predefined similarity assumptions about transition matrices.

Robust loss functions. A major line of research focuses on designing loss functions that are inherently robust to label noise. These approaches can be divided into two categories, depending on whether they guarantee that the minimizer for the expected risk on noisy data is identical to the minimizer for the expected risk on clean data.

The first category provides such a guarantee. For these methods, the expected risk on noisy data can be expressed as an affine transformation of the expected risk on clean data, which guarantees that the two expected risks share identical minimizers. For example, Ghosh et al. (2015, 2017) show that the loss satisfying the symmetric condition is inherently robust to symmetry-flipping label noise. Similarly, Van Rooyen et al. (2015) introduced the unhinged loss, which is robust to symmetry-flipping label noise, because the contributions of incorrectly labeled positive examples and incorrectly labeled negative examples can cancel out in expectation. Furthermore, Menon et al. (2015) demonstrate that certain performance metrics, such as balanced error rate (BER) and area under the ROC curve (AUC), are robust to label noise, which can thus be used for optimization in the presence of label noise.

The second category does not guarantee the identical minimizers, but instead focuses on designing loss functions that reduce the influence of incorrect labels. The generalized

cross-entropy (GCE) loss (Zhang and Sabuncu, 2018; Li et al., 2023) interpolates between cross-entropy and mean absolute error (MAE), thus balancing the optimization stability of cross-entropy and the robustness of MAE. Symmetric cross-entropy (SCE) loss (Wang et al., 2019) combines cross-entropy loss with its reverse counterpart and has been shown to improve robustness in practice. Feng et al. (2021) analyze cross-entropy via a Taylor expansion and derive a bounded surrogate that enhances robustness to label noise. More recently, Wang et al. (2024b) propose ϵ-Softmax, a modified softmax transformation that approximates one-hot target vectors within a controllable approximation error, and prove that integrating it into existing objectives can further improve robustness to label noise.

3. Enhancing Variance of Losses for Label-Noise Learning

In this section, we propose our method, *i.e.*, loss-**V**ariance **R**egularization for label-**N**oise **L**earning (VRNL). We reveal how the proposed method reduces the side effect of label noise by boosting the memorization effect. We also illustrate how our regularizer can be generally applied to existing statistically consistent algorithms.

3.1 A Loss-Variance Regularization for Label-Noise Learning (VRNL)

The proposed method demonstrates its effectiveness in preventing the model from memorizing incorrect labels. Thus, the classification performance can be improved. Intuitively, by encouraging a large variance of losses, the proposed method prevents the losses of incorrectly labeled examples from decreasing while promoting the reduction of losses for correctly labeled examples, as illustrated in Fig 1. From a theoretical perspective, the proposed method assigns small weights to gradients of large-loss examples, which are likely to be incorrectly labeled, and assigns large weights to the gradients of small-loss examples, which are likely to be correctly labeled. Consequently, the model places more trust in small-loss examples and gives them a greater influence on parameter updates. This reduces the harmful impact of incorrectly labeled examples by reducing their influence on the learning process.

In the rest of this section, we first introduce a general form of VRNL. Then, we analyze the influence of VRNL. Next, we describe the induced robustness–fitting tradeoff. We also provide a variance view of existing methods. Finally, we discuss the advantages of VRNL.

A general form of VRNL. Let Var[.] denote the variance of a distribution. For a random variable X, $\mathrm{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}^2[X]$. Let C denote the number of classes. Let $f_\theta : \mathcal{X} \to \Delta^{C-1}$ be a mapping parameterized by θ (*e.g.*, a neural network), where Δ^{C-1} denotes a probability simplex. Generally, the expected risk w.r.t. noisy data is formulated as $\mathbb{E}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[\ell(f_\theta(X), \tilde{Y})]$, where $\ell(\cdot)$ is the loss function employed. We propose to add a variance regularizer to the losses. Specifically, the risk function of our method is

$$R_G(f_\theta) = \mathbb{E}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[\ell(f_\theta(X), \tilde{Y})] - \alpha \mathrm{Var}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[\ell(f_\theta(X), \tilde{Y})], \tag{1}$$

where $\mathrm{Var}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[\ell(f_\theta(X), \tilde{Y})]$ is a regularization term, and α is an adjustable hyper-parameter to control the strength of the regularization effect. Unlike conventional regularizers that explicitly encourage simplicity of models, *e.g.*, by penalizing large parameter norms, our variance regularization term does not directly constrain model complexity. Instead, it induces an inductive bias on the optimization process by encouraging a large

variance of loss, which can prevent models from memorizing high-loss examples that are more likely to be incorrectly labeled when the dataset contains label noise.

In practice, α is set to a small positive value so that the regularizer can encourage a large variance of losses but does not dominate the original objective. Empirical analyses in Sec. 5.2 demonstrate that our regularizer only has a small impact on the statistical consistency of the algorithms. Since the accuracy of models on noisy data is consistent with the accuracy on clean data through the statistically consistent algorithms, a suitable value for α can be selected by employing the noisy validation set. For further details and discussions on this matter, please refer to the experiment section of the paper.

Influence of VRNL. To provide a theoretical grounding for our method, we exploit the influence of our designed regularizer with respect to the update of parameter θ by deriving the gradient of the risk function with respect to θ, *i.e.*,

$$\begin{aligned}
\frac{R_G(f_\theta)}{\partial\theta} &= \frac{\partial\mathbb{E}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[\ell(f_\theta(X),\tilde{Y})]}{\partial\theta} - \alpha\frac{\partial\mathrm{Var}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[\ell(f_\theta(X),\tilde{Y})]}{\partial\theta} \\
&= \frac{\partial\mathbb{E}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[\ell(f_\theta(X),\tilde{Y})]}{\partial\theta} - \alpha\left\{\frac{\partial\mathbb{E}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[\ell^2(f_\theta(X),\tilde{Y})]}{\partial\theta} - \frac{\partial\mathbb{E}^2_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[\ell(f_\theta(X),\tilde{Y})]}{\partial\theta}\right\} \\
&= \mathbb{E}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[\frac{\partial\ell(f_\theta(X),\tilde{Y})}{\partial\theta}] - \alpha\left\{\mathbb{E}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}\left[2\ell(f_\theta(X),\tilde{Y})\frac{\partial\ell(f_\theta(X),\tilde{Y})}{\partial\theta}\right]\right. \\
&\left.-2\mathbb{E}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[\ell(f_\theta(X),\tilde{Y})]\mathbb{E}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}\left[\frac{\partial\ell(f_\theta(X),\tilde{Y})}{\partial\theta}\right]\right\} \\
&= \mathbb{E}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}\left[\left(1+2\alpha\mathbb{E}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[\ell(f_\theta(X),\tilde{Y})] - 2\alpha\ell(f_\theta(X),\tilde{Y})\right)\frac{\partial\ell(f_\theta(X),\tilde{Y})}{\partial\theta}\right] \\
&= \mathbb{E}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}\left[W\frac{\partial\ell(f_\theta(X),\tilde{Y})}{\partial\theta}\right], \qquad (2)
\end{aligned}$$

where

$$W = 1 + 2\alpha\left(\mathbb{E}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[\ell(f_\theta(X),\tilde{Y})] - \ell(f_\theta(X),\tilde{Y})\right).$$

The Eq. 3.1 demonstrates that VRNL is mathematically equivalent to an explicit gradient reweighting mechanism: For a specific example $(x,\tilde{y})$, its corresponding gradient is $w\frac{\partial\ell(f_\theta(x),\tilde{y})}{\partial\theta}$, where the weight w is $w = 1 + 2\alpha(\mathbb{E}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[\ell(f_\theta(X),\tilde{Y})] - \ell(f_\theta(x),\tilde{y}))$. In our experiments, α is selected to be small, which implies most of w is positive. The above equation shows that 1) if the loss of the example $(x,\tilde{y})$ is smaller than the expectation of the losses $\mathbb{E}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[\ell(f_\theta(X),\tilde{Y})]$, $\left(\mathbb{E}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[\ell(f_\theta(X),\tilde{Y})] - \ell(f_\theta(x),\tilde{y})\right)$ will be positive, and the weight w associated with its gradient is larger than 1. Then the example $(x,\tilde{y})$ contributes more to the update of the parameter θ. 2) If the loss of the example $(x,\tilde{y})$ is larger than the expectation of the losses, $\left(\mathbb{E}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[\ell(f_\theta(X),\tilde{Y})] - \ell(f_\theta(x),\tilde{y})\right)$ will be negative. The weight w associated with its gradient is small. Then the example $(x,\tilde{y})$ contributes less to the update of parameter θ.

The VRNL can decrease the influence of incorrectly labeled examples by assigning small weights for the gradients of incorrectly labeled examples. Due to the memorization effect,

deep neural networks tend to learn easy (clean) patterns first and gradually learn hard (noisy) patterns (Bai et al., 2021; Han et al., 2018b; Arpit et al., 2017). In learning with noisy labels, large-loss examples are more likely to have incorrect labels and should not be trusted. By employing the proposed method, the gradients of examples with incorrect labels are assigned with small weights W. In such a way, incorrectly labeled examples would have less contribution to the update of the parameter θ, which prevents the model from memorizing incorrect labels. Therefore, the classification performance can be improved.

A variance view of existing methods. We show that existing sample selection and reweighting methods (Han et al., 2018b; Ren et al., 2018; Li et al., 2019) encourage a large variance of losses implicitly. Intuitively, these methods aim to guide the model to focus on learning correctly labeled examples while ignoring incorrectly labeled examples during the training. This process results in small losses for correctly labeled examples and large losses for incorrectly labeled examples. Consequently, these methods inherently promotes a large variance in the overall loss distribution.

Specifically, the loss functions of these methods can be unified as follows:

$$R_U(f_\theta) = \mathbb{E}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[M(X,\tilde{Y})\ell_{CE}(f_\theta(X),\tilde{Y})], \tag{3}$$

where ℓ_{CE} is the standard cross-entropy loss, and $M(X,\tilde{Y})$ represents the weight assigned to the loss of each training example.

To mitigate the impact of label noise, existing methods aim to assign smaller weights ($M(X,\tilde{Y})$) to incorrectly labeled examples and larger weights to correctly labeled examples, which inherently encourages a large variance of losses. For example, sample selection methods (Han et al., 2018b; Li et al., 2019) use the small-loss trick to distinguish confident (likely correct) examples from unconfident (likely incorrect) ones. They assign binary weights: $M(X,\tilde{Y}) = 1$ for confident examples and $M(X,\tilde{Y}) = 0$ for unconfident examples. Similarly, reweighting-based methods (Ren et al., 2018) leverage a small clean validation set to compute weights. These weights are continuous and determined by minimizing the validation loss, assigning larger weights to examples likely to have correct labels.

For both methods, examples with smaller weights contribute less to parameter updates, resulting in smaller reductions in their corresponding losses during training. Conversely, examples with larger weights contribute more significantly to parameter updates, leading to greater reductions in their losses. As a result, after optimization, the losses of correctly labeled examples tend to be small, while the losses of incorrectly labeled examples remain large. This process encourages a large variance of losses, quantified as $\mathrm{Var}_{(X,\tilde{Y})\sim\tilde{\mathcal{D}}}[\ell_{CE}(f_\theta(X),\tilde{Y})]$.

The advantages of VRNL The first advantage of VRNL is that it does not rely on a clean validation set for hyperparameter selection. In contrast, the sample reweighting method (Ren et al., 2018) requires a clean validation set to compute weights that correct the gradient directions during training. Theoretically, these weights are calculated based on the difference between the clean and noisy data distributions. Specifically, Ren et al. (2018) optimize weights $\{w_i\}_{i=1}^n$ (where n is the mini-batch size) by employing a clean validation set to match the gradient direction of the noisy training loss to that of the clean validation loss, thereby mitigating the impact of label noise. In contrast, VRNL introduces only one hyperparameter, α, which can be selected using a noisy validation set. This is because

VRNL is designed for statistically consistent algorithms, and its influence on statistical consistency is small. Further details are provided in Section 4.4.

Our method also offers computational efficiency in calculating gradient weights. The sample reweighting method proposed by Ren et al. (2018) requires optimizing weights to minimize the loss on a clean validation set, which involves additional optimization steps. As stated in their paper, this process increases training time by 3 times compared to regular methods. In contrast, VRNL implicitly assigns weights to gradient losses by encouraging a large variance of losses, eliminating the need for extra optimization steps, and significantly improving computational efficiency.

Furthermore, our method can effectively leverage information from the entire training dataset. Unlike the existing sample selection methods (Jiang et al., 2018; Han et al., 2018b; Cheng et al., 2022) that aim to discard incorrectly labeled examples, our method does not discard any training examples since discarding them introduces several limitations. First, completely ignoring certain examples creates a significant distribution gap between the training and test sets, which is well-known to negatively impact classifier performance. For example, to address the distribution gap, Cheng et al. (2020) employs human annotators and utilizes active learning. Second, some discarded examples can be hard but correctly labeled examples, which should be leveraged for training. Third, although discarded examples have incorrect labels, applying the transition matrix can make use of them. Specifically, the transition matrix models the relationship between clean and noisy labels, allowing the algorithm to estimate the true label distribution from the noisy labels. By leveraging this relationship, the noisy labels can still provide valuable information about the clean labels. This partial recovery of label information still makes these examples beneficial for training rather than discarding them entirely.

3.2 Integrating VRNL into Statistically Consistent Algorithms

In practice, since the expected risk $R_G(f_\theta)$ in Eq. 1 can not be calculated, the average loss is employed as an approximation. Let n be the number of training examples. Generally, the objective loss of our method is as follows:

$$\hat{R}(f_\theta) = \frac{1}{n}\sum_{i=1}^{n}\ell(f_\theta(x_i),\tilde{y}_i) - \alpha\left(\frac{1}{n}\sum_{i=1}^{n}\ell(f_\theta(x_i),\tilde{y}_i)^2 - \left(\frac{1}{n}\sum_{i=1}^{n}\ell(f_\theta(x_i),\tilde{y}_i)\right)^2\right).$$

We further illustrate specific forms and settings of our designed regularization working with existing methods, *i.e.*, Forward (Patrini et al., 2017), Importance Reweighting (Liu and Tao, 2015), and VolMinNet (Li et al., 2021). Empirically, our method improves their classification accuracy.

Work with Forward. Forward correction (Patrini et al., 2017) exploits the noise transition matrix $\boldsymbol{T}$ to infer the clean class posterior distribution, *i.e.*, $P(\boldsymbol{Y}|X=\boldsymbol{x}) = [\boldsymbol{T}(\boldsymbol{x})]^{-1}P(\tilde{\boldsymbol{Y}}|X=\boldsymbol{x})$. We use the same method in the original paper (Patrini et al., 2017) to estimate the transition matrix. The objective loss function by combining our method

with Forward can be formulated as follows:

$$\hat{R}_{\text{Forward}}(f_\theta, \hat{\boldsymbol{T}}) = \frac{1}{n}\sum_{i=1}^{n} \ell_{CE}(\hat{\boldsymbol{T}} f_\theta(x_i), \tilde{y}_i) \\ - \alpha \left(\frac{1}{n}\sum_{i=1}^{n} \ell_{CE}(\hat{\boldsymbol{T}} f_\theta(x_i), \tilde{y}_i)^2 - \left(\frac{1}{n}\sum_{i=1}^{n} \ell_{CE}(\hat{\boldsymbol{T}} f_\theta(x_i), \tilde{y}_i) \right)^2 \right),$$

where ℓ_{CE} is the cross-entropy loss, $\hat{\boldsymbol{T}}$ is the estimated transition matrix, f_θ models the clean class-posterior distribution, $\hat{\boldsymbol{T}} f_\theta$ models the noisy class-posterior distribution.

Work with Importance Reweighting. Importance Reweighting uses the weighted expected risk on the noisy data to estimate the expected risk on the clean data, *i.e.*, $\mathbb{E}_{(X,Y)}[\ell(f_\theta(X), Y)] = \mathbb{E}_{(X,\tilde{Y})}[\frac{P(X,Y)}{P(X,\tilde{Y})}\ell(f_\theta(X), \tilde{Y})] = \mathbb{E}_{(X,\tilde{Y})}[\frac{P(Y|X)}{P(\tilde{Y}|X)}\ell(f_\theta(X), \tilde{Y})]$ (Liu and Tao, 2015). In practice, the average loss is used to estimate the expected risk. To calculate the weight of the loss, both noisy class-posterior distribution $P(\tilde{Y}|X)$ and clean class-posterior distribution $P(Y|X)$ need to be estimated. The objective loss function by combining our method with Important Reweighting can be formulated as follows:

$$\hat{R}_{IR}(f_\theta) = \frac{1}{n}\sum_{i=1}^{n} \hat{\beta}_i \ell_{CE}(f_\theta(x_i), \tilde{y}_i) \\ - \alpha \left(\frac{1}{n}\sum_{i=1}^{n} \hat{\beta}_i^2 \ell_{CE}(f_\theta(x_i), \tilde{y}_i)^2 - \left(\frac{1}{n}\sum_{i=1}^{n} \hat{\beta}_i \ell_{CE}(f_\theta(x_i), \tilde{y}_i) \right)^2 \right),$$

where $\hat{\beta}_i = \frac{\hat{P}_D(y_i|x_i)}{\hat{P}_{D_\rho}(\tilde{y}_i|x_i)}$, D is the clean distribution, D_ρ is the noisy distribution. The gradient of $\hat{R}_{IR}$ w.r.t. an example $(x_i, \tilde{y}_i)$ is as follows:

$$\nabla \hat{R}_{IR}(f_\theta, (x, \tilde{y})) = \frac{1}{n}\sum_{i=1}^{n} \hat{w}_i \left(\ell_{CE}(f_\theta(x_i), \tilde{y}_i) \frac{\partial \hat{\beta}_i}{\partial \theta} + \hat{\beta}_i \frac{\partial \ell_{CE}(f_\theta(x_i), \tilde{y}_i)}{\partial \theta} \right),$$

where

$$\hat{w}_i = 1 + 2\alpha \left(\frac{1}{n}\sum_{j=1}^{n} \hat{\beta}_j \ell_{CE}(f_\theta(x_j), \tilde{y}_j) - \hat{\beta}_i \ell_{CE}(f_\theta(x_i), \tilde{y}_i) \right).$$

The $\ell_{CE}(f_\theta(x), \tilde{y}) \frac{\partial \hat{\beta}_i}{\partial \theta} + \hat{\beta}_i \frac{\partial \ell_{CE}(f_\theta(x), \tilde{y})}{\partial \theta}$ is the gradient of the original Importance Reweighting loss. When the label $\tilde{y}_i$ is incorrect, the reweighted loss $\hat{\beta}_i \ell_{CE}(f_\theta(x_i), \tilde{y}_i)$ is usually larger than the average loss $\frac{1}{n}\sum_{j=1}^{n} \hat{\beta}_j \ell_{CE}(f_\theta(x_j), \tilde{y}_j)$. Then their difference is negative, which leads the weight $\hat{w}_i$ to be small because the hyper-parameter α is positive. As a result, the instance with an incorrect label has a small contribution to the update of parameter θ, thereby preventing the model from memorizing the incorrect labels.

In the implementation, the early stopping technique is used for the approximation of the clean class-posterior distribution. Specifically, the model f_θ is trained on noisy data with 20 epochs, and we feed the model output to a softmax function, then use the output of

the softmax function $g(x)$ to approximate the clean class-posterior distribution. The noise transition matrix $\boldsymbol{T}$ has also been estimated by using the same approach as in Forward correction. Then, the model f_θ is further optimized by both weighted loss and regularization for the variance of losses as follows:

$$\hat{R}_{IR}(f_\theta) = \frac{1}{n}\sum_{i=1}^{n}\left[\ell_{CE}(f_\theta(x_i),\tilde{y}_i)\frac{g_{\tilde{y}}(x_i)}{(\hat{\boldsymbol{T}}g)_{\tilde{y}}(x_i)}\right] - \alpha\hat{\sigma}_\theta^2,$$

here

$$\hat{\sigma}_\theta^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\ell_{CE}(f_\theta(x_i),\tilde{y}_i)\frac{g_{\tilde{y}}(x_i)}{(\hat{\boldsymbol{T}}g)_{\tilde{y}}(x_i)}\right)^2 - \left(\frac{1}{n}\sum_{i=1}^{n}\ell_{CE}(f_\theta(x_i),\tilde{y}_i)\frac{g_{\tilde{y}}(x_i)}{(\hat{\boldsymbol{T}}g)_{\tilde{y}}(x_i)}\right)^2.$$

Work with VolMinNet. VolMinNet is an end-to-end label-noise learning method that learns the transition matrix and the clean class-posterior distribution simultaneously (Li et al., 2021). It optimizes two objectives: 1). a trainable diagonally dominant column stochastic matrix $\hat{\boldsymbol{T}}$ by minimizing the determinate $\log\det(\hat{\boldsymbol{T}})$; 2). the parameter θ of the model by the cross-entropy loss between the noisy label and the predicted probability by the neural network. In experiments, our VRNL only regularizes the parameter θ by calculating the variance of cross-entropy losses. The objective of combining our method with VolMinNet can be formulated as follows:

$$\begin{aligned}\hat{R}_{vol}(f_\theta,\hat{\boldsymbol{T}}) = &\frac{1}{n}\sum_{i=1}^{n}\ell_{CE}(\hat{\boldsymbol{T}}f_\theta(x_i),\tilde{y}_i) + \lambda\log\det(\hat{\boldsymbol{T}}) \\ &- \alpha\left(\frac{1}{n}\sum_{i=1}^{n}\ell_{CE}(\hat{\boldsymbol{T}}f_\theta(x_i),\tilde{y}_i)^2 - \left(\frac{1}{n}\sum_{i=1}^{n}\ell_{CE}(\hat{\boldsymbol{T}}f_\theta(x_i),\tilde{y}_i)\right)^2\right),\end{aligned}$$

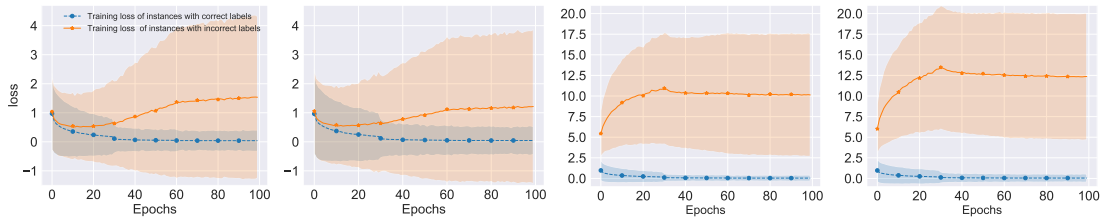
where $\lambda > 0$ is an adjustable hyper-parameter, we set $\lambda = 0.0001$ in all experiments. The transition matrix $\hat{\boldsymbol{T}}$ should be differentiable, diagonally dominant and column stochastic.

Our method could help the *state-of-the-art* transition matrix estimation method VolMinNet (Li et al., 2021) to better estimate the transition matrix and the clean class-posterior distribution. Specifically, VolMinNet requires the clean class posteriors to be diverse, which is called the *sufficiently scattered* assumption (Li et al., 2021). By encouraging a large variance of the loss, the diversity of the estimated noisy class posteriors is encouraged, so the estimated clean class posteriors are also encouraged. Then, the transition matrix can be better learned, which leads to the clean class-posterior distribution being better estimated.

4. Experiments

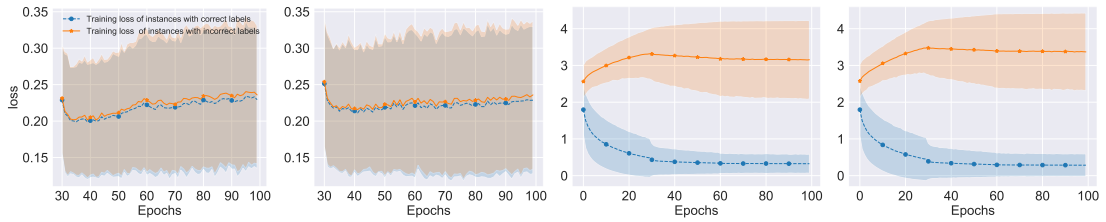
In this section, we present the empirical results of VRNL. We begin by showing the classification performance of VRNL on both synthetic and real-world noisy datasets. Next, we analyze the impact of VRNL on the clean and noisy class posteriors. Additionally, we conduct a sensitivity analysis of the hyperparameter α. Finally, we perform an ablation study to assess the effects of different schedule strategies for α.

Datasets. We evaluate the performance of the proposed method on three manually corrupted datasets: Fashion-MNIST (Xiao et al., 2017), CIFAR-10 (Krizhevsky et al., 2009),



(a) (b) (c) (d)

Figure 2: Change of losses by increasing training epochs for Reweighting. (a) and (b) illustrate CE losses of $P(Y|X)$ without or with encouraging a large variance of losses; (c) and (d) illustrate CE losses of $P(\tilde{Y}|X)$ without or with encouraging a large variance of losses, respectively. The standard deviation for the training losses in each figure is shaded.



(a) (b) (c) (d)

Figure 3: Change of losses by increasing training epochs for VolMinNet. (a) and (b) illustrate CE losses of $P(Y|X)$ without or with encouraging a large variance of losses; (c) and (d) illustrate CE losses of $P(\tilde{Y}|X)$ without or with encouraging a large variance of losses, respectively. The standard deviation for the training losses in each figure is shaded.

and CIFAR-100 (Krizhevsky et al., 2009). Additionally, we assess its effectiveness on three real-world noisy datasets: CIFAR-N (Wei et al., 2022), Clothing1M (Xiao et al., 2015) and WebVision (Li et al., 2017). CIFAR-N contains human-annotated noisy labels for CIFAR-10 and CIFAR-100 datasets. For CIFAR-10, CIFAR-N provides five types of noisy labels: "Worst", "Aggregate", "Random 1", "Random 2" and "Random 3". The dataset is referred to as CIFAR-10N. For CIFAR-100, CIFAR-N provides one type of noisy label: "Noisy Fine". The dataset is referred to as CIFAR-100N. Clothing1M (Xiao et al., 2015) consists of 1 million images with real-world noisy labels, including 50,000, 14,000, and 10,000 images with clean labels for training, validation, and testing, respectively. WebVision contains 2.4 million images obtained from the internet, each associated with a real-world noisy label. Following the approach of previous work (Chen et al., 2019), we select the first 50 classes of the Google image subset as the training set. The trained model is then evaluated on the WebVision validation set and the ILSVRC12 validation set. It is worth noting that existing methods such as Forward (Patrini et al., 2017) and T-revision (Xia et al., 2019) use the 50,000 clean data for initializing the transition matrix and validate their performance on the 14,000 clean data. However, we assume that access to the clean data is unavailable, and therefore, we do not use the clean data for training and validation. For all datasets, we reserve 10% of the training examples as a noisy validation set.

Table 1: Means and standard deviations (percentage) of classification accuracy on Fashion-MNIST under class-dependent label noise. Results with "*" mean that they are the highest accuracy. The symbol "•" indicates the improvements are statistically significant.

	Fashion-MNIST					
	Sym-20%	Sym-50%	Sym-80%	Sym-90%	Pair-20%	Pair-45%
CE	92.22 ± 0.07	89.95 ± 0.22	81.52 ± 0.26	74.26 ± 0.85	92.78 ± 0.15	82.44 ± 3.85
Decoupling	90.23 ± 0.10	87.76 ± 0.71	78.65 ± 0.79	15.79 ± 7.95	91.52 ± 0.40	90.15 ± 0.27
MentorNet	92.28 ± 0.18	90.42 ± 0.10	78.90 ± 0.61	50.21 ± 4.20	92.07 ± 0.13	89.28 ± 0.18
Co-Teaching	93.63 ± 0.09*	91.84 ± 0.17*	81.15 ± 0.50	58.34 ± 3.31	93.64 ± 0.04*	91.38 ± 0.53
T-Revision	91.27 ± 0.61	85.35 ± 1.45	75.54 ± 4.55	59.89 ± 9.02	92.42 ± 0.37	71.74 ± 11.99
Forward	90.71 ± 0.60	87.30 ± 0.56	76.89 ± 2.23	54.24 ± 4.27	91.58 ± 0.32	77.69 ± 13.81
Forward-VRNL	**92.16 ± 0.27** •	**90.38 ± 0.25** •	**80.86 ± 1.90** •	**63.52 ± 4.46** •	**92.14 ± 0.55** •	**79.50 ± 12.08** •
Reweight	91.36 ± 0.45	89.12 ± 0.59	82.55 ± 1.94	45.63 ± 24.18	91.04 ± 0.48	73.28 ± 4.78
Reweight-VRNL	**91.73 ± 0.48** •	**89.74 ± 0.48** •	**85.26 ± 0.89*** •	**77.74 ± 2.06*** •	**91.35 ± 0.38**	**76.39 ± 5.56** •
VolMinNet	91.82 ± 0.38	88.88 ± 0.60	81.16 ± 1.21	67.46 ± 2.70	92.64 ± 0.18	92.98 ± 0.11
VolMinNet-VRNL	**92.31 ± 0.21** •	**90.52 ± 0.20** •	**82.41 ± 0.79** •	**69.86 ± 1.67** •	**93.21 ± 0.20** •	**93.14 ± 0.20***
BLTM	91.63 ± 0.89	89.43 ± 0.51	82.08 ± 1.72	68.34 ± 5.66	92.61 ± 0.19	87.93 ± 6.14
BLTM-VRNL	**92.28 ± 0.14** •	**90.16 ± 0.17** •	**84.15 ± 0.67** •	**74.95 ± 2.71** •	**92.90 ± 0.19** •	**90.51 ± 0.19** •

Table 2: Means and standard deviations (percentage) of classification accuracy on Fashion-MNIST under instance-dependent label noise. Results with "*" mean that they are the highest accuracy. The symbol "•" indicates the improvements are statistically significant.

	Fashion-MNIST			
	IDN-20%	IDN-30%	IDN-40%	IDN-50%
CE	92.49 ± 0.13	91.88 ± 0.05	89.76 ± 0.19	68.14 ± 1.67
Decoupling	90.64 ± 0.17	89.84 ± 0.05	89.21 ± 0.24	86.11 ± 0.61
MentorNet	92.38 ± 0.14	91.66 ± 0.04	90.64 ± 0.25	84.33 ± 1.12
Co-Teaching	93.78 ± 0.14*	93.06 ± 0.08*	92.33 ± 0.04*	83.59 ± 5.36
T-Revision	88.35 ± 0.68	86.00 ± 0.45	78.77 ± 3.16	63.92 ± 2.72
Forward	90.31 ± 0.47	87.85 ± 1.42	84.22 ± 1.96	64.99 ± 5.31
Forward-VRNL	**91.78 ± 0.21** •	**89.84 ± 1.70** •	**85.47 ± 1.36** •	**66.09 ± 6.24**
Reweight	89.89 ± 0.45	88.08 ± 0.73	84.29 ± 1.37	71.03 ± 5.88
Reweight-VRNL	**90.81 ± 0.84** •	**89.25 ± 0.50** •	**85.74 ± 1.24** •	**74.85 ± 4.69** •
VolMinNet	92.04 ± 0.17	90.51 ± 0.23	87.73 ± 0.28	80.48 ± 5.97
VolMinNet-VRNL	**92.18 ± 0.08** •	**91.42 ± 0.18** •	**88.31 ± 0.48** •	**81.61 ± 6.24** •
BLTM	91.54 ± 0.37	90.02 ± 1.17	91.08 ± 0.55	89.71 ± 2.46
BLTM-VRNL	**92.19 ± 0.15** •	**91.80 ± 0.26** •	**91.37 ± 0.51**	**91.21 ± 0.97***

Baselines. The baselines used in our experiments: 1). CE, standard Cross-Entropy loss; 2). Decoupling (Malach and Shalev-Shwartz, 2017) trains two models at the same time, and only the instances that have different predictions from two networks are used to update the parameter; 3). MentorNet (Jiang et al., 2018) pre-trains an extra model used to select clean examples for the main model training; 4). Co-teaching (Han et al., 2018b) trains two networks simultaneously, and each network is used to select small-loss examples as trust examples to its peer network for further training; 5). Forward (Patrini et al., 2017) estimates the transition matrix in advance, then uses it to approximate the clean class posteriors; 6). T-Revision (Xia et al., 2019) proposes a method to fine-tune the estimated transition matrix to improve the classification performance; 7). VolMinNet (Li et al., 2021) is an end-to-end label-noise learning method that can learn the transition matrix and the classifier simultaneously; 8). Reweight (Liu and Tao, 2015) uses the importance reweighting technique to

Table 3: Means and standard deviations (percentage) of classification accuracy on CIFAR-10 under class-dependent label noise. Results with "*" mean that they are the highest accuracy. The symbol "•" indicates the improvements are statistically significant.

	CIFAR-10					
	Sym-20%	Sym-50%	Sym-80%	Sym-90%	Pair-20%	Pair-45%
CE	84.12 ± 0.20	72.80 ± 0.75	47.39 ± 0.69	34.43 ± 0.75	85.60 ± 0.06	66.09 ± 0.96
Decoupling	80.90 ± 1.25	69.90 ± 2.37	36.64 ± 19.45	24.93 ± 10.81	79.09 ± 1.04	72.37 ± 0.43
MentorNet	84.84 ± 0.41	76.05 ± 0.98	25.39 ± 1.67	17.87 ± 1.82	84.19 ± 0.24	65.97 ± 0.87
Co-Teaching	88.74 ± 0.19	80.23 ± 0.81	29.57 ± 1.39	16.15 ± 4.37	87.92 ± 0.46	76.65 ± 2.97
T-Revision	87.83 ± 0.63	82.75 ± 0.73	57.61 ± 3.93	20.56 ± 4.26	88.49 ± 0.19	72.81 ± 7.01
Forward	88.43 ± 0.28	80.50 ± 1.46	47.41 ± 2.28	25.44 ± 3.92	**90.04 ± 0.44***	71.59 ± 8.70
Forward-VRNL	**90.16 ± 0.20*** •	**84.59 ± 0.47** •	**53.58 ± 2.05** •	**27.85 ± 4.38** •	89.98 ± 0.58	**72.51 ± 8.14** •
Reweight	88.88 ± 0.28	84.04 ± 0.36	51.70 ± 4.27	24.96 ± 5.68	89.00 ± 0.32	69.38 ± 11.00
Reweight-VRNL	**89.79 ± 0.15** •	**85.37 ± 0.29*** •	**63.24 ± 4.10** •	**34.20 ± 4.76** •	**89.16 ± 0.36**	**70.60 ± 10.57** •
VolMinNet	89.48 ± 0.30	84.18 ± 0.28	56.37 ± 1.91	36.36 ± 1.36	89.26 ± 0.23	81.01 ± 1.43
VolMinNet-VRNL	**89.74 ± 0.19** •	**85.32 ± 0.30** •	**66.65 ± 0.97*** •	**38.72 ± 2.35***	**89.40 ± 0.12**	**85.30 ± 0.27*** •
BLTM	76.71 ± 1.50	64.68 ± 1.55	38.97 ± 3.15	23.43 ± 2.97	77.62 ± 2.87	67.78 ± 1.38
BLTM-VRNL	**78.19 ± 0.50** •	**66.78 ± 0.58** •	**43.14 ± 1.20** •	**28.09 ± 1.48** •	**79.66 ± 0.57** •	**68.57 ± 1.44**

Table 4: Means and standard deviations (percentage) of classification accuracy on CIFAR-10 under instance-dependent label noise. Results with "*" mean that they are the highest accuracy. The symbol "•" indicates the improvements are statistically significant.

	CIFAR-10			
	IDN-20%	IDN-30%	IDN-40%	IDN-50%
CE	84.24 ± 0.35	81.06 ± 0.28	76.43 ± 0.46	59.56 ± 1.48
Decoupling	79.98 ± 1.84	77.62 ± 1.38	72.37 ± 1.17	10.00 ± 0.00
MentorNet	84.32 ± 0.13	81.03 ± 0.18	74.47 ± 1.18	49.48 ± 0.52
Co-Teaching	87.81 ± 0.21	85.75 ± 0.14	80.95 ± 0.52	56.56 ± 0.40
T-Revision	85.79 ± 0.38	82.64 ± 0.84	77.81 ± 0.42	63.23 ± 3.54
Forward	89.02 ± 0.29	87.12 ± 0.54	82.84 ± 2.16	69.05 ± 5.49
Forward-VRNL	**90.16 ± 0.31** •	**88.51 ± 1.53** •	**84.67 ± 2.03** •	**70.12 ± 5.97**
Reweight	87.59 ± 0.45	85.18 ± 0.85	78.34 ± 3.32	68.41 ± 2.25
Reweight-VRNL	**88.21 ± 0.38** •	**85.34 ± 0.84** •	**80.06 ± 2.62** •	**68.91 ± 2.55**
VolMinNet	**90.29 ± 0.18***	89.31 ± 0.27	87.32 ± 0.51	64.90 ± 6.83
VolMinNet-VRNL	90.27 ± 0.12	**89.50 ± 0.18***•	**87.55 ± 0.43***•	**71.34 ± 5.89** •
BLTM	77.03 ± 0.79	72.86 ± 2.49	69.81 ± 2.78	61.87 ± 5.65
BLTM-VRNL	**77.71 ± 1.14**	**76.39 ± 1.56** •	**74.33 ± 1.66** •	**73.41 ± 2.19***•

estimate the expected risk on the clean data; 9). BLTM (Yang et al., 2022) leverages Bayes optimal labels to estimate the Bayes-label transition matrix, and the estimated transition matrix is used to train a classifier by using the Forward algorithm. Note that the aim of this paper is not to design a *state-of-the-art* noisy-label learning algorithm. We want to explore whether the variance of training losses should be penalized for label noise learning. We embed VRNL into baselines: Forward, Reweight, VolMinNet and BLTM. Experiment results show that VRNL can improve their performance in most cases.

Noise Types. To generate a noisy dataset, we corrupted the training and validation sets manually according to a special transition matrix $\boldsymbol{T}$. Specifically, we conduct experiments on synthetic noisy datasets with three widely used types of noise: 1). Symmetry-flipping label noise (Sym-ϵ) (Patrini et al., 2017); 2). Pair-flipping label noise (Pair-ϵ) (Han et al., 2018b); 3). Part-dependent label noise (IDN-ϵ) (Xia et al., 2020).

Table 5: Means and standard deviations (percentage) of classification accuracy on CIFAR-100 under class-dependent label noise. Results with "*" mean that they are the highest accuracy. The symbol "•" indicates the improvements are statistically significant.

	CIFAR-100					
	Sym-20%	Sym-50%	Sym-80%	Sym-90%	Pair-20%	Pair-45%
CE	53.68 ± 0.56	35.64 ± 0.46	14.57 ± 0.57	7.49 ± 1.02	55.96 ± 0.45	35.55 ± 0.32
Decoupling	51.66 ± 2.29	30.54 ± 0.84	12.98 ± 0.86	4.51 ± 2.55	52.27 ± 0.92	35.56 ± 1.97
MentorNet	57.21 ± 0.84	44.17 ± 0.30	12.95 ± 0.87	4.17 ± 0.29	54.87 ± 0.50	30.93 ± 0.43
Co-Teaching	64.02 ± 0.24	48.32 ± 1.07	11.54 ± 0.61	2.09 ± 0.68	60.38 ± 0.92	35.85 ± 0.96
T-Revision	59.84 ± 0.17	46.75 ± 1.27	5.29 ± 1.86	2.09 ± 0.52	59.88 ± 0.11	38.50 ± 0.26
Forward	59.87 ± 0.43	43.45 ± 0.94	17.19 ± 1.18	6.71 ± 0.59	65.13 ± 0.68	44.35 ± 1.05
Forward-VRNL	**67.93 ± 0.39*** •	**56.49 ± 1.45** •	**21.35 ± 0.84** •	**7.84 ± 0.71** •	**65.65 ± 0.38** •	**44.47 ± 1.32**
Reweight	60.70 ± 0.42	46.71 ± 1.95	6.47 ± 1.09	1.40 ± 0.50	63.30 ± 0.46	39.26 ± 0.87
Reweight-VRNL	**66.97 ± 0.53** •	**52.36 ± 1.10** •	**18.53 ± 1.35** •	**5.73 ± 1.83** •	**63.57 ± 0.50** •	**40.15 ± 1.36** •
VolMinNet	65.28 ± 0.61	54.35 ± 0.60	22.05 ± 0.85	9.62 ± 0.75	67.35 ± 0.57	59.27 ± 2.80
VolMinNet-VRNL	**66.33 ± 0.85** •	**57.11 ± 0.85***•	**25.41 ± 0.77***•	**10.37 ± 0.99***	**67.59 ± 0.67***	**60.97 ± 1.71***•
BLTM	45.60 ± 0.90	30.56 ± 0.98	11.80 ± 0.81	5.32 ± 0.37	46.82 ± 1.50	33.06 ± 1.23
BLTM-VRNL	**46.56 ± 0.31** •	**31.32 ± 0.48**	**12.93 ± 0.39** •	**6.04 ± 0.37** •	**48.98 ± 0.55** •	**34.32 ± 0.45** •

Table 6: Means and standard deviations (percentage) of classification accuracy on CIFAR-100. Results with "*" mean that they are the highest accuracy. The symbol "•" indicates the improvements are statistically significant.

	CIFAR-100			
	IDN-20%	IDN-30%	IDN-40%	IDN-50%
CE	54.70 ± 0.43	50.42 ± 0.30	44.39 ± 0.35	34.79 ± 0.26
Decoupling	49.94 ± 1.90	46.64 ± 0.65	40.49 ± 3.47	35.29 ± 1.63
MentorNet	55.99 ± 0.55	51.33 ± 0.23	43.99 ± 0.61	34.05 ± 0.74
Co-Teaching	62.68 ± 0.56	57.60 ± 0.98	50.31 ± 1.08	39.02 ± 1.57
T-Revision	59.85 ± 0.54	53.28 ± 0.22	48.78 ± 3.43	37.57 ± 0.54
Forward	61.27 ± 0.37	56.39 ± 0.59	51.14 ± 0.41	42.18 ± 0.51
Forward-VRNL	**63.48 ± 0.47** •	**61.61 ± 2.93** •	**58.44 ± 0.54** •	**48.30 ± 1.20** •
Reweight	60.63 ± 0.43	55.60 ± 0.78	49.38 ± 0.50	40.81 ± 0.96
Reweight-VRNL	**62.54 ± 1.69** •	**57.00 ± 0.95** •	**50.13 ± 0.56** •	**40.88 ± 0.54**
VolMinNet	68.79 ± 0.34	68.28 ± 0.56	66.99 ± 0.97	64.44 ± 0.82
VolMinNet-VRNL	**69.04 ± 0.48***	**68.75 ± 0.40***•	**67.66 ± 0.29 ***•	**65.73 ± 0.79***•
BLTM	46.37 ± 0.77	42.15 ± 0.45	36.52 ± 1.29	28.73 ± 1.29
BLTM-VRNL	**46.89 ± 0.83** •	**43.20 ± 0.71** •	**37.86 ± 0.75** •	**31.28 ± 1.24** •

Network structure and optimization. We implemented the proposed methods and baselines using PyTorch 1.9.1 and trained the models on an RTX 3090 GPU. For a fair comparison, we use the same model architectures for all baselines and the proposed method. Specifically, we used a ResNet-18 network for FashionMNIST and CIFAR-10, a ResNet-34 network (He et al., 2016) for CIFAR-100, a ResNet-50 pretrained on ImageNet for Clothing1M, and an Inception-ResNet v2 (Szegedy et al., 2017) for WebVision. On synthetic noise datasets, we employed stochastic gradient descent (SGD) to train the neural networks with a batch size of 128, a momentum of 0.9, weight decay of 10^{-4}, and an initial learning rate of 10^{-2}. The models were trained for 80 epochs, and the learning rate was divided by 10 after the 30th and 60th epochs. When the dataset was Clothing1M and WebVision, we used SGD with a batch size of 64, momentum of 0.9, and weight decay of 10^{-4} for the Forward and Reweight methods. For VolMinNet, we used SGD with a batch size of 64,

Table 7: Means and standard deviations (percentage) of classification accuracy on the real-world dataset CIFAR-N.

	CIFAR-10N					CIFAR-100N
	Worst	Aggregate	Random 1	Random 2	Random 3	Noisy Fine
CE	79.39 ± 0.35	87.91 ± 0.18	86.05 ± 0.13	86.12 ± 0.12	86.12 ± 0.16	50.97 ± 0.69
Decoupling	71.94 ± 1.58	82.05 ± 1.84	80.10 ± 0.21	79.74 ± 1.57	79.65 ± 1.82	48.55 ± 0.73
MentorNet	77.91 ± 0.38	75.56 ± 0.25	77.10 ± 0.25	77.06 ± 0.13	77.06 ± 0.13	53.32 ± 0.08
Co-Teaching	81.86 ± 0.40	82.45 ± 0.08	82.90 ± 0.46	82.95 ± 0.26	82.66 ± 0.12	57.08 ± 0.28
T-Revision	82.30 ± 0.96*	89.59 ± 0.14*	87.69 ± 0.47	87.33 ± 0.23	87.49 ± 0.12	53.07 ± 0.54
Forward	78.10 ± 1.25	89.05 ± 0.22	86.82 ± 0.22	86.91 ± 0.11	86.53 ± 0.32	53.59 ± 0.43
Forward-VRNL	**79.45 ± 1.44**	**89.42 ± 0.27**	**88.71 ± 0.36**	**88.52 ± 0.28***	**88.19 ± 0.36**	**55.45 ± 0.58**
Reweight	79.68 ± 0.71	89.12 ± 0.32	87.32 ± 0.27	87.18 ± 0.44	87.05 ± 0.09	53.76 ± 0.18
Reweight-VRNL	**80.96 ± 0.48**	**89.41 ± 0.21**	**88.33 ± 0.44***	**88.03 ± 0.33**	**87.76 ± 0.09**	**54.14 ± 0.25**
VolMinNet	80.53 ± 0.26	89.34 ± 0.14	88.21 ± 0.21	88.10 ± 0.44	88.06 ± 0.31	57.24 ± 0.70
VolMinNet-VRNL	**81.21 ± 0.14**	**89.46 ± 0.35**	**88.32 ± 0.24**	**88.30 ± 0.24**	**88.32 ± 0.32***	**57.44 ± 0.76***
BLTM	68.27 ± 0.46	78.78 ± 0.98	76.94 ± 1.17	77.16 ± 0.85	77.10 ± 1.16	41.54 ± 0.85
BLTM-VRNL	**68.79 ± 1.24**	**80.41 ± 0.75**	**77.20 ± 1.21**	**77.28 ± 0.23**	**77.38 ± 0.62**	**41.62 ± 0.61**

Table 8: Classification accuracy(percentage) on Clothing1M. Only noisy data are exploited for training and validation.

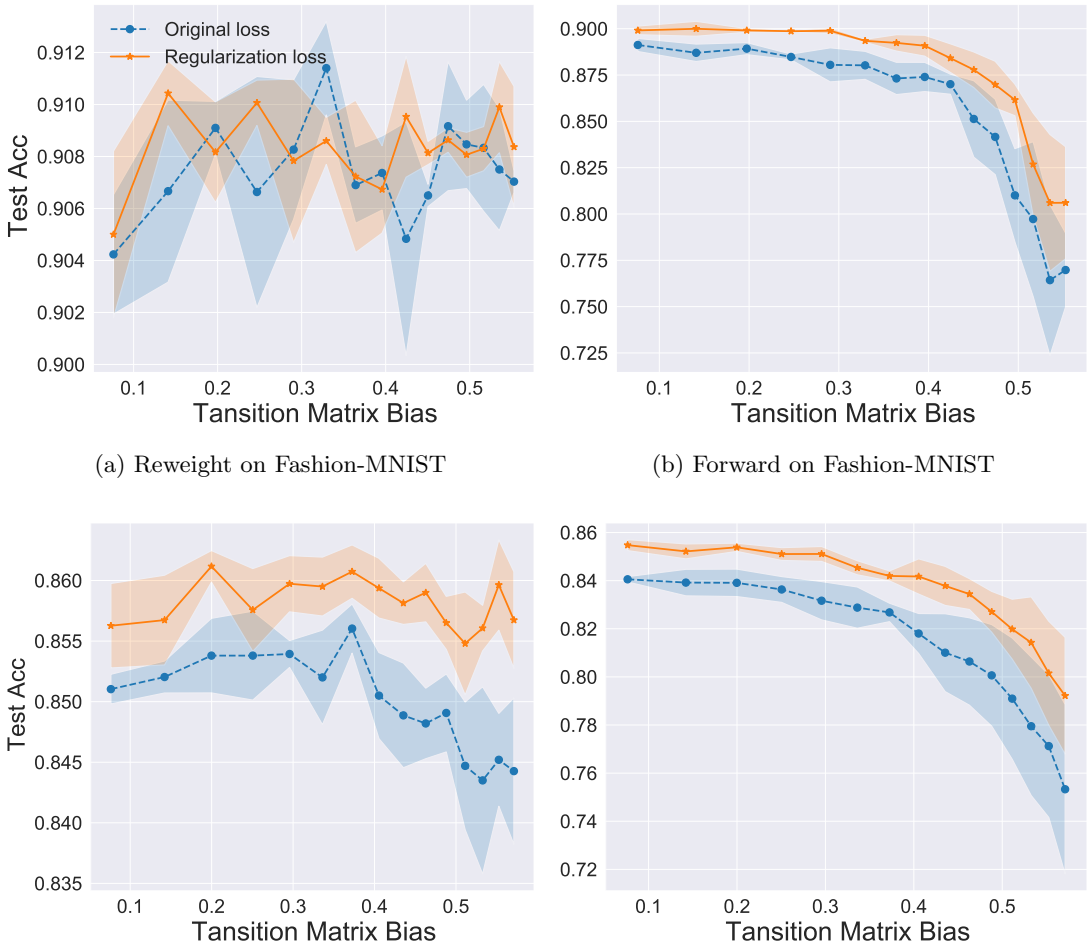
CE	Decoupling	MentorNet	Co-teaching	T-Revision	DMI	Dual T	PTD
69.21	54.53	56.79	60.15	70.97	70.12	71.49	71.67
Forward	Forward-VRNL	Reweight	Reweight-VRNL	VolMinNet	VolMinNet-VRNL	BLTM	BLTM-VRNL
71.27	**72.43**	71.62	**72.14**	72.29	**72.66**	71.37	**72.03**

Table 9: Test accuracy (percentage) on the WebVision validation set and the ImageNet ILSVRC12 validation set.

	WebVision		ILSVRC12	
	top-1	top-5	top-1	top-5
Decoupling	62.54	84.74	58.26	82.26
MentorNet	63.00	81.40	57.80	79.92
Co-Teaching	63.58	85.20	61.48	84.70
Forward	73.36	91.04	68.24	89.80
Forward-VRNL	**73.92**	**91.56**	**68.40**	89.28
Reweight	73.88	90.48	67.84	88.76
Reweight-VRNL	**74.16**	**91.08**	**68.76**	**88.80**
VolMinNet	72.96	90.84	67.24	89.36
VolMinNet-VRNL	**74.16**	**92.04**	**68.56**	89.32
BLTM	59.65	82.91	54.43	82.16
BLTM-VRNL	**60.48**	**84.10**	**56.76**	**83.70**

momentum of 0.9, and weight decay of 10^{-3}. For the BLTM algorithm, we followed the experiment settings in the original paper (Yang et al., 2022).

The hyperparameter α was tuned as follows. On synthetic datasets, α was automatically selected based on the accuracy on the noisy validation set (see Sec. 4.1 and Sec. 4.4 for details), incorporating a 5-epoch linear warm-up from 0 for VolMinNet. On real-world datasets, α was set separately for each method. For Forward and Reweight, α was fixed at 0.1 on CIFAR-10N, Clothing1M, and WebVision, and at 0.05 on CIFAR-100N. For VolMinNet, α was linearly increased from 0 to 0.1 on CIFAR-10N and from 0 to 0.05 on CIFAR-



(a) Reweight on Fashion-MNIST

(b) Forward on Fashion-MNIST

(c) Reweight on CIFAR-10

(d) Forward on CIFAR-10

Figure 4: Test accuracies of the models trained on Fashion-MNIST and CIFAR-10 by using biased transition matrices. The error bar for standard deviation in each figure is shaded. We increase the error of transition matrices manually. The proposed VRNL is robust to the biased transition matrix.

100N; on Clothing1M it was fixed at 0.005, while on WebVision it was gradually increased from 0 to 0.01 during the first 10 epochs. For BLTM, α was fixed at 0.1 on CIFAR-10N, 0.01 on CIFAR-100N, and 0.2 on Clothing1M, and was increased from 0 to 0.1 in the first three epochs on WebVision.

For the Forward, Reweight and BLTM methods, the transition matrix $\boldsymbol{T}$ needed to be estimated in advance. To estimate the transition matrix, we followed the experimental settings described in their original papers (Patrini et al., 2017; Li et al., 2021; Yang et al., 2022). The parameters of the model used to estimate the transition matrix were then used to initialize the weights of the classifier. In the end-to-end methods, VolMinNet, the transition matrix $\boldsymbol{T}$ and the classifier were learned simultaneously.

4.1 Classification Accuracy Evaluation

We embed VRNL into existing statistically consistent algorithms, *e.g.*, Forward, Reweight, VolMinNet and BLTM which are named Forward-VRNL, Reweight-VRNL, VolMinNet-VRNL and BLTM-VRNL, respectively.

In Tab. 1, 2, 3, 4, 5, and 6, we illustrate classification accuracies on datasets containing symmetry-flipping label noise, part-dependent label noise and pair-flipping label noise. The boldface entries in the table denote that the VRNL improves the performance. The hyperparameter α is automatically tuned on noisy validation sets. Specifically, since VRNL is designed for statistically consistent algorithms and its influence on the statistical consistency is small, the suitable hyperparameter α can be automatically tuned by using noisy validation accuracy (more details are in Sec. 4.4). To tune the hyperparameter, α is selected from the candidate set $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.2\}$ by choosing the value that achieves the highest validation accuracy. Each experiment was repeated eight times.

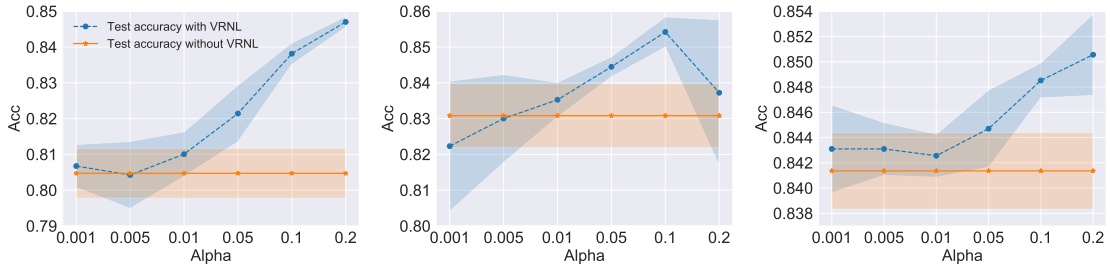
To evaluate statistical significance, we conduct two complementary paired Wilcoxon signed-rank analyses. First, at the *seed level*, for each benchmark setting (i.e., one dataset under one specific noise type and noise rate), we compare the paired results of each baseline method and its VRNL variant across the same 8 random seeds. This analysis evaluates whether VRNL yields consistent improvements across repeated runs within the same setting. Second, at the *benchmark-task level*, we treat each dataset-noise configuration as one benchmark task, average the performance over the 8 seeds for that task, and then perform a paired Wilcoxon signed-rank test across all benchmark tasks. This provides a cross-benchmark statistical significance analysis of VRNL against its paired baseline. To account for multiple comparisons among the four method pairs (Forward, Reweight, VolMinNet, and BLTM), we further apply the Holm–Bonferroni correction (Holm, 1979; Demšar, 2006) to the benchmark-task-level tests.

The results demonstrate that VRNL improves the classification accuracy of all the label-noise learning methods across various datasets and noise types. In particular, VRNL improves the test accuracy of Reweight by 32.11% on Fashion-MNIST under symmetry-flipping label noise with noise rate 0.9. At the seed level, 84.16% of these performance improvements are statistically significant ($p < 0.05$), indicating that VRNL consistently improves the paired baseline across repeated runs within the same benchmark setting. Moreover, at the benchmark-task level, all four method pairs remain statistically significant after Holm–Bonferroni correction. Detailed statistics are reported in Appendix A.

In Tab. 7, 8 and 9, we illustrate the results on the real-world datasets CIFAR-N, Clothing1M and WebVision. VRNL improves the performance of existing statistically consistent algorithms on real-world datasets. The performance of VolMinNet-VRNL outperforms all other baselines.

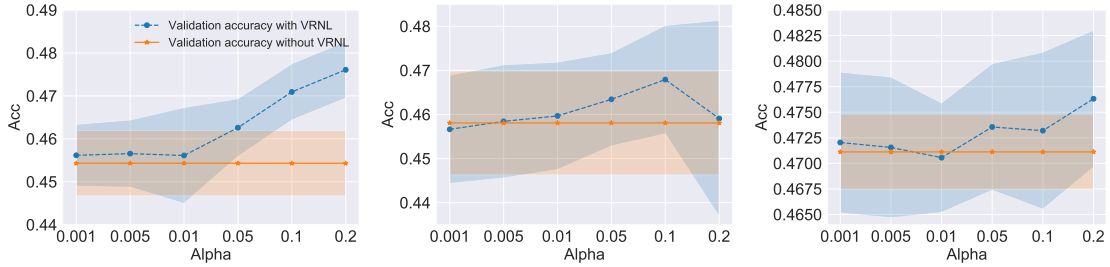
4.2 The Influence on Clean and Noisy Class Posteriors

To analyze the influence of variance of losses on clean class posteriors and noisy class posteriors, we conduct an experiment on CIFAR-10 with symmetry-flipping label noise with a noise rate of 0.5. In Fig. 2 and Fig. 3, we visualize the change of cross-entropy losses for instances with clean labels and instances with noisy labels during the model training, respectively. The algorithms used are Reweight and VolMinNet. By comparing Reweight-



(a) Forward-VRNL (Sym-50%) (b) Reweight-VRNL (Sym-50%) (c) VolMinNet-VRNL (Sym-50%)

Figure 5: Accuracy on the test set for different values of α. The error bar for standard deviation in each figure is shaded.



(a) Forward-VRNL (Sym-50%) (b) Reweight-VRNL (Sym-50%) (c) VolMinNet-VRNL (Sym-50%)

Figure 6: Accuracy on the noisy validation set for different values of α. The error bar for standard deviation in each figure is shaded.

VRNL with Reweight and VolMinNet-VRNL with VolMinNet, the loss of noisy labels for mislabeled examples is larger, but the loss of noisy labels for correctly labeled examples is almost unchanged, as shown in Fig. 2c and Fig. 2d. Therefore, VRNL prevents the model from memorizing incorrect labels and has little influence on learning correctly labeled examples. By comparing Fig. 2b with Fig. 2a, the loss calculated by clean class posterior and clean labels for mislabeled examples becomes smaller by employing our method. It implies that our method helps the model learn clean class posteriors of mislabeled examples.

4.3 Performance with the biased Transition Matrix

In practice, the noise transition matrix generally is not given and is required to be estimated. However, the estimated transition matrix could contain the estimation error. Therefore, we investigate the performance of our regularizer when the transition matrix contains bias. To simulate the estimation error, we manually inject noise into the transition matrix, *i.e.*, $\boldsymbol{T}^{\rho} = \boldsymbol{T} + \gamma|\Delta|$, where $\Delta \in \mathbb{R}^{C\times C}$ sampled from the standard multivariate normal distribution, and $\gamma \in [0.01, 0.15]$. Then we normalize the column of the transition matrix $\boldsymbol{T}_{\rho}$ sum up to 1 by $\boldsymbol{T}^{N}_{ij} = {\boldsymbol{T}^{\rho}}_{ij} / \sum_{k=1}^{C} {\boldsymbol{T}^{\rho}}_{ik}$. The estimation error ϵ_T of a transition matrix is calculated by

Table 10: Means and standard deviations (percentage) of classification accuracy on CIFAR-10 and CIFAR-100 under class-dependent label noise.

	CIFAR-10			CIFAR-100		
	Sym-00%	Sym-05%	Sym-10%	Sym-00%	Sym-05%	Sym-10%
Forward	**92.78 ± 0.30**	90.74 ± 0.50	89.84 ± 0.49	72.35 ± 0.22	68.75 ± 0.24	65.51 ± 0.32
Forward-VRNL	92.71 ± 0.17	**91.34 ± 0.32**	**91.40 ± 0.08**	**72.47 ± 0.15**	**70.70 ± 0.19**	**68.66 ± 0.37**
Reweight	92.69 ± 0.19	91.08 ± 0.19	90.37 ± 0.32	72.21 ± 0.12	69.01 ± 0.23	65.72 ± 0.21
Reweight-VRNL	**92.74 ± 0.08**	**91.77 ± 0.16**	**91.23 ± 0.22**	**72.31 ± 0.37**	**69.88 ± 0.25**	**67.57 ± 3.62**
VolMinNet	91.85 ± 0.12	91.28 ± 0.17	90.60 ± 0.07	70.35 ± 0.86	69.48 ± 0.25	68.27 ± 0.44
VolMinNet-VRNL	**92.04 ± 0.15**	**91.30 ± 0.14**	**90.77 ± 0.12**	**70.99 ± 0.53**	**69.61 ± 0.25**	**68.50 ± 0.40**

employing the entry-wise matrix norm, *i.e.*,

$$\epsilon_{\boldsymbol{T}} = \frac{\|\boldsymbol{T} - \boldsymbol{T}^N\|_{1,1}}{\|\boldsymbol{T}\|_{1,1}}.$$

The biased transition matrix T^N is adopted to Reweight, Reweight-VRNL, Forward and Forward-VRNL, respectively. Experimental results shown in Fig. 4 illustrate that the methods with VRNL are more robust to the biased transition matrix compared with the ones without VRNL. Specifically, for most experiments and different levels of bias ϵ_T, the test accuracies of Reweight-VRNL and Forward-VRNL are higher than Reweight and Forward. Additionally, the test accuracy of Reweight-VRNL drops much slower than Reweight with the increasing of bias ϵ_T.

4.4 Sensitivity Analysis

As mentioned in the previous Section, VRNL is designed for statistically consistent algorithms, and the variance regularizer only has a small influence on the consistency of these algorithms. Thus, we claim that a suitable α can be chosen by employing the noisy validation set. To validate it, we conduct a sensitivity analysis experiment on the synthetic dataset, CIFAR-10, under symmetry-flipping label noise. The noise rate is 50%. We increase the value of α from 0.001 to 0.2 and report the accuracy of the model on the noisy validation set and test set. The experiment results are shown in Fig. 5 and 6. Overall, the curve is smooth, and the VRNL is not sensitive to the hyperparameter. The tendency of noisy validation accuracy is the same as test accuracy. Therefore, the validation set can be used to determine a suitable value of α.

4.5 Performance under Low Noise Rates

To further evaluate the robustness of our approach, we investigate its behavior under low noise rates and on clean data. This analysis is important because the noise rate could be low in practice. It is crucial to ensure that the proposed method does not introduce negative effects when labels are nearly clean.

We report results on CIFAR-10 and CIFAR-100 under symmetry-flipping label noise with noise rates of 0%, 5%, and 10% in Table 10. Two observations can be made. First, when the dataset is clean (0% noise), the VRNL variants achieve performance that is essentially on par with their baselines, confirming that our method does not degrade accuracy

Table 11: Test accuracy (%) on CIFAR-10 with constant α and warmup α under different noise types.

(a) Symmetry and Pair flipping label noise

Method	Sym-20%	Sym-50%	Sym-80%	Sym-90%	Pair-20%	Pair-45%
VolMinNet	89.46 ± 0.18	84.14 ± 0.30	55.89 ± 0.76	35.58 ± 3.13	89.22 ± 0.17	80.73 ± 1.10
VolMinNet-VRNL (constant)	**89.69 ± 0.19**	84.85 ± 0.14	59.88 ± 1.15	34.79 ± 4.04	89.18 ± 0.12	80.69 ± 0.82
VolMinNet-VRNL (warmup)	89.64 ± 0.19	**85.01 ± 0.19**	**61.23 ± 1.79**	**37.41 ± 1.64**	**89.39 ± 0.18**	**81.14 ± 0.92**

(b) Instance-dependent (IDN) noise

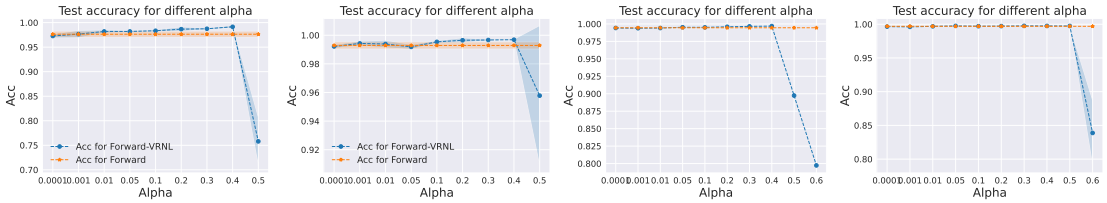
Method	IDN-20%	IDN-30%	IDN-40%	IDN-50%
VolMinNet	89.81 ± 0.15	88.76 ± 0.21	87.53 ± 0.31	64.30 ± 7.02
VolMinNet-VRNL (constant)	**90.17 ± 0.21**	**89.48 ± 0.19**	87.15 ± 0.41	61.48 ± 7.32
VolMinNet-VRNL (warmup)	89.90 ± 0.08	88.94 ± 0.06	**87.58 ± 0.38**	**64.94 ± 6.79**

on the clean dataset (e.g., Forward-VRNL obtains 92.71% compared to 92.78% for Forward on CIFAR-10). Second, under small noise rates (5% and 10%), the VRNL variants consistently outperform the corresponding baselines. For instance, Reweight-VRNL improves upon Reweight by 0.67% at 5% noise and by 1.85% at 10% noise on CIFAR-100.

These results demonstrate that our method is effective not only on large label noise scenarios but also on low label noise scenarios.

4.6 Ablation study

In this subsection, we carry out the ablation study about the warmup strategy, which linearly increases the α from zero gradually, used in the experiment. The warmup strategy can improve the performance of the network trained by using the VolMinNet-VRNL algorithm. Since the parameters of the network are initialized randomly when using the VolMinNet algorithm to train the network, the network might not be able to determine which examples should be large and which should be small. There is no memorization effect at the beginning of training, and the loss of instances with incorrect labels may not be larger than the correct ones. Therefore, the strength of regularization should be zero at the beginning of the training processing and then increase gradually so that the gradients with respect to the losses for correctly labeled examples would not be assigned small weights. For the Forward-VRNL and Reweight-VRNL algorithms, the parameters of the network are obtained through a network after early stopping, which is also used to estimate the transition matrix. Thus, the memorization effect exists at the beginning of training, and the losses for incorrectly labeled examples are usually larger than those for correctly labeled ones. We conducted experiments on the CIFAR-10 dataset, where α was linearly increased from 0 to 0.1 across each mini-batch from epoch 1 to epoch 5. Adjustments to α were made incrementally after each iteration step. The results of these experiments are detailed in Tab. 11. Empirical evidence demonstrates that a warmup of α significantly enhances the classification performance of VolMinNet-VRNL.



(a) 10000 examples (b) 20000 examples (c) 30000 examples (d) 40000 examples

Figure 7: The test accuracy of the model trained with Forward and Forward-VRNL on the synthetic datasets with different sample sizes under symmetry-flipping label noise at a noise rate of 0.8. The error bar for standard deviation is shaded. With the range from 0 to 0.4, our regularizer only has a small influence on the statistical consistency of algorithms.

5. Discussions

In this section, we will first examine the sensitivity of our approach to the hyperparameter α. Next, we analyze the influence of the regularizer on the statistical consistency of algorithms. We then discuss the factors which can affect the memorization effect. Finally, we discuss the influence of VRNL on hard examples.

5.1 Sensitivity Analysis for α

We perform a sensitivity analysis on synthetic data with different sample sizes and find that our method is not sensitive to the hyperparameter α within a width range.

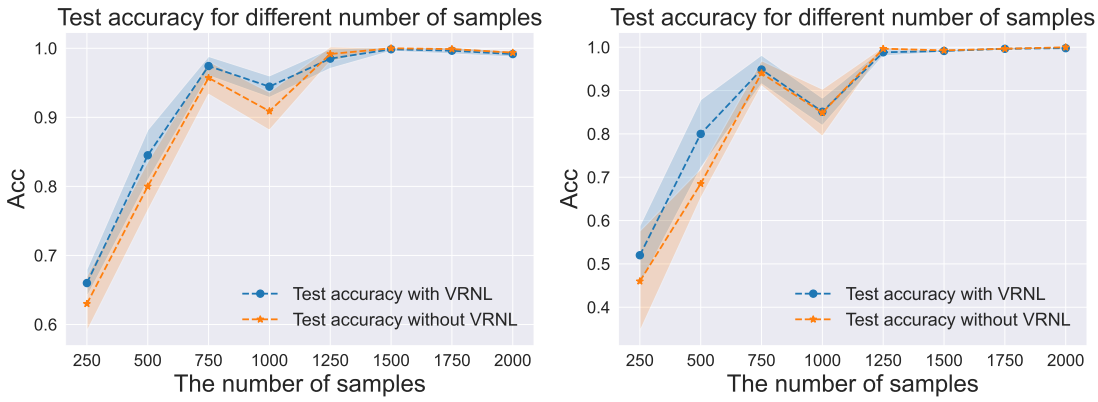
Specifically, we train classification networks using the statistically consistent algorithm Forward with or without using our method on datasets with different sample sizes, respectively. Following a previous work (Yao et al., 2020b), we generate these datasets by sampling from a multidimensional Gaussian distribution involving 10 variables. The datasets contain 10 classes, each distinguished by unique mean values set at 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, while employing identity matrices for their covariance structures. Theoretically, the optimal accuracy, computed using the Gaussian density function, is 99.98%. To introduce label noise, we applied symmetry flipping with a noise rate of 0.8. We trained a two-layer Multi-layer Perceptron (MLP) for label prediction.

The experiment results are shown in Fig. 7, which show that our method is not sensitive to the hyperparameter α within a width range (from 0 to 0.4).

5.2 The Influence on the Statistical Consistency

In this section, we analyze the impact of our regularizer on the statistical consistency of algorithms. Empirical results indicate that the regularizer only has a small influence on the statistical consistency. To evaluate this, we conducted experiments on synthetic datasets with varying sample sizes. The noise type is symmetry flipping with a noise rate of 0.8. A two-layer Multi-Layer Perceptron (MLP) was trained to classify the examples using the Forward algorithm and Forward-VRNL algorithm.

The experimental results are presented in Figure 8. The results indicate that our regularizer does not have a large impact on the statistical consistency of the algorithms. Specifi-



(a) Sym-45%

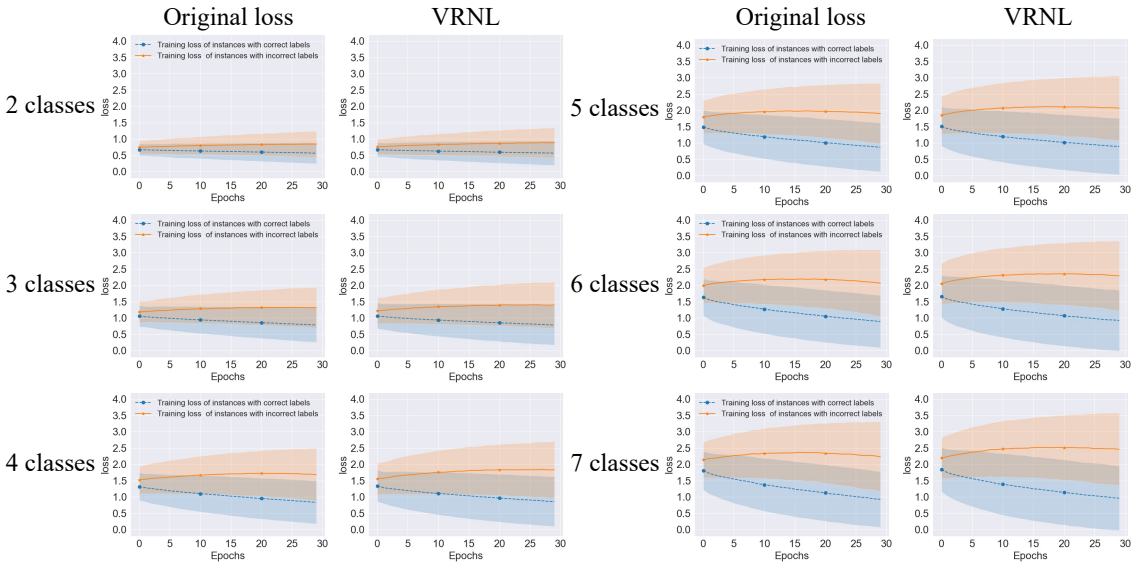
(b) Pair-45%

Figure 8: The test accuracy of the model trained with Forward and Forward-VRNL on datasets with different sample sizes under symmetry-flipping label noise and pair-flipping label noise. When the sample size is large enough, the test accuracy for the method with VRNL is consistent with that without VRNL, which shows that our regularizer only has a small influence on the statistical consistency. The error bar for standard deviation is shaded.

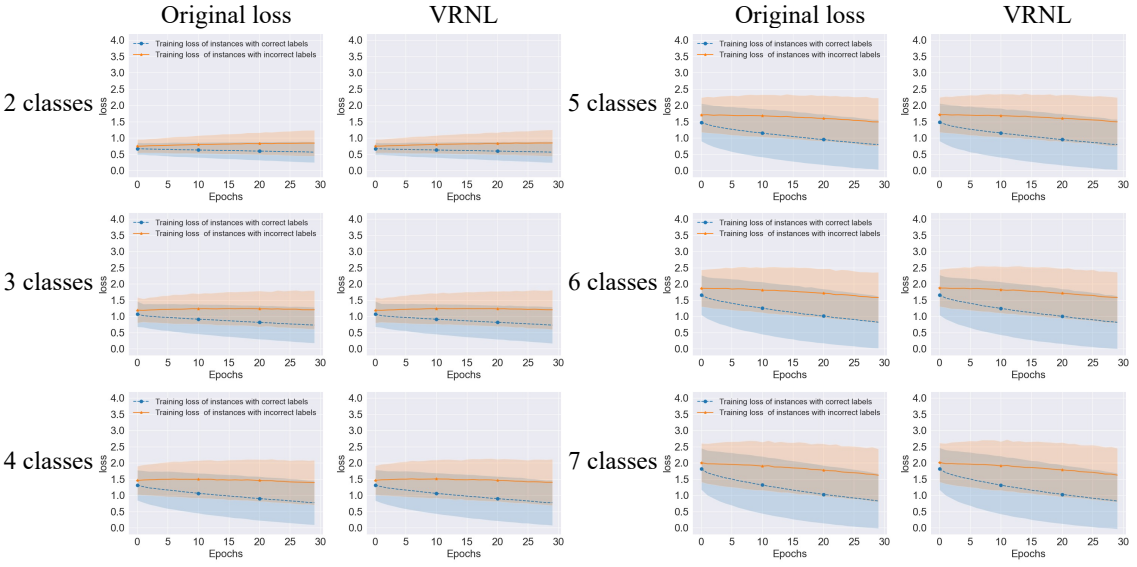
cally, as the training sample size increases, the test accuracies for both the Forward method (without our regularization) and Forward-VRNL (with our regularization) approach the theoretical maximum of 99.98% for the synthetic dataset. Specifically, at a training sample size of 2000, Forward achieves an accuracy of 0.006 below the optimal, while Forward-VRNL's accuracy is 0.008 below the optimal. The performance discrepancy between the algorithm without VRNL (Forward) and with VRNL (Forward-VRNL) is less than 0.002.

5.3 Discussions about the Memorization Effect

In the classification experiment, we found that the improvement of VRNL on pair-flipping label noise is smaller than on symmetry-flipping label noise, and the improvement of VRNL on CIFAR-100 is significantly larger than on CIFAR-10. Since the VRNL relies on the memorization effect, we guess that the memorization effect is related to the noise type and the number of classes. To verify them, we conduct an experiment. We use the 100 classes on CIFAR-100 to obtain datasets with a wide range of class numbers. Specifically, we divide 100 classes into a certain number of classes, *e.g.*, when the number of classes is 2, the instances with labels ranging from 0 to 49 are labeled as 0, and the instances with labels ranging from 50 to 99 are labeled as 1. We get a series of datasets whose class number is from 2 to 7. We use symmetry flipping and pair flipping to corrupt the labels manually. Then we train a ResNet-18 (He et al., 2016) using Forward algorithm (Patrini et al., 2017) on these datasets. We visualize the change of cross-entropy losses for correctly labeled instances and incorrectly labeled instances during the model training in Fig. 9. We also visualize the discrepancy between correctly labeled instances and incorrectly labeled

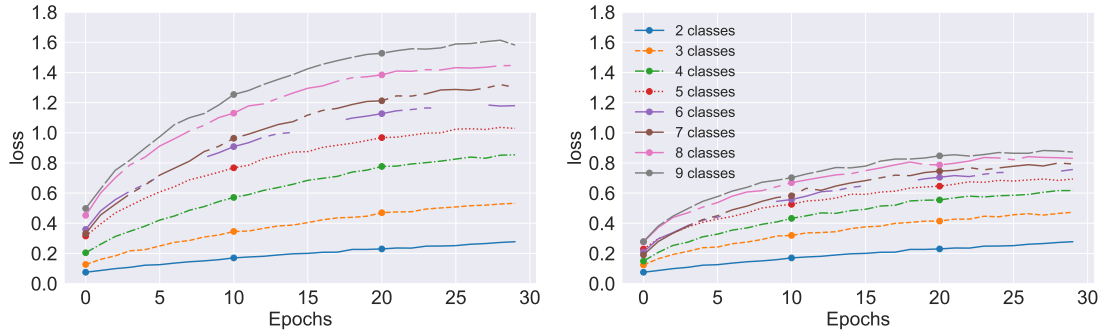


(a) Symmetry-flipping label noise



(b) Pair-flipping label noise

Figure 9: We increase the number of classes gradually, and the discrepancy between the loss of instances with incorrect labels and the loss of instances with correct labels in symmetry-flipping label noise are larger than the discrepancy in pair-flipping label noise. The discrepancy are increasing gradually with the increase in the number of classes. The standard deviation for the training losses in each figure is shaded.



(a) Symmetry-flipping label noise

(b) Pair-flipping label noise

Figure 10: The discrepancy between the loss of instances with incorrect labels and the loss of instances with correct labels in symmetry-flipping label noise and pair-flipping label noise.

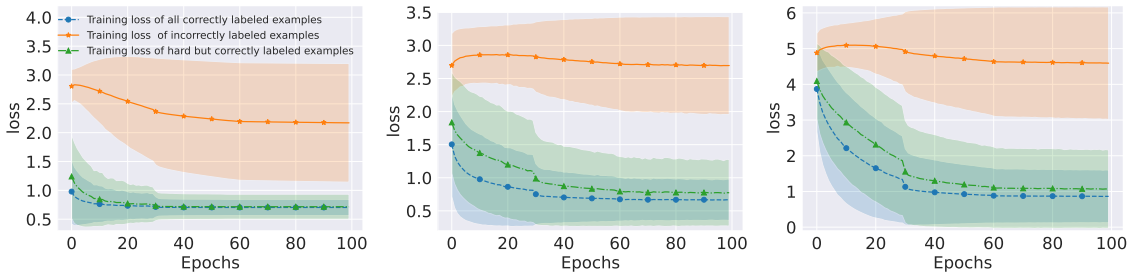
instances in Fig. 10. The results of the original Forward are titled "Original loss", and the results of Forward-VRNL are titled "VRNL".

The experiment results show that the discrepancy between the loss of instances with incorrect labels and the loss of instances with correct labels becomes larger with the increase of class number in both symmetry-flipping and pair-flipping label noise, which indicates that the strength of the memorization effect increases with class number. The experiment results also demonstrate that the discrepancy between the loss of instances with incorrect labels and the loss of instances with correct labels on symmetry flipping are larger than on pair flipping, which indicates that the strength of the memorization effect on symmetry flipping is larger than on pair flipping. The VRNL can make the discrepancy between the loss of instances with incorrect labels and the loss of instances with correct labels become larger on all datasets. Since the strength of the memorization effect on pair-flipping label noise is weaker than on symmetry-flipping label noise, the improvement of VRNL on pair-flipping label noise is smaller than on symmetry-flipping label noise.

Why is the strength of the memorization effect on symmetry-flipping label noise greater? We thought that each noisy class on symmetry-flipping label noise contains the labels flipping from all other classes. The randomness of flipping from different labels makes it difficult for the model to memorize these examples. By contrast, the noisy labels in each class on pair-flipping label noise are flipping from only a class. Thus, it is easy for the model to find the common features and memorize these examples.

5.4 The Influence on Hard Examples

In practice, some examples are hard to be memorized but correctly labeled. It is challenging for models to distinguish had examples from incorrectly labeled examples because the losses of those examples often are usually high. The VRNL might have a negative effect on the memorization of hard but correctly labeled examples for models because a small weight would be assigned to the gradients with respect to the hard but correctly labeled examples. There are two cases: 1). the loss of incorrectly labeled examples equals the loss of hard



(a) Fashion-MNIST (b) CIFAR-10 (c) CIFAR-100

Figure 11: The influence of VRNL on hard examples. The training loss of incorrectly labeled examples is larger than the training loss of hard but correctly labeled examples. The standard deviation for the training losses is shaded.

but correctly labeled examples (*i.e.*, these examples are entangled). It is hard to separate hard but correctly labeled examples from incorrectly labeled examples. In such a case, all existing small-loss-based sample selection and reweighting methods would have the same problem; 2). the loss of incorrectly labeled examples is larger than the loss of hard but correctly labeled examples (*e.g.*, the number of hard but correctly labeled examples is more than the number of incorrectly labeled examples), VRNL should not have a large negative impact on hard but correctly labeled examples because VRNL can still separate hard but correctly labeled examples from incorrectly labeled examples.

We conducted an experiment on Fashion-MNIST, CIFAR-10, and CIFAR-100 to investigate which cases occur in the real world. Specifically, we train a classification network on the clean dataset for 50 epochs and sort the cross-entropy loss of all training examples. The top 30% large-loss examples are defined as hard examples. Then, we corrupt all training examples manually by using 50% symmetry-flipping label noise and train a new classification network using the Forward-VRNL algorithm. We visualize the losses of all correctly labeled examples, the losses of incorrectly labeled examples, and the losses of hard but correctly labeled examples. The visualized result is shown in Fig. 11. The experiment result indicates that the loss of incorrectly labeled examples is larger than the loss of hard but correctly labeled examples. Thus, the VRNL can still separate hard, correctly labeled examples from incorrectly labeled examples and does not have a large negative impact on the memorization of hard but correctly labeled examples.

6. Applicability and Limitations

Our regularization is mainly designed for statistically consistent algorithms. Within this algorithmic scope, the effectiveness of our method relies on two conditions: **1). Existence of the Memorization Effect:** Our approach hinges on the widely observed phenomenon that deep networks tend to learn simple patterns (correctly labeled examples) before memorizing noise (incorrectly labeled examples) (Arpit et al., 2017; Han et al., 2018b). Consequently, in learning with noisy labels, incorrectly labeled examples typically exhibit larger losses than correctly labeled ones during the early stages of training. VRNL can boost the mem-

orization effect by encouraging a large variance of losses, resulting in preventing the loss for large-loss examples, which are likely incorrectly labeled, from decreasing. Thus, our method can prevent the model from memorizing incorrectly labeled examples and improve the performance. **2). Presence of Label Noise:** VRNL is designed to mitigate the negative effects of label noise. Empirically, the performance gains are large when the dataset contains a non-negligible amount of label noise. In scenarios with low or zero noise, the regularization is less essential. However, as demonstrated in our experiments, properly tuning the strength α ensures that our method does not degrade performance in clean settings.

One limitation of our method is that it is time-consuming when tuning the hyperparameter α automatically on a noisy validation set. A grid search is required to determine the suitable value. This procedure increases the computational cost.

7. Conclusion

In this paper, we study whether we should penalize the variance of losses for label-noise learning. Interestingly, we found that encouraging a large variance of losses could be helpful, as it can boost the memorization effect and reduce the harmfulness of incorrect labels. Theoretically, we show that encouraging a large variance of losses can reduce the weights of the gradient with respect to incorrectly labeled examples. Therefore, these examples have a small contribution to parameter updates. A simple and effective method, VRNL, is also proposed, which can be easily integrated into existing label-noise learning methods to improve their robustness. The experimental results on both synthetic and real-world noisy datasets demonstrate that VRNL can dramatically improve the performance of existing label-noise learning methods. Empirically, we have shown that the proposed method can help models better learn clean class posteriors. We have also illustrated that VRNL can improve the classification performance of existing methods when the transition matrix is poorly estimated, which makes our method be practically useful.

Acknowledgments

The work of Jun Yu was supported by the Natural Science Foundation of China (62276242), Hefei Municipal Natural Science Foundation (HZR2431), CAAI-MindSpore Open Fund, developed on OpenI Community. The work of Bo Han was supported by RGC Young Collaborative Research Grant No. C2005-24Y and RGC General Research Fund No. 12200725. The work of Mingming Gong was supported by ARC DP240102088 and WIS-MBZUAI 142571. The work of Tongliang Liu was supported by the following Australian Research Council projects: FT220100318, DP220102121, LP220100527, LP220200949, and IC190100031.

Appendix A. Statistical Significance Analysis

To assess whether the observed improvements are statistically significant, we conduct two complementary one-sided paired Wilcoxon signed-rank analyses. The Wilcoxon signed-rank test is a non-parametric paired test and is suitable here because it compares matched results without assuming that the performance differences follow a normal distribution.

Table 12: One-sided Wilcoxon signed-rank test p values under different noise levels (Symmetry-flipping and Pair-flipping label noise).

Dataset	Method	Sym-20%-p	Sym-50%-p	Sym-80%-p	Sym-90%-p	Pair-20%-p	Pair-45%-p
Fashion-MNIST	Forward	**0.0039**	**0.0039**	**0.0039**	**0.0039**	**0.0195**	**0.0078**
	Reweight	**0.0195**	**0.0195**	**0.0039**	**0.0039**	0.0977	**0.0391**
	VolMinNet	**0.0078**	**0.0039**	**0.0391**	**0.0195**	**0.0039**	0.0538
	BLTM	**0.0391**	**0.0039**	**0.0195**	**0.0039**	**0.0039**	**0.0391**
CIFAR-10	Forward	**0.0039**	**0.0039**	**0.0039**	**0.0078**	0.8086	**0.0391**
	Reweight	**0.0039**	**0.0039**	**0.0039**	**0.0078**	0.1914	**0.0273**
	VolMinNet	**0.0273**	**0.0039**	**0.0039**	0.0547	0.0547	**0.0039**
	BLTM	**0.0078**	**0.0039**	**0.0117**	**0.0039**	**0.0195**	0.1250
CIFAR-100	Forward	**0.0039**	**0.0039**	**0.0039**	**0.0039**	**0.0391**	0.3677
	Reweight	**0.0039**	**0.0039**	**0.0039**	**0.0078**	**0.0391**	**0.0273**
	VolMinNet	**0.0117**	**0.0039**	**0.0039**	**0.0742**	0.0547	**0.0195**
	BLTM	**0.0117**	0.0547	**0.0117**	**0.0078**	**0.0078**	**0.0273**

Table 13: One-sided Wilcoxon signed-rank test p values under different noise levels (Instance-dependent noise).

Dataset	Method	Ins-20%-p	Ins-30%-p	Ins-40%-p	Ins-50%-p
Fashion-MNIST	Forward	**0.0039**	**0.0039**	**0.0117**	0.3203
	Reweight	**0.0090**	**0.0039**	**0.0391**	**0.0273**
	VolMinNet	**0.0315**	**0.0039**	**0.0039**	**0.0039**
	BLTM	**0.0039**	**0.0039**	0.1562	0.0977
CIFAR-10	Forward	**0.0039**	**0.0391**	**0.0039**	0.1250
	Reweight	**0.0078**	**0.0117**	**0.0391**	0.6797
	VolMinNet	0.6289	**0.0391**	**0.0315**	**0.0078**
	BLTM	0.1562	**0.0039**	**0.0117**	**0.0039**
CIFAR-100	Forward	**0.0039**	**0.0039**	**0.0039**	**0.0039**
	Reweight	**0.0391**	**0.0039**	**0.0273**	0.3711
	VolMinNet	0.1562	**0.0391**	**0.0391**	**0.0039**
	BLTM	**0.0391**	**0.0039**	**0.0195**	**0.0039**

First, we perform a *seed-level* analysis. For each benchmark setting (i.e., one dataset under one specific noise type and noise rate), we compare the paired results of each baseline method and its VRNL variant across the same 8 random seeds. This analysis evaluates whether VRNL yields consistent improvements across repeated runs within the same setting. The corresponding p-values are reported in Tab. 12 and Tab. 13, where bold values indicate statistical significance at the 5% level ($p < 0.05$).

Table 14: Benchmark-task-level paired Wilcoxon signed-rank tests across 30 dataset-noise configurations, with Holm–Bonferroni correction over four paired comparisons.

Comparison	Wins/Losses	Raw p	Corrected p	Significant
Forward-VRNL vs Forward	29/1	1.86×10^{-9}	3.73×10^{-9}	Yes
Reweight-VRNL vs Reweight	30/0	9.31×10^{-10}	3.73×10^{-9}	Yes
VolMinNet-VRNL vs VolMinNet	29/1	1.86×10^{-9}	3.73×10^{-9}	Yes
BLTM-VRNL vs BLTM	30/0	9.31×10^{-10}	3.73×10^{-9}	Yes

Second, we perform a *benchmark-task-level* analysis. Here, each dataset-noise configuration is treated as one benchmark task. For each task, we first average the performance over the 8 random seeds, and then conduct a paired Wilcoxon signed-rank test across all benchmark tasks for each method pair. To account for multiple comparisons among the four method pairs, we further apply the Holm–Bonferroni correction (Holm, 1979; Demšar, 2006). The detailed benchmark-task-level results are reported in Tab. 14.

Overall, the statistical significance analysis supports the effectiveness of VRNL at both the seed level and the benchmark-task level. At the seed level, 84.16% of the paired comparisons yield significant p-values. At the benchmark-task level, all four method pairs remain statistically significant after Holm–Bonferroni correction.

Appendix B. Results with Manually Preset α

Table 15: Means and standard deviations (percentage) of classification accuracy on Fashion-MNIST under class-dependent label noise. Results with "*" mean that they are the highest accuracy.

	Fashion-MNIST					
	Sym-20%	Sym-50%	Sym-80%	Sym-90%	Pair-20%	Pair-45%
CE	92.22 ± 0.07	89.95 ± 0.22	81.52 ± 0.26	74.26 ± 0.85	92.78 ± 0.15	82.44 ± 3.85
Decoupling	90.23 ± 0.10	87.76 ± 0.71	78.65 ± 0.79	15.79 ± 7.95	91.52 ± 0.40	90.15 ± 0.27
MentorNet	92.28 ± 0.18	90.42 ± 0.10	78.90 ± 0.61	50.21 ± 4.20	92.07 ± 0.13	89.28 ± 0.18
Co-Teaching	93.63 ± 0.09*	91.84 ± 0.17*	81.15 ± 0.50	58.34 ± 3.31	93.64 ± 0.04*	91.38 ± 0.53
T-Revision	91.27 ± 0.61	85.35 ± 1.45	75.54 ± 4.55	59.89 ± 9.02	92.42 ± 0.37	71.74 ± 11.99
Forward	90.87 ± 0.32	87.54 ± 0.50	76.30 ± 1.42	54.65 ± 4.59	91.67 ± 0.27	62.76 ± 6.47
Forward-VRNL	**92.26 ± 0.12**	**88.84 ± 0.44**	**78.37 ± 2.06**	**60.94 ± 4.49**	**92.19 ± 0.41**	**63.30 ± 5.52**
Reweight	91.17 ± 0.45	88.43 ± 0.51	82.83 ± 1.54	58.15 ± 9.91	90.99 ± 0.39	76.38 ± 11.56
Reweight-VRNL	**91.62 ± 0.37**	**89.65 ± 0.35**	**84.32 ± 1.09***	**77.86 ± 1.70***	**91.30 ± 0.41**	**79.22 ± 9.74**
VolMinNet	91.95 ± 0.19	89.06 ± 0.51	81.48 ± 1.07	67.86 ± 2.11	92.50 ± 0.44	92.69 ± 0.39
VolMinNet-VRNL	**92.16 ± 0.17**	**89.80 ± 0.28**	**82.24 ± 0.67**	**69.47 ± 2.77**	**92.63 ± 0.38**	**92.75 ± 0.38***
BLTM	92.03 ± 0.24	**89.47 ± 0.41**	81.23 ± 1.94	68.23 ± 6.01	91.98 ± 0.10	90.04 ± 0.07
BLTM-VRNL	**92.05 ± 0.16**	89.23 ± 1.30	**83.83 ± 0.79**	**69.52 ± 2.13**	**92.07 ± 0.10**	**90.27 ± 0.27**

Table 16: Means and standard deviations (percentage) of classification accuracy on Fashion-MNIST under instance-dependent label noise. Results with "*" mean that they are the highest accuracy.

	Fashion-MNIST			
	IDN-20%	IDN-30%	IDN-40%	IDN-50%
CE	92.49 ± 0.13	91.88 ± 0.05	89.76 ± 0.19	68.14 ± 1.67
Decoupling	90.64 ± 0.17	89.84 ± 0.05	89.21 ± 0.24	86.11 ± 0.61
MentorNet	92.38 ± 0.14	91.66 ± 0.04	90.64 ± 0.25	84.33 ± 1.12
Co-Teaching	93.78 ± 0.14*	93.06 ± 0.08*	92.33 ± 0.04*	83.59 ± 5.36
T-Revision	88.35 ± 0.68	86.00 ± 0.45	78.77 ± 3.16	63.92 ± 2.72
Forward	89.55 ± 0.66	87.22 ± 1.59	76.84 ± 3.51	63.40 ± 5.93
Forward-VRNL	**91.40 ± 0.50**	**89.07 ± 0.84**	**79.05 ± 2.49**	**66.42 ± 7.41**
Reweight	89.90 ± 0.26	85.73 ± 4.54	78.43 ± 5.51	63.61 ± 3.63
Reweight-VRNL	**90.74 ± 0.19**	**87.63 ± 1.62**	**80.23 ± 5.53**	**67.10 ± 6.85**
VolMinNet	91.92 ± 0.16	90.67 ± 0.35	84.25 ± 2.06	72.12 ± 5.10
VolMinNet-VRNL	**92.01 ± 0.11**	**91.01 ± 0.19**	**85.20 ± 1.54**	**74.21 ± 3.63**
BLTM	91.93 ± 0.19	91.08 ± 0.28	90.65 ± 0.56	88.84 ± 4.31
BLTM-VRNL	**92.17 ± 0.19**	**91.60 ± 0.34**	**90.91 ± 0.30**	**90.05 ± 0.36***

Table 17: Means and standard deviations (percentage) of classification accuracy on CIFAR-10 under class-dependent label noise. Results with "*" mean that they are the highest accuracy.

	CIFAR-10					
	Sym-20%	Sym-50%	Sym-80%	Sym-90%	Pair-20%	Pair-45%
CE	84.12 ± 0.20	72.80 ± 0.75	47.39 ± 0.69	34.43 ± 0.75	85.60 ± 0.06	66.09 ± 0.96
Decoupling	80.90 ± 1.25	69.90 ± 2.37	36.64 ± 19.45	24.93 ± 10.81	79.09 ± 1.04	72.37 ± 0.43
MentorNet	84.84 ± 0.41	76.05 ± 0.98	25.39 ± 1.67	17.87 ± 1.82	84.19 ± 0.24	65.97 ± 0.87
Co-Teaching	88.74 ± 0.19	80.23 ± 0.81	29.57 ± 1.39	16.15 ± 4.37	87.92 ± 0.46	76.65 ± 2.97
T-Revision	87.83 ± 0.63	82.75 ± 0.73	57.61 ± 3.93	20.56 ± 4.26	88.49 ± 0.19	72.81 ± 7.01
Forward	88.31 ± 0.23	80.70 ± 0.59	47.18 ± 4.63	23.60 ± 2.37	89.53 ± 0.81	76.76 ± 4.94
Forward-VRNL	**90.05 ± 0.24***	**83.81 ± 0.30**	**50.01 ± 5.26**	**25.90 ± 2.26**	**89.55 ± 0.69***	**77.44 ± 5.28**
Reweight	88.19 ± 0.33	80.31 ± 1.31	44.23 ± 3.60	26.77 ± 2.61	89.14 ± 0.33	70.66 ± 4.29
Reweight-VRNL	**89.96 ± 0.14**	**83.66 ± 0.57**	**47.12 ± 4.14**	**28.91 ± 1.93**	**89.18 ± 0.32**	**70.73 ± 4.63**
VolMinNet	89.46 ± 0.18	84.14 ± 0.30	55.89 ± 0.76	35.58 ± 3.13	89.22 ± 0.17	80.73 ± 1.10
VolMinNet-VRNL	**89.64 ± 0.19**	**85.01 ± 0.19***	**61.23 ± 1.79***	**37.41 ± 1.64***	**89.39 ± 0.18**	**81.14 ± 0.92***
BLTM	76.54 ± 1.37	63.50 ± 1.78	39.28 ± 4.00	**23.70 ± 3.36**	**76.97 ± 0.67**	67.97 ± 1.45
BLTM-VRNL	**77.68 ± 0.42**	**65.33 ± 1.95**	**40.41 ± 1.33**	**23.70 ± 3.21**	76.25 ± 0.84	**68.53 ± 0.76**

Table 18: Means and standard deviations (percentage) of classification accuracy on CIFAR-10 under instance-dependent label noise. Results with "*" mean that they are the highest accuracy.

	CIFAR-10			
	IDN-20%	IDN-30%	IDN-40%	IDN-50%
CE	84.24 ± 0.35	81.06 ± 0.28	76.43 ± 0.46	59.56 ± 1.48
Decoupling	79.98 ± 1.84	77.62 ± 1.38	72.37 ± 1.17	10.00 ± 0.00
MentorNet	84.32 ± 0.13	81.03 ± 0.18	74.47 ± 1.18	49.48 ± 0.52
Co-Teaching	87.81 ± 0.21	85.75 ± 0.14	80.95 ± 0.52	56.56 ± 0.40
T-Revision	85.79 ± 0.38	82.64 ± 0.84	77.81 ± 0.42	63.23 ± 3.54
Forward	88.71 ± 0.28	86.33 ± 0.99	83.94 ± 2.44	69.46 ± 12.83
Forward-VRNL	**89.91 ± 0.17***	**87.82 ± 0.57**	**84.87 ± 2.40**	**74.76 ± 5.89***
Reweight	88.29 ± 0.64	84.12 ± 1.24	74.76 ± 4.43	61.48 ± 3.46
Reweight-VRNL	**88.88 ± 0.35**	**85.41 ± 0.50**	**77.41 ± 3.91**	**62.75 ± 3.97**
VolMinNet	89.81 ± 0.15	88.76 ± 0.21	87.53 ± 0.31	64.30 ± 7.02
VolMinNet-VRNL	**89.90 ± 0.08**	**88.94 ± 0.06***	**87.58 ± 0.38***	**64.94 ± 6.79**
BLTM	76.44 ± 1.06	73.07 ± 1.27	69.19 ± 1.77	64.68 ± 2.16
BLTM-VRNL	**77.43 ± 2.15**	**74.67 ± 0.67**	**71.42 ± 0.43**	**66.28 ± 0.73**

Automatically tuning the hyperparameter α using validation sets is time-consuming since it requires grid search. Therefore, we also conduct experiments with manually preset α. Empirically, VRNL with manually preset α still improves the performance of statistically consistent algorithms.

Specifically, the hyperparameter α was set as follows. For the Forward and Reweight methods, we kept the hyperparameter α constant at 0.1 on symmetry-flipping label noise and part-dependent noise, and we kept the hyperparameter α constant at 0.01 on pair-flipping label noise. For VolMinNet, we implemented a gradual increase in the hyperparameter α during the first five epochs for different types of noise. For symmetry-flipping label noise, α was linearly increased from 0 to 0.1. For both pair-flipping label noise and part-dependent noise, α was linearly increased from 0 to 0.01. Throughout this process, α was adjusted incrementally after each iteration step. After the initial five epochs, the value of α was held constant. For the BLTM method, we kept the hyperparameter α constant at 0.1 on symmetry-flipping label noise and pair-flipping label noise, and we kept the hy-

Table 19: Means and standard deviations (percentage) of classification accuracy on CIFAR-100 under class-dependent label noise. Results with "*" mean that they are the highest accuracy.

	CIFAR-100					
	Sym-20%	Sym-50%	Sym-80%	Sym-90%	Pair-20%	Pair-45%
CE	53.68 ± 0.56	35.64 ± 0.46	14.57 ± 0.57	7.49 ± 1.02	55.96 ± 0.45	35.55 ± 0.32
Decoupling	51.66 ± 2.29	30.54 ± 0.84	12.98 ± 0.86	4.51 ± 2.55	52.27 ± 0.92	35.56 ± 1.97
MentorNet	57.21 ± 0.84	44.17 ± 0.30	12.95 ± 0.87	4.17 ± 0.29	54.87 ± 0.50	30.93 ± 0.43
Co-Teaching	64.02 ± 0.24	48.32 ± 1.07	11.54 ± 0.61	2.09 ± 0.68	60.38 ± 0.92	35.85 ± 0.96
T-Revision	59.84 ± 0.17	46.75 ± 1.27	5.29 ± 1.86	2.09 ± 0.52	59.88 ± 0.11	38.50 ± 0.26
Forward	59.98 ± 0.45	44.50 ± 0.96	16.00 ± 2.04	7.60 ± 0.56	**65.46 ± 0.41**	40.49 ± 4.65
Forward-VRNL	**67.52 ± 0.78**	**54.28 ± 0.53**	**19.65 ± 1.67**	**8.28 ± 0.69**	65.40 ± 0.41	**42.14 ± 2.06**
Reweight	59.64 ± 0.40	43.52 ± 1.41	17.58 ± 0.87	7.39 ± 0.95	63.21 ± 0.31	39.19 ± 0.42
Reweight-VRNL	**67.66 ± 0.25***	**53.69 ± 0.32**	**19.99 ± 0.94**	**7.82 ± 0.94**	**63.52 ± 0.33**	**40.16 ± 0.39**
VolMinNet	65.51 ± 0.30	54.44 ± 0.36	23.03 ± 0.96	9.13 ± 1.06	**67.58 ± 0.87***	59.04 ± 1.87
VolMinNet-VRNL	**66.44 ± 0.74**	**56.77 ± 0.21***	**24.80 ± 1.23***	**10.85 ± 0.65***	67.57 ± 0.56	**59.33 ± 0.35***
BLTM	46.11 ± 1.19	**30.47 ± 0.99**	12.18 ± 0.50	5.37 ± 0.36	45.37 ± 0.65	33.45 ± 0.86
BLTM-VRNL	**46.29 ± 0.74**	29.92 ± 0.78	**12.27 ± 0.35**	**5.88 ± 0.31**	**46.11 ± 0.79**	**33.63 ± 0.53**

Table 20: Means and standard deviations (percentage) of classification accuracy on CIFAR-100. Results with "*" mean that they are the highest accuracy.

	CIFAR-100			
	IDN-20%	IDN-30%	IDN-40%	IDN-50%
CE	54.70 ± 0.43	50.42 ± 0.30	44.39 ± 0.35	34.79 ± 0.26
Decoupling	49.94 ± 1.90	46.64 ± 0.65	40.49 ± 3.47	35.29 ± 1.63
MentorNet	55.99 ± 0.55	51.33 ± 0.23	43.99 ± 0.61	34.05 ± 0.74
Co-Teaching	62.68 ± 0.56	57.60 ± 0.98	50.31 ± 1.08	39.02 ± 1.57
T-Revision	59.85 ± 0.54	53.28 ± 0.22	48.78 ± 3.43	37.57 ± 0.54
Forward	60.98 ± 0.47	55.34 ± 0.69	50.32 ± 0.72	42.91 ± 1.08
Forward-VRNL	**65.41 ± 2.97**	**63.66 ± 1.16**	**58.82 ± 0.74**	**49.61 ± 1.25**
Reweight	61.30 ± 0.30	55.31 ± 0.35	46.48 ± 1.50	37.67 ± 2.23
Reweight-VRNL	**63.51 ± 0.35**	**56.71 ± 0.59**	**47.96 ± 1.31**	**39.99 ± 1.15**
VolMinNet	**68.91 ± 0.40***	68.12 ± 0.19	67.35 ± 0.47	65.49 ± 0.83
VolMinNet-VRNL	68.48 ± 0.38	**68.53 ± 0.22***	**67.40 ± 0.76***	**65.90 ± 1.02***
BLTM	45.44 ± 0.43	41.20 ± 1.01	**35.48 ± 0.74**	29.76 ± 0.79
BLTM-VRNL	**45.81 ± 0.71**	**41.75 ± 0.16**	35.20 ± 0.87	**29.92 ± 0.81**

perparameter α constant at 0.2 on part-dependent noise. We repeat the experiments five times.

The experiment results are shown in Tab. 15, 16, 17, 18, 19, and 20, which demonstrates that VRNL can still improve the classification performance when manually setting α.

References

Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4): 343–370, 1988.

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR, 2017.

HeeSun Bae, Seungjae Shin, Byeonghu Na, JoonHo Jang, Kyungwoo Song, and Il-Chul Moon. From noisy prediction to true label: Noisy prediction calibration via generative model. In *International Conference on Machine Learning*, pages 1277–1297. PMLR, 2022.

HeeSun Bae, Seungjae Shin, Byeonghu Na, and Il-Chul Moon. Dirichlet-based per-sample weighting by transition matrix for noisy label learning. *arXiv preprint arXiv:2403.02690*, 2024.

Yingbin Bai, Erkun Yang, Bo Han, Yanhua Yang, Jiatong Li, Yinian Mao, Gang Niu, and Tongliang Liu. Understanding and improving early stopping for learning with noisy labels. *Advances in Neural Information Processing Systems*, 34, 2021.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019.

Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, pages 1062–1070. PMLR, 2019.

De Cheng, Tongliang Liu, Yixiong Ning, Nannan Wang, Bo Han, Gang Niu, Xinbo Gao, and Masashi Sugiyama. Instance-dependent label-noise learning with manifold-regularized transition matrix estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16630–16639, 2022.

Jiacheng Cheng, Tongliang Liu, Kotagiri Ramamohanarao, and Dacheng Tao. Learning with bounded instance and label-dependent label noise. In *International Conference on Machine Learning*, pages 1789–1799. PMLR, 2020.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Erik Englesson and Hossein Azizpour. Robust classification via regression for learning with noisy labels. In *ICLR 2024-The Twelfth International Conference on Learning Representations, Messe Wien Exhibition and Congress Center, Vienna, Austria, May 7-11t, 2024*, 2024.

Lei Feng, Senlin Shu, Zhuoyi Lin, Fengmao Lv, Li Li, and Bo An. Can cross entropy loss be robust to label noise? In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*, pages 2206–2212, 2021.

Arpit Garg, Cuong Nguyen, Rafael Felix, Thanh-Toan Do, and Gustavo Carneiro. Instance-dependent noisy label learning via graphical modelling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2288–2298, 2023.

Arpit Garg, Cuong Nguyen, Rafael Felix, Yuyuan Liu, Thanh-Toan Do, and Gustavo Carneiro. Aeon: Adaptive estimation of instance-dependent in-distribution and out-of-distribution label noise for robust learning. *arXiv preprint arXiv:2501.13389*, 2025.

Aritra Ghosh, Naresh Manwani, and PS Sastry. Making risk minimization tolerant to label noise. *Neurocomputing*, 160:93–107, 2015.

Aritra Ghosh, Himanshu Kumar, and P Shanti Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

Hui Guo, Boyu Wang, and Grace Yi. Label correction of crowdsourced noisy annotations with an instance-dependent noise transition model. *Advances in Neural Information Processing Systems*, 36:347–386, 2023.

Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150, 2018.

Bo Han, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Ivor Tsang, Ya Zhang, and Masashi Sugiyama. Masking: A new perspective of noisy supervision. *arXiv preprint arXiv:1805.08193*, 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. *arXiv preprint arXiv:1804.06872*, 2018b.

Bo Han, Gang Niu, Xingrui Yu, Quanming Yao, Miao Xu, Ivor Tsang, and Masashi Sugiyama. Sigua: Forgetting may make learning with noisy labels more robust. In *International Conference on Machine Learning*, pages 4006–4016. PMLR, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

Bin Huang, Ying Xie, and Chaoyang Xu. Learning with noisy labels via clean aware sharpness aware minimization. *Scientific Reports*, 15(1):1350, 2025.

Zhuo Huang, Li Shen, Jun Yu, Bo Han, and Tongliang Liu. Flatmatch: Bridging labeled data and unlabeled data with cross-sharpness for semi-supervised learning. *Advances in neural information processing systems*, 36:18474–18494, 2023.

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 *International Conference on Machine Learning*, pages 2304–2313. PMLR, 2018.

Daehwan Kim, Kwangrok Ryoo, Hansang Cho, and Seungryong Kim. Splitnet: learnable clean-noisy label splitting for learning with noisy labels. *International Journal of Computer Vision*, 133(2):549–566, 2025.

Jan Kremer, Fei Sha, and Christian Igel. Robust active label correction. In *International conference on artificial intelligence and statistics*, pages 308–316. PMLR, 2018.

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

Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *ICLR*, 2019.

Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

Xiu-Chuan Li, Xiaobo Xia, Fei Zhu, Tongliang Liu, Xu-Yao Zhang, and Cheng-Lin Liu. Dynamics-aware loss for learning with label noise. *Pattern Recognition*, 144:109835, 2023.

Xuefeng Li, Tongliang Liu, Bo Han, Gang Niu, and Masashi Sugiyama. Provably end-to-end label-noise learning without anchor points. *arXiv preprint arXiv:2102.02400*, 2021.

Yexiong Lin, Yu Yao, Xiaolong Shi, Mingming Gong, Xu Shen, Dong Xu, and Tongliang Liu. Cs-isolate: Extracting hard confident examples by content and style isolation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Yexiong Lin, Yu Yao, and Tongliang Liu. Learning the latent causal structure for modeling label noise. *Advances in Neural Information Processing Systems*, 37:120549–120577, 2024.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33:20331–20342, 2020.

Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015.

Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, pages 3355–3364. PMLR, 2018.

Eran Malach and Shai Shalev-Shwartz. Decoupling" when to update" from" how to update". *arXiv preprint arXiv:1706.02613*, 2017.

Aditya Menon, Brendan Van Rooyen, Cheng Soon Ong, and Bob Williamson. Learning from corrupted binary labels via class-probability estimation. In *International conference on machine learning*, pages 125–134. PMLR, 2015.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *Advances in neural information processing systems*, 26, 2013.

Tri Nguyen, Shahana Ibrahim, and Xiao Fu. Noisy label learning with instance-dependent outliers: Identifiability via crowd wisdom. *Advances in Neural Information Processing Systems*, 37:97261–97298, 2024.

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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1944–1952, 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 *International Conference on Machine Learning*, pages 4334–4343. PMLR, 2018.

Bernhard Schölkopf. Causality for machine learning. In *Probabilistic and causal inference: The works of Judea Pearl*, pages 765–804. 2022.

Mengmeng Sheng, Zeren Sun, Tao Chen, Shuchao Pang, Yucheng Wang, and Yazhou Yao. Foster adaptivity and balance in learning with noisy labels. In *European Conference on Computer Vision*, pages 217–235. Springer, 2024.

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. *Advances in neural information processing systems*, 32, 2019.

Bingrui Su, Liangxiao Jiang, and Shanshan Si. Confident learning-based noise correction for crowdsourcing. *Pattern Recognition*, 169:111962, 2026.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5552–5560, 2018.

Arash Vahdat. Toward robustness against label noise in training deep discriminative neural networks. *arXiv preprint arXiv:1706.00038*, 2017.

Brendan Van Rooyen and Robert C Williamson. A theory of learning with corrupted labels. *Journal of Machine Learning Research*, 18(228):1–50, 2018.

Brendan Van Rooyen, Aditya Menon, and Robert C Williamson. Learning with symmetric label noise: The importance of being unhinged. *Advances in neural information processing systems*, 28, 2015.

Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 839–847, 2017.

Sudheendra Vijayanarasimhan and Kristen Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. *International journal of computer vision*, 108:97–114, 2014.

Haoyu Wang, Zhuo Huang, Zhiwei Lin, and Tongliang Liu. Noisegpt: Label noise detection and rectification through probability curvature. *Advances in Neural Information Processing Systems*, 37:120159–120183, 2024a.

Jialiang Wang, Xiong Zhou, Deming Zhai, Junjun Jiang, Xiangyang Ji, and Xianming Liu. *epsilon*-softmax: Approximating one-hot vectors for mitigating label noise. *Advances in Neural Information Processing Systems*, 37:32012–32038, 2024b.

Ke Wang, Guillermo Ortiz-Jimenez, Rodolphe Jenatton, Mark Collier, Efi Kokiopoulou, and Pascal Frossard. Pi-dual: Using privileged information to distinguish clean from noisy labels. In *Forty-first International Conference on Machine Learning*, 2024c.

Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019.

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

Tong Wei, Jiang-Xin Shi, Min-Ling Zhang, and Yu-Feng Li. Robust long-tailed learning under label noise. *Frontiers of Computer Science*, 20(1):1–12, 2026.

Peter Welinder and Pietro Perona. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 25–32. IEEE, 2010.

Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? *Advances in Neural Information Processing Systems*, 32:6838–6849, 2019.

Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems*, 33, 2020.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699, 2015.

Gezheng Xu, Li Yi, Pengcheng Xu, Jiaqi Li, Ruizhi Pu, Changjian Shui, Charles Ling, A Ian McLeod, and Boyu Wang. Unraveling the mysteries of label noise in source-free domain adaptation: Theory and practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Shuo Yang, Erkun Yang, Bo Han, Yang Liu, Min Xu, Gang Niu, and Tongliang Liu. Estimating instance-dependent bayes-label transition matrix using a deep neural network. In *International Conference on Machine Learning*, pages 25302–25312. PMLR, 2022.

Quanming Yao, Hansi Yang, Bo Han, Gang Niu, and James Tin-Yau Kwok. Searching to exploit memorization effect in learning with noisy labels. In *International Conference on Machine Learning*, pages 10789–10798. PMLR, 2020a.

Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual t: Reducing estimation error for transition matrix in label-noise learning. *arXiv preprint arXiv:2006.07805*, 2020b.

Yu Yao, Tongliang Liu, Mingming Gong, Bo Han, Gang Niu, and Kun Zhang. Instance-dependent label-noise learning under a structural causal model. *Advances in Neural Information Processing Systems*, 34:4409–4420, 2021.

Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, pages 7164–7173. PMLR, 2019.

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

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Jingfeng Zhang, Bo Song, Haohan Wang, Bo Han, Tongliang Liu, Lei Liu, and Masashi Sugiyama. Badlabel: A robust perspective on evaluating and enhancing label-noise learning. *IEEE transactions on pattern analysis and machine intelligence*, 46(6):4398–4409, 2024.

Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *32nd Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

Evgenii Zheltonozhskii, Chaim Baskin, Avi Mendelson, Alex M Bronstein, and Or Litany. Contrast to divide: Self-supervised pre-training for learning with noisy labels. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1657–1667, 2022.