CLID-MU: Cross-Layer Information Divergence Based Meta Update Strategy for Learning with Noisy Labels

Ruofan Hu
Worcester Polytechnic Institute
Worcester, MA, USA
rhu@wpi.edu

Dongyu Zhang
ByteDance
San Jose, CA, USA
dongyu.zhang@bytedance.com

Huayi Zhang
ByteDance
San Jose, CA, USA
huayi.zhang@bytedance.com

Elke Rundensteiner
Worcester Polytechnic Institute
Worcester, MA, USA
rundenst@wpi.edu

Abstract

Learning with noisy labels (LNL) is essential for training deep neural networks with imperfect data. Meta-learning approaches have achieved success by using a clean unbiased labeled set to train a robust model. However, this approach heavily depends on the availability of a clean labeled meta-dataset, which is difficult to obtain in practice. In this work, we thus tackle the challenge of meta-learning for noisy label scenarios without relying on a clean labeled dataset. Our approach leverages the data itself while bypassing the need for labels. Building on the insight that clean samples effectively preserve the consistency of related data structures across the last hidden and the final layer, whereas noisy samples disrupt this consistency, we design the Cross-layer Information Divergence-based Meta Update Strategy (CLID-MU). CLID-MU leverages the alignment of data structures across these diverse feature spaces to evaluate model performance and use this alignment to guide training. Experiments on benchmark datasets with varying amounts of labels under both synthetic and real-world noise demonstrate that CLID-MU outperforms state-of-the-art methods. The code is released at *https://github.com/ruofanhu/CLID-MU*.

CCS Concepts

• **Computing methodologies → Neural networks**; **Supervised learning**; **Learning from implicit feedback**; *Semi-supervised learning settings*.

Keywords

Noisy Labels, Neural Network, Meta-learning

ACM Reference Format:
Ruofan Hu, Dongyu Zhang, Huayi Zhang, and Elke Rundensteiner. 2025. CLID-MU: Cross-Layer Information Divergence Based Meta Update Strategy for Learning with Noisy Labels. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3736880

KDD '25, Toronto, ON, Canada
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3736880

KDD Availability Link:
The source code of this paper has been made publicly available at https://doi.org/10.5281/zenodo.15595961.

1 Introduction

Background. Developing deep neural networks (DNN) requires a large amount of labeled data. Yet due to the high cost and difficulty of data labeling, curating large datasets with high-quality labels is challenging. Some real-world datasets are curated via web crawling or crowdsourcing, inevitably yielding noisy labels [27]. Recent advances in learning with noisy labels (LNL) have shown success in training robust models under such conditions [27]. Among them, meta-learning approaches [33, 54] are particularly effective by leveraging *a small clean unbiased (balanced) labeled set* as a meta-dataset to evaluate model performance during training and effectively guide the training process.

State-of-the-Art and its limitations. However, acquiring a high-quality, unbiased labeled dataset is often infeasible in real-world scenarios due to the substantial time, cost, and effort required. For example, the popular CIFAR-100 dataset consists of images categorized into 100 classes, including various animal species and vehicle types. Selecting an equal number of images per class and accurately assigning labels would be highly time-consuming and labor-intensive. Moreover, meta-learning-based methods have been shown to perform poorly when using a noisy labeled meta-dataset instead of a clean one [61], posing a major obstacle in real-world applications where label noise is unavoidable. Existing studies have explored two main strategies: (1) using a noisy labeled set with a robust loss function, such as MAE [5, 45], as meta-loss to mitigate the impact of noisy labels, and (2) progressively selecting a pseudo-clean subset as the meta-dataset during training utilizing the small-loss trick [28, 42, 57]. However, the effectiveness of these methods is significantly hindered by the noisy patterns of the datasets. Robust loss functions struggle to handle complex noise patterns, such as instance-dependent noise [31]. While pseudo-clean subset selection methods often require careful threshold tuning, the small-loss trick fails to reliably distinguish between clean and noisy labeled samples under instance-dependent noise. Consequently, the evaluation of model performance based on robust loss functions or pseudo-clean subsets becomes unreliable. The guidance derived from such unreliable evaluation approaches can mislead the training process, increasing the risk of overfitting to noisy labels.

Challenges. The primary challenge arises from the noisy labels. In the absence of clean labels, it becomes impractical to establish well-founded criteria for selecting pseudo-clean subsets. Prior works typically rely on conventional supervised loss functions as the meta-objective to update the meta-model. However, the meta-update process is susceptible to the potential noisy labels in the meta-dataset, which causes bias propagation to the meta-model. Consequently, a solution that operates independently of noisy labels must be developed.

Proposed method. In this paper, we first propose an unsupervised metric called Cross-Layer Information Divergence (CLID), which operates independently of labels. CLID leverages the insight that clean samples maintain alignment between the data distribution in the penultimate latent space and the output layer, whereas noisy samples tend to disrupt this alignment. Our CLID metric measures the divergence of the data distribution at the last hidden layer and the final layer of the model. We demonstrate that CLID indeed closely correlates with the model performance. Building on this, we introduce a novel CLID-based meta-update strategy, termed CLID-MU, that addresses the above challenge of meta-learning with noisy labels in the absence of clean labeled data. CLID-MU exploits the alignment of data structures across diverse feature spaces and is designed to function independently of label quality, ensuring robust model performance and producing more compact features. The core idea is to dynamically measure the CLID of the model for each training batch, providing informative guidance to the model training process. Specifically, the meta-model is updated using the meta-gradients derived from CLID calculations on the data itself (independent of the labels). The meta-model, in turn, offers valuable signals to enhance the performance of the classification model.

Contributions. Our contributions include the following:

- We propose Cross-Layer Information Divergence (CLID), a novel unsupervised evaluation metric designed for scenarios lacking clean labeled data.
- We introduce CLID-MU, a CLID-guided meta-update strategy for meta-learning with noisy labels, without requiring clean validation data.
- Extensive experiments on benchmark datasets with synthetic and real-world noise show that CLID-MU consistently outperforms state-of-the-art methods.

2 Related Work

Numerous methods have been proposed to train robust deep networks with noisy labels. Easy-to-plug-in solutions like robust-loss functions, MAE [4], GCE [59], and APL [20], aim to resist label noise, but they still overfit when noise levels are high or complex. Similarly, regularization terms [18, 44, 50] are added to the loss function to reduce overfitting implicitly. Loss correction methods adjust sample loss based on noise transition matrices during training [39, 43, 45], while other strategies reduce weights for noisy samples [8, 12, 23]. Hybrid methods like CoLafier [49], DISC [17], and UNITY [9] incorporate both clean sample selection and label correction. However, these methods involve complex training procedures, requiring the coordination of multiple models and the careful tuning of dataset-specific hyperparameters, which makes them difficult to apply in practice.

Meta-learning [10, 23, 24, 37, 51, 53] is a general approach for learning with imperfect data. These methods optimize various configurations by using a clean validation set to evaluate the model, such as the weight for each training sample [23], the label transition matrix [43], the explicit weighting function [24, 28] for example re-weighting, the teacher model parameter [29] for label correction. Due to limited resources, constructing a clean and balanced validation set using expert knowledge is often impractical. To eliminate the need for a clean validation set, recent approaches employ robust loss functions on noisy labels [5, 45] or utilize heuristic approaches to select presumably clean samples as a validation set [28, 57]. Despite their promise, these methods encounter a performance ceiling when handling complex noise patterns, primarily due to their reliance on the quality of the labels. This dependency can result in overfitting to noise and hinder generalization. This highlights the need for more robust meta-learning approaches that can effectively deal with this challenging yet realistic problem.

Model selection without a clean validation set is a known challenge in weakly supervised settings like semi-supervised learning (SSL) and partial-label learning (PLL). Recent methods attempt to address this using validation-free strategies. For example, SLAM [15] and QLDS [3] estimate generalization errors in SSL. PLENCH [34] benchmarks PLL methods and proposes new selection criteria. However, these methods, being non-differentiable, are not suitable for gradient-based meta-learning.

3 Preliminaries

3.1 Problem Formulation

Let $D = \{x_i\}_{i=1}^N$ denote an unlabeled dataset drawn from a distribution $(x_i, y_i) \sim \mathcal{X}\times\mathcal{Y}$, where $y_i \in \{0,1\}^c$ is the one-hot ground truth label of x_i over c classes. With weak labelers such as crowdsourced workers, D is converted to a *noisy training set* $\tilde{D} = \{(x_i, \tilde{y}_i)\}_{i=1}^N$. $\tilde{D}$ may contain *inaccurate labels*, where $\tilde{y}_i \neq y_i$. We assume no clean labeled subset (*i.e.*, validation set) is available in $\tilde{D}$.

Given a noisy labeled dataset $\tilde{D}$, our goal is to develop a classification model that, without access to clean labeled data, can correctly predict the labels of unseen test data. The classification model f_θ is formulated as $f_\theta(x) = f_{\theta_2}^{cls} \circ f_{\theta_1}^{ext}(x)$ on instance x, where $f_{\theta_1}^{ext}$ is a feature extractor and $f_{\theta_2}^{cls}$ is a classifier. Let $\boldsymbol{z} = f_{\theta_1}^{ext}(x)$ denote the feature embedding of x and $\boldsymbol{q} = f_{\theta_2}^{cls}(\boldsymbol{z})$ the class probability, with $\boldsymbol{z}$ and $\boldsymbol{q}$ residing in the output space of the last hidden layer and final (output) layer, respectively.

3.2 Meta-learning Procedure

Here, we introduce the preliminaries on meta-learning upon which our proposed method rests [5, 23, 24, 45]. In meta-learning for noisy labels, there is a noisy training set $\tilde{D} = \{(x_i, \tilde{y}_i)\}_{i=1}^N$, and a separate meta-dataset $D^{meta} = \{(x_j^{meta}, y_j^{meta})\}_{j=1}^M$, where $M \ll N$ and y_j^{meta} could be inaccurate [5, 30]. Below, we explain the main strategy of reweighting training samples using an example based on the procedure outlined in WNet [24].

Namely, a meta-model $\Omega(\cdot;\psi)$ is coupled with the classification model $f(\cdot;\theta)$ to learn a weight for each training example. The meta-model, instantiated as a multilayer perceptron network (MLP), takes the training loss as input and maps it to a weight for the training sample. This mapping allows the meta-model to dynamically adjust the importance of each training sample during the training process. The parameter θ^* is optimized by minimizing the weighted loss:

$$\theta^*(\psi) = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \Omega(L_i(\theta); \psi) \cdot L_i(\theta), \tag{1}$$

where $L_i(\theta) = l(f(x_i; \theta); \tilde{y}_i)$ denotes cross-entropy loss for the i-th training sample and $\Omega(L_i(\theta); \psi)$ represents the corresponding generated weight for that sample.

Meta objective. Eq.(2) denotes the meta loss, where l^{meta} could be the cross-entropy loss (CE) or mean absolute error (MAE). The optimal parameter ψ^* for the meta-model can be obtained by minimizing the meta loss defined in Eq.(3).

$$L_j^{meta}(\theta^*(\psi)) = l^{meta}(f(x_j^{meta}; \theta^*(\psi)); y_j^{meta}). \tag{2}$$

$$\psi^* = \arg\min_{\psi} \frac{1}{M} \sum_{i=j}^{M} L_j^{meta}(\theta^*(\psi)). \tag{3}$$

Bi-level optimization. To solve both Eq.(1) and Eq.(3), an online updating strategy is widely used in the meta-learning literature [55] to update ψ and θ iteratively. Consider the t-th iteration, three steps are involved: *Virtual-Train, Meta-Train*, and *Actual-Train.* First, a batch of labeled samples $\{(x_i, \tilde{y}_i)\}_{i=1}^{n}$ and meta-dataset $\{(x_j^{meta}, y_j^{meta})\}_{j=1}^{m}$ are sampled, n and m represent the batch sizes, respectively. We may approximate θ^* and ψ^* with one gradient descent step updated value via a first-order Taylor expansion of the loss function. In the *Virtual-Train* step, the update of classification model's parameter θ is formulated as:

$$\hat{\theta}^{t+1}(\psi) = \theta^t - \alpha \frac{1}{n} \sum_{i=1}^{n} \Omega(L_i; \psi^t) \nabla_\theta L_i(\theta) \mid_{\theta^t}, \tag{4}$$

where α is the learning rate for the classification model. Then *Meta-Train* updates the meta-model by:

$$\psi^{t+1} = \psi^t - \gamma \frac{1}{m} \sum_{j=1}^{m} \nabla_\psi L_j^{meta}(\hat{\theta}^{t+1}(\psi)) \mid_{\psi^t}, \tag{5}$$

where γ is the learning rate for the meta-model. In the *Actual-Train* step, the classification model is finally updated using the updated meta-model by:

$$\theta^{t+1} = \theta^t - \alpha \frac{1}{n} \sum_{i=1}^{n} \Omega(L_i; \psi^{t+1}) \nabla_\theta L_i(\theta) \mid_{\theta^t}, \tag{6}$$

4 Our Proposed Method: CLID-MU

In this section, we present our proposed method CLID-MU. Our method builds on the *cluster assumption*, namely, samples forming a structure are more likely to belong to the same class [60]. We postulate that clean training samples align the data's structure in the feature space with that in the label space. To assess this alignment, we propose an unsupervised metric, CLID, which measures the divergence between the data distributions in the feature space and the label space. We then demonstrate how our proposed CLID metric correlates with the classification performance of DNNs. This important insight allows us to utilize our proposed differentiable metric effectively for non-supervised meta-learning.

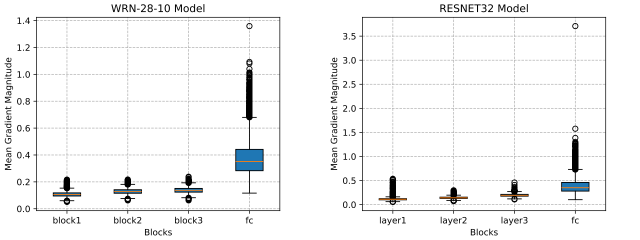


Figure 1: Illustration of gradient magnitudes in WRN-28-10 and ResNet32 residual layer blocks.

4.1 Cross-Layer Information Divergence (CLID)

Given a batch of data $\{x_j\}_{j=1}^{m}$ and a classification model f_θ, we create the feature embedding $\{z_j\}_{j=1}^{m}$ and class probability $\{\boldsymbol{q}_j\}_{j=1}^{m}$, which are data representations generated from the last hidden layer and final layer of DNN.

We generate a *fully connected embedding graph* $\boldsymbol{G^e}$ to capture the similarity of samples in the latent space as:

$$G_{ij}^{e} = \exp(\cos(z_i, z_j)/\tau), \quad \forall i, j \in \{1, \ldots, m\}, \tag{7}$$

where τ is a hyperparameter for temperature scaling and the exponential function is used to emphasize strong similarities. We then build the *class probability graph* by constructing the similarity matrix $\boldsymbol{G^q}$ as:

$$G_{ij}^{q} = \cos(\boldsymbol{q}_i, \boldsymbol{q}_j) \quad \forall i, j \in \{1, \ldots, m\} \tag{8}$$

Recall that we aim to measure the divergence of the representations produced by different DNN layers, while each graph represents instead the similarities between the representations. To better model the global structure of the two graphs and represent a valid probability distribution, we normalize both $\boldsymbol{G^e}$ and $\boldsymbol{G^q}$ with $\hat{G}_{ij} := G_{ij} / \sum_j G_{ij}$.

Given that we have constructed two graphs, each representing a data distribution, we can now measure the cross-layer information divergence between the two normalized graphs. Since the evolution speed (gradient magnitude) of each layer differs, layers with larger gradient magnitudes learn more information during each training step. Consequently, the data distribution generated by the slower-updating layer should gradually align with that of the faster-updating layer. We find that the gradient magnitude of the classifier layer is larger than that of the hidden layer, as shown in Figure 1. Therefore, we compute CLID using the cross-entropy between the two normalized graphs, as shown below:

$$L^{clid} = \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} -\hat{G}_{ij}^{q} \log \hat{G}_{ij}^{e} \tag{9}$$

4.2 CLID and Model Performance

The cross-entropy loss on the clean labeled test set is usually used to evaluate model performance. To establish a relationship between our novel unsupervised CLID metric and this standard supervised cross-entropy loss, we define two empirical alignment measures:

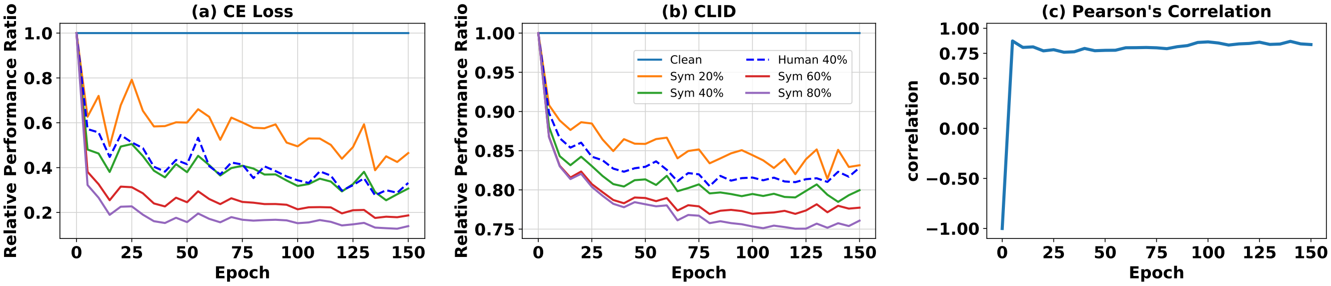


Figure 2: Demonstration on CIFAR-10: The relative performance ratio of (a) Cross-Entropy (CE) loss, (b) CLID, and (c) Pearson's correlation between CE loss and CLID across all data settings at each epoch.

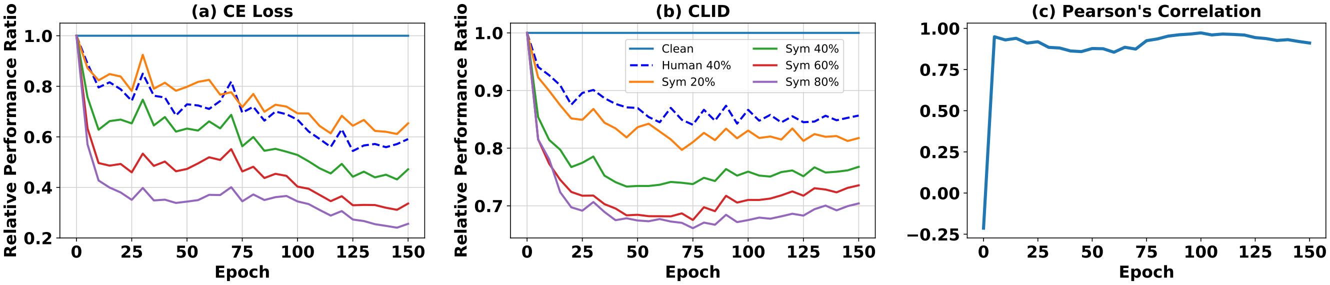


Figure 3: Demonstration on CIFAR-100: The relative performance ratio of (a) Cross-Entropy (CE) loss, (b) CLID, and (c) Pearson's correlation between CE loss and CLID across all data settings at each epoch.

the Relative Performance Ratio (RPR) and Performance Pearson's Correlation, as defined below.

Definition 1 (Relative Performance Ratio (RPR))**.** Let $\mathcal{D}$ denote a dataset, and let f_θ^n and f_θ^{clean} represent models trained on the dataset under noisy setting n and perfect clean labels, respectively. Denote the test performance of f_θ^n as P^n and the test performance of f_θ^{clean} as P^{clean}. The *Relative Performance Ratio (RPR)* of the model trained on setting n is defined as: $\text{RPR}(n) = \frac{P^{\text{clean}}}{P^n}$.

The RPR quantifies the relative performance degradation of a model trained under a noisy or altered setting compared to the ideal clean label scenario.

Definition 2 (Performance Pearson's Correlation)**.** Let $\mathcal{D}$ be a dataset, and $\mathcal{N}$ be a set of noisy settings applied to $\mathcal{D}$. For a deep neural network f_θ trained on $(\mathcal{D}, \mathcal{N})$ under a fixed training protocol, let $A_t(\mathcal{N})$ and $B_t(\mathcal{N})$ denote two evaluation metrics measured at epoch t for each noisy setting $n \in \mathcal{N}$. We define the correlation between A and B at training epoch t as the Pearson correlation coefficient:

$$r(A_t(\mathcal{N}), B_t(\mathcal{N})) = \frac{\sum_n (A_t(n) - \overline{A_t})(B_t(n) - \overline{B_t})}{\sqrt{\sum_n (A_t(n) - \overline{A_t})^2}\sqrt{\sum_n (B_t(n) - \overline{B_t})^2}},$$

where $\overline{A_t}$ and $\overline{B_t}$ are the mean values of $A_t(\mathcal{N})$ and $B_t(\mathcal{N})$ across all $n \in \mathcal{N}$, respectively. For all training epoch $t \in \{1, 2, \ldots, T\}$, we say that A and B exhibit a strong correlation if $r(A_t(\mathcal{N}), B_t(\mathcal{N})) \geq \rho$, for some threshold $\rho \in [0, 1]$, where ρ represents a strong positive correlation (e.g., $\rho > 0.7$).

We empirically demonstrate that CLID correlates with model performance on real-world datasets. Specifically, we explore a Resnet-34 [7] model on the CIFAR-10 and CIFAR-100 [13] datasets using 50,000 labeled samples and then evaluate the model using 10,000 test samples. The model's performance is assessed under various noise conditions, including noisy labels generated with symmetric noise ratios of {0, 20%, 40%, 60%, 80%}, where the correct label is randomly replaced with one of the other classes. Additionally, we consider noisy labels introduced by human annotators, with 40.21% and 40.20% of labeled samples affected (CIFAR-10N Worst and CIFAR-100N Fine [36]).

We compute the Relative Performance Ratio (RPR) of CE loss and CLID on the test set. Notably, the computation of CLID does not rely on clean labels. Thus, its value remains the same whether evaluated on clean or noisy data. The RPR of CE loss and CLID, as shown in Figures 2(a)(b) and Figures 3(a)(b), exhibit similar trends. This indicates that CLID effectively captures the performance degradation of models trained under various noisy settings, offering insights from the perspective of data structure alignment across different feature spaces. Further, the Performance Pearson's Correlations depicted in Figures 2(c) and 3(c) demonstrate a strong correlation between model performances measured by CLID and CE loss throughout

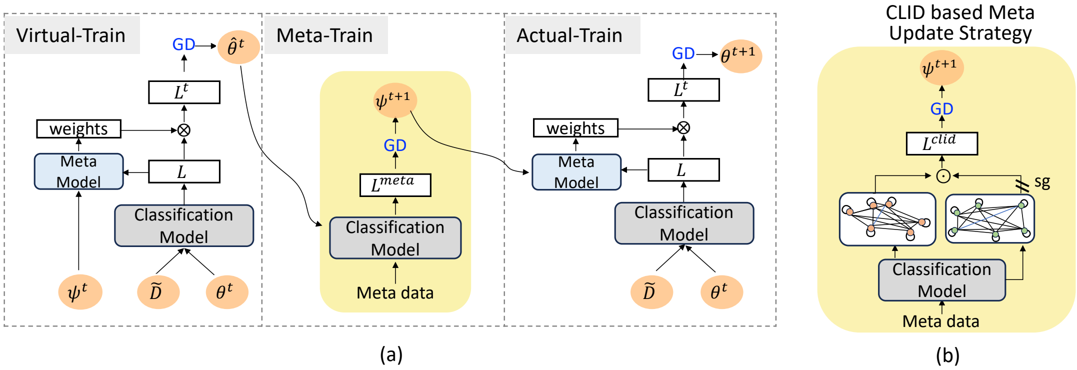


Figure 4: (a) Illustration of the three major steps of a meta-learning method, using a reweight-based approach as an example. (b) We propose a new meta loss, CLID, for the Meta-Train step. Given an unlabeled dataset, a class probability graph and an embedding graph are constructed to measure the similarity between samples in their respective spaces. CLID measures the divergence of the data distribution of the two graphs. The sg denotes stop-gradient.

the training process. This consistency underscores the agreement between the two metrics in evaluating model performance. Therefore, we can conclude that CLID is a robust and effective metric for assessing model performance.

4.3 CLID-MU: CLID-based Meta-Update Strategy

This connection between the CLID metric and model performance lays a foundation for using CLID to evaluate models when a guaranteed clean labeled set is not available. We propose to have CLID serve as meta loss in the *Meta-Train* step, as illustrated in Figure 4 The stop-gradient operation is designed primarily to prevent trivial constant solutions. Substituting L^{clid} into Eq. 5, the *meta-update step* becomes:

$$\psi^{t+1} = \psi^t + \frac{\alpha\gamma}{n} \sum_{i=1}^{n} g_i \frac{\partial \Omega(L_i; \psi^t)}{\partial \psi}, \tag{10}$$

where $g_i = \frac{\partial L_i(\theta)}{\partial \theta} \mid_{\theta^t}^T \frac{\partial L^{clid}(X, \hat{\theta}^{t+1}(\psi))}{\partial \theta} \mid_{\hat{\theta}^{t+1}}$, $L_i(\theta)$ denotes cross-entropy loss of x_i. g_i represents the similarity between the gradient of the loss for the training sample x_i and the gradient of L^{clid} on the complete unlabeled batch. The meta-model is then updated accordingly. The overall optimization procedure can be found in Algorithm 1 (appendix). We note the potential risk of overfitting to noisy labels during the training process of CLID-MU. However, unlike prior meta-update strategies based on supervised loss, our CLID-MU has a reduced risk of overfitting.

Remark 1. *Prior approaches aim to minimize the supervised meta loss L^{meta} for $g_i = \frac{\partial L_i(\theta)}{\partial \theta} \mid_{\theta^t}^T \frac{\partial L^{meta}(X, \tilde{Y}, \hat{\theta}^{t+1}(\psi))}{\partial \theta} \mid_{\hat{\theta}^{t+1}}$. That is, the noisy labels $\tilde{Y}$ are involved in the meta-model updating step. This can lead to overfitting to these noisy labels. In contrast, our CLID-MU does not rely on noisy labels in the meta-update step. As training progresses, the meta-model provides guidance to the classification model, which in turn enhances the meta-model, creating a* ***virtuous cycle****.*

Computational complexity analysis. Given a batch of validation data of size m, sample-wise supervised evaluation metrics, such as Cross-Entropy (CE) and Mean Absolute Error (MAE), have a computational complexity of $O(m)$. In contrast, CLID, being a pairwise metric, has a computational complexity of $O(m^2)$, making it m times more computationally intensive than supervised metrics. However, we argue that this additional computational cost is practically justified, as CLID-MU obviates the need for a clean validation set and substantially reduces the extensive hyperparameter tuning required by alternative methods (e.g., determining thresholds for selecting pseudo-clean sets using the small-loss criterion).

4.4 Snapshot Ensembling

With CLID as an effective evaluation metric for model evaluation, we propose leveraging snapshot ensembling [11] for inference. Specifically, the top K snapshots (i.e., model weights) are selected based on their CLID scores under CLID-MU, evaluated on the entire meta-dataset, and subsequently saved. During inference, the predictions from all saved snapshots are averaged to generate the final output. Let y_i be the one-hot ground truth label of x_i, and $F_{y_i}(x_i)$ be the predicted probability that x_i belongs to y_i. The final prediction, $F(x_i)$, is the average of the K saved model snapshots $\{f^k(x_i)\}_{k=1}^K$. The exponential loss $L^{exp} = \frac{1}{n} \sum_i^n \exp(-F_{y_i}(x_i))$ is often used to measure the error of the model [32]. We analyze the convergence of the snapshot ensembling by presenting the upper bound of the exponential loss.

Theorem 1. *The exponential loss L^{exp} is bounded by*

$$L^{exp} \leq \Pi_{k=1}^K R_k^{1/K},$$

where $R_k = \frac{1}{n} \sum_{i=1}^n exp(-f_{y_i}^k(x_i))$. The upper bound of L^{exp} decreases as K increases.

Theorem 1 demonstrates an exponential decrease in the L^{exp} bound as we save more snapshots.

Table 1: Comparison across meta-learning methods.

(a) Test accuracy (mean and std dev over 3 data folds) on CIFAR-10 and CIFAR-100 using WNet variants under symmetric (Sym.), asymmetric (Asy.), and instance-dependent noise (IDN) conditions. Only WNet-CE uses a clean meta-dataset. Bolded values indicate the highest and those within one standard deviation of the highest in each column.

Method		CIFAR-10					CIFAR-100				
		Sym. 20%	Sym. 40%	Sym. 60%	Asy. 40%	IDN 40%	Sym. 20%	Sym. 40%	Sym. 60%	Asy. 40%	IDN 40%
WNet-CE	best	**94.00**$_{\pm.24}$	91.77$_{\pm.14}$	86.68$_{\pm.06}$	91.54$_{\pm.02}$	91.19$_{\pm.52}$	77.42$_{\pm.07}$	73.11$_{\pm.07}$	64.87$_{\pm.32}$	62.49$_{\pm.31}$	71.24$_{\pm.52}$
	ens	**94.44**$_{\pm.21}$	**92.17**$_{\pm.25}$	87.11$_{\pm.18}$	91.98$_{\pm.03}$	91.40$_{\pm.62}$	**78.46**$_{\pm.06}$	**74.00**$_{\pm.24}$	**66.00**$_{\pm.50}$	64.73$_{\pm.07}$	**72.87**$_{\pm.66}$
WNet-CE(n)	best	94.92$_{\pm.07}$	92.61$_{\pm.27}$	88.38$_{\pm.08}$	92.08$_{\pm.16}$	90.53$_{\pm.75}$	75.82$_{\pm.73}$	70.30$_{\pm1.1}$	63.06$_{\pm.73}$	61.85$_{\pm.63}$	68.16$_{\pm.84}$
	ens	67.10$_{\pm.91}$	56.47$_{\pm.06}$	46.09$_{\pm1.09}$	51.72$_{\pm2.4}$	53.32$_{\pm.60}$	26.50$_{\pm1.81}$	14.54$_{\pm1.46}$	9.39$_{\pm0.59}$	16.09$_{\pm1.32}$	16.69$_{\pm.94}$
WNet-CE(p)	best	94.02$_{\pm.03}$	91.70$_{\pm.21}$	86.80$_{\pm.33}$	91.90$_{\pm.04}$	91.30$_{\pm.78}$	76.98$_{\pm.22}$	72.32$_{\pm.18}$	63.60$_{\pm71}$	62.70$_{\pm.17}$	70.30$_{\pm.23}$
	ens	**94.42**$_{\pm.01}$	**92.22**$_{\pm.44}$	**87.52**$_{\pm.11}$	**92.02**$_{\pm.07}$	91.54$_{\pm.53}$	78.05$_{\pm.07}$	73.57$_{\pm.04}$	64.34$_{\pm.42}$	64.12$_{\pm.52}$	71.54$_{\pm.24}$
WNet-MAE	best	94.10$_{\pm.17}$	91.66$_{\pm.35}$	86.88$_{\pm.13}$	91.65$_{\pm.18}$	91.42$_{\pm.46}$	77.16$_{\pm.16}$	72.63$_{\pm.08}$	63.51$_{\pm.31}$	63.74$_{\pm.10}$	70.90$_{\pm.08}$
	ens	**94.31**$_{\pm.07}$	**92.09**$_{\pm.42}$	87.29$_{\pm.08}$	**92.12**$_{\pm.07}$	91.76$_{\pm.35}$	**78.20**$_{\pm.26}$	73.72$_{\pm.04}$	64.41$_{\pm.34}$	65.08$_{\pm.28}$	**72.72**$_{\pm.66}$
WNet-CLID	best	94.18$_{\pm.23}$	91.66$_{\pm.01}$	86.88$_{\pm.01}$	91.97$_{\pm.04}$	91.83$_{\pm.21}$	77.36$_{\pm.02}$	73.12$_{\pm.38}$	64.82$_{\pm.43}$	65.78$_{\pm.1}$	71.50$_{\pm.40}$
	ens	**94.34**$_{\pm.06}$	**92.22**$_{\pm.03}$	**87.27**$_{\pm.17}$	91.93$_{\pm.06}$	**92.28**$_{\pm.11}$	**78.40**$_{\pm.01}$	**74.32**$_{\pm.51}$	**65.98**$_{\pm.54}$	**67.66**$_{\pm.37}$	**73.10**$_{\pm.32}$

(b) Test accuracy (mean and std dev over 3 data folds) on CIFAR-10 and CIFAR-100 using VRI variants under symmetric (Sym.), asymmetric (Asy.), and instance-dependent noise (IDN) conditions. Only VRI-CE uses a clean meta-dataset. Bolded values indicate the highest and those within one standard deviation of the highest in each column.

Method		CIFAR-10					CIFAR-100				
		Sym. 20%	Sym. 40%	Sym. 60%	Asy. 40%	IDN 40%	Sym 20%	Sym 40%	Sym 60%	Asy. 40%	IDN 40%
VRI-CE	best	**93.32**$_{\pm.02}$	91.15$_{\pm.05}$	87.44$_{\pm.24}$	91.43$_{\pm.16}$	90.08$_{\pm.30}$	71.21$_{\pm.03}$	65.67$_{\pm.47}$	57.60$_{\pm.13}$	**63.28**$_{\pm.24}$	62.13$_{\pm.01}$
	ens	**93.46**$_{\pm.59}$	**91.92**$_{\pm.12}$	**88.27**$_{\pm.11}$	89.94$_{\pm.67}$	88.80$_{\pm.88}$	71.42$_{\pm.22}$	66.22$_{\pm.63}$	58.75$_{\pm.32}$	60.64$_{\pm.06}$	65.88$_{\pm.29}$
VRI-CE(n)	best	93.49$_{\pm.13}$	91.44$_{\pm.03}$	87.94$_{\pm.12}$	90.38$_{\pm.72}$	88.71$_{\pm.44}$	72.08$_{\pm.34}$	65.69$_{\pm.25}$	57.15$_{\pm.28}$	56.88$_{\pm.59}$	62.96$_{\pm.47}$
	ens	70.86$_{\pm.65}$	60.14$_{\pm1.18}$	52.01$_{\pm.60}$	57.97$_{\pm4.71}$	60.66$_{\pm2.76}$	33.50$_{\pm1.06}$	23.84$_{\pm1.01}$	15.52$_{\pm.72}$	24.39$_{\pm.18}$	24.94$_{\pm.74}$
VRI-CE(p)	best	93.58$_{\pm.21}$	91.38$_{\pm.08}$	87.52$_{\pm.21}$	91.77$_{\pm.85}$	89.43$_{\pm.22}$	71.87$_{\pm.28}$	65.97$_{\pm.23}$	58.06$_{\pm.23}$	55.92$_{\pm.11}$	62.70$_{\pm.02}$
	ens	**93.95**$_{\pm.35}$	90.96$_{\pm.18}$	**88.39**$_{\pm.13}$	89.77$_{\pm.20}$	87.52$_{\pm.94}$	72.45$_{\pm.01}$	64.03$_{\pm.10}$	55.47$_{\pm2.23}$	54.82$_{\pm.16}$	58.88$_{\pm.81}$
VRI-MAE	best	93.42$_{\pm.02}$	91.54$_{\pm.08}$	87.56$_{\pm.12}$	91.17$_{\pm.27}$	88.91$_{\pm.38}$	71.45$_{\pm.34}$	65.36$_{\pm.25}$	57.18$_{\pm1.05}$	55.36$_{\pm1.0}$	62.62$_{\pm.26}$
	ens	**93.82**$_{\pm.02}$	**92.22**$_{\pm.06}$	**88.39**$_{\pm.23}$	**91.87**$_{\pm.45}$	89.56$_{\pm.08}$	**73.19**$_{\pm.31}$	67.30$_{\pm.11}$	58.03$_{\pm1.34}$	56.70$_{\pm1.56}$	62.98$_{\pm.79}$
VRI-CLID	best	93.10$_{\pm.21}$	90.96$_{\pm.06}$	86.34$_{\pm.19}$	91.59$_{\pm.15}$	90.98$_{\pm.25}$	71.85$_{\pm.04}$	67.13$_{\pm.40}$	58.85$_{\pm.17}$	62.42$_{\pm1.36}$	66.48$_{\pm.14}$
	ens	**93.35**$_{\pm.41}$	90.50$_{\pm.04}$	86.58$_{\pm.31}$	**92.10**$_{\pm.47}$	**91.10**$_{\pm.01}$	**73.40**$_{\pm.16}$	**68.99**$_{\pm.11}$	**60.45**$_{\pm.36}$	**63.70**$_{\pm1.16}$	**68.20**$_{\pm.47}$

5 Experimental Study

With CLID-MU being a model-agnostic approach, we conduct experiments to validate its effectiveness on benchmark datasets across various learning methods. We focus on four research questions:

(1) **RQ1**: How effective is CLID-MU compared to alternative baselines across diverse meta-learning methods when a clean labeled dataset is unavailable?
(2) **RQ2**: How does CLID-MU perform compared to state-of-the-art LNL frameworks?
(3) **RQ3**: How robust is CLID-MU in real-world scenarios where only a small portion of the data is noisily labeled, while the majority remains unlabeled, i.e., in semi-supervised settings?
(4) **RQ4**: Is CLID-MU sensitive to the selection of its hyperparameters?

5.1 Comparison with Meta-learning Methods

This subsection demonstrates that, without requiring access to a clean labeled dataset, our method achieves superior performance across two meta-learning methods: WNet [24] and VRI [28].

Experimental setup. Experiments are run on the *CIFAR-10* and *CIFAR-100* [13] with three types of noise: symmetric, asymmetric, and instance-dependent noise. Symmetric noise uniformly flips labels to a random class with probability p. Asymmetric noise means the labels are flipped to similar classes with probability p. The instance-dependent noise is obtained by setting a random noise probability p for each instance following a truncated Gaussian distribution [38]. For all methods, the meta-dataset size is fixed at 1000 samples. CLID-MU uses a randomly sampled meta-dataset from the noisy training set, while baseline methods select it evenly across classes based on training labels. We report the *best* accuracy, defined as the highest accuracy achieved on the clean test set during training. For fair comparisons, we apply snapshot ensembling to all methods and report the *ensemble* accuracy, obtained using five model snapshots selected based on the meta-objective of each method (i.e., evaluation performance).

Baselines. We take the standard meta-learning with clean metadata using cross-entropy (CE) loss as the meta-objective to reference ceiling performance. We compare CLID-MU against three alternative baseline methods: 1) **CE(n)**: Noisy samples are randomly selected with class balance to form the meta-dataset, using CE as the meta-objective. 2) **CE(p)**: Following [28], we select reliable samples with smaller losses using Gaussian Mixture Model (GMM) clustering, ensuring an even selection across all classes. These samples are designated as the pseudo-clean meta-dataset with CE as the meta-objective. Following prior work, we perform an initial warming-up phase (10 epochs for CIFAR-10 and 30 epochs for CIFAR-100) before proceeding with sample selection. 3) **MAE**: Following [5], we set MAE as the meta-objective and apply it to a randomly selected meta-dataset drawn from the noisy training set. See the appendix for implementation details.

Results. Tables 1a and 1b present the results on CIFAR datasets for the meta-learning methods WNet and VRI, respectively. Our findings demonstrate that CLID-MU is effective when integrated into different meta-learning frameworks. For both methods, While using cross-entropy loss on a noisy validation set yields strong *best accuracy* under simple noise, it leads to substantial degradation in *ensemble accuracy* and overall performance in complex settings. When incorporating other meta-objectives into WNet, we observe that the performance of all methods remains relatively close on CIFAR-10 across different noise settings. WNet-CLID achieves superior performance in high-noise scenarios, including 60% symmetric noise, 40% asymmetric noise, and 40% instance-dependent noise. Impressively, WNet-CLID even outperforms WNet-CE, which leverages a clean meta-dataset, confirming its effectiveness in handling complex and challenging noise patterns.

In the VRI framework, VRI-CLID demonstrates competitive performance against other baselines under both symmetric and asymmetric noise on CIFAR-10 and excels under instance-dependent noise. On CIFAR-100, VRI-CLID excels across all noise settings, consistently outperforming every baseline. It even surpasses the performance ceiling of VRI-CE, which is trained on a clean meta-dataset. The *ensemble* accuracy generally exceeds the *best* accuracy in most scenarios, underscoring the benefits of snapshot ensembling during inference. However, in more challenging settings (asymmetric and instance-dependent noise) on CIFAR-10 and CIFAR-100, the ensemble accuracy of VRI-CE(p) falls below its best accuracy, indicating that the pseudo-clean set selected using the small-loss trick may be unreliable for model evaluation.

5.2 Comparison with State-of-the-art Methods

We compare our method with competitive methods on datasets with real-world noise, CIFAR-10N (Worst) and CIFAR-100N [36] in Table 2. The meta-dataset with 1000 samples is randomly sampled from the noisy training set. We compare with the competitive methods:(1) Co-teaching [6] and ELR+ [18] train two networks that mutually refine each other; (2) SOP [19] models label noise through over-parameterization and incorporates an implicit regularization term; (3) DivideMix [14] employs two networks, dynamically separating the training set into clean and noisy subsets using the small-loss trick and handling the noisy subset through a semi-supervised learning fashion. To compare with the state-of-the-art methods, we integrated VRI-CLID into DivideMix to demonstrate that our method is compatible with existing LNL methods and enhances performance on both datasets.

We also experiment on the Animal-10N [26] and Clothing1M [40] data sets, both of which contain naturally occurring noisy labels introduced by human error. As shown in Tables 3 and 4, we compare the performance of VRI-CLID and WNet-CLID with state-of-the-art methods. On Animal-10N, VRI-CLID and VRI-CE(p) achieve similar performance to each other, and outperform all competing methods except DISC, demonstrating the effectiveness of CLID-MU in handling real-world label noise. On Clothing1M, both WNet-CLID and VRI-CLID achieve performance comparable to other leading methods and surpass the baseline of using MAE loss as the meta-objective.

Table 2: Test Accuracy (mean and std dev over 3 runs) on CIFAR-10N Worst and CIFAR-100N. * denotes our implementation, other results are from [36]. "†" means the reported accuracy is from snapshot ensembling.

Method	CIFAR-10N(Worst)	CIFAR-100N
CE	77.69 ± 1.55	55.5 ± 0.66
Co-teaching	83.83 ± 0.13	60.37 ± 0.27
SOP	**93.24 ± 0.21**	67.81 ± 0.23
ELR+	91.09 ± 1.60	66.72 ± 0.07
Divide-Mix*	90.43 ± 0.57	67.04 ± 0.59
VRI-CLID†	89.07 ± 0.18	67.53 ± 0.34
VRI-CLID + Divide-Mix†	90.70 ± 0.11	**70.05 ± 0.20**

Table 3: Test Accuracy (mean and std dev over 3 runs) on Animal-10N. Results are directly from the original papers. "†" means the reported accuracy is from snapshot ensembling.

Method	Accuracy	Method	Accuracy
CE [2]	79.4 ± 0.14	GJS [2]	84.2 ± 0.07
GCE [58]	81.5 ± 0.08	DISC [17]	**87.1 ± 0.15**
SELIE [26]	81.8 ± 0.09	Nested Co-teaching [1]	84.1 ± 0.10
MixUp [52]	82.7 ± 0.03	VRI-CE(p)†	85.5 ± 0.51
PLC [56]	83.4 ± 0.43	VRI-CLID† (ours)	85.6 ± 0.57

Table 4: Test Accuracy on Clothing1M. Results are directly from the original papers. "†" means the reported accuracy is from snapshot ensembling.

Method	Accuracy	Method	Accuracy
CE [45]	68.94	Forward [22]	69.91
Co-teaching [6]	60.15	ELR [46]	72.87
Dual T [43]	71.49	WNet-MAE†	72.85
BARE [21]	72.28	WNet-CLID† (ours)	**72.93**
VolminNet [16]	72.42	VRI-MAE†	67.78
ROBOT (RCE) [45]	72.70	VRI-CLID† (ours)	72.83

5.3 Semi-supervised Real-world Scenarios

Experimental setup. We conducted experiments on CIFAR-10 with symmetric noise at {20%, 50%} and asymmetric noise at 40%, generated following the scheme in [22]. Experiments are also conducted on the real-world human-annotated dataset CIFAR-10N [36].

Table 5: Test accuracy (mean and std dev over 3 data folds) on CIFAR-10 with different noise types and noise ratios. Average noise ratios of human annotations over these three folds are in parentheses. Best results in bold, second highest underlined.

Methods	Symmetric		Asymmetric	Human		
	20%	50%	40%	Aggregate (8.8%)	Random1 (16.9%)	Worst (40.6%)
UDA	72.04 ± 0.18	49.97 ± 2.91	70.95 ± 0.27	79.97 ± 0.37	75.05 ± 0.40	63.39 ± 0.89
w/ELR	78.22 ± 0.95	63.61 ± 0.33	72.38 ± 0.41	80.96 ± 0.28	79.14 ± 0.06	64.86 ± 3.73
w/MixUp	77.00 ± 0.53	58.23 ± 1.49	73.27 ± 0.75	82.78 ± 0.31	79.52 ± 0.33	68.26 ± 0.12
w/WNet-MAE	85.25 ± 0.92	60.79 ± 16.55	72.92 ± 0.85	86.49 ± 0.84	82.90 ± 0.88	70.72 ± 2.03
w/WNet-CLID	**86.22± 1.90**	**78.73± 3.52**	**73.95± 0.32**	**88.17 ± 0.81**	**85.27 ± 1.65**	**75.09 ± 4.88**
FixMatch	73.36 ± 0.26	51.07 ± 1.10	71.76 ± 0.79	83.00 ± 0.37	78.08 ± 0.54	61.37 ± 0.40
w/ELR	74.17 ± 1.56	51.00 ± 1.34	72.80 ± 0.51	83.01 ± 0.48	81.07 ± 0.40	70.10 ± 4.52
w/MixUp	76.27 ± 0.17	58.48 ± 1.32	71.93 ± 0.91	83.02 ± 0.21	79.27 ± 0.66	67.66 ± 0.91
w/WNet-MAE	84.97 ± 3.02	53.43 ± 4.76	**72.85 ± 0.88**	86.85 ± 0.18	82.05 ± 0.56	63.32 ± 2.68
w/WNet-CLID	**88.87± 0.22**	**84.00± 2.34**	72.57± 0.58	**89.78 ± 0.19**	**88.20 ± 0.86**	**80.95 ± 2.52**
FlexMatch	78.49 ± 0.30	68.86 ± 1.15	76.06 ± 0.42	85.01 ± 0.23	81.44 ± 0.68	72.37 ± 0.86
w/ELR	81.46 ± 0.43	63.50 ± 2.08	75.78 ± 0.61	83.24 ± 0.36	81.74 ± 0.87	67.20 ± 0.60
w/MixUp	84.73 ± 0.21	77.31 ± 0.81	77.84 ± 0.56	87.68 ± 0.04	85.76 ± 0.60	78.20 ± 0.28
w/WNet-MAE	87.40 ± 1.23	76.95 ± 6.29	78.51 ± 0.25	88.52 ± 0.32	85.28 ± 0.63	76.79 ± 0.90
w/WNet-CLID	**89.29± 0.79**	**83.57 ± 1.99**	**78.80± 1.04**	**89.71 ± 0.48**	**88.27 ± 1.15**	**81.58 ± 2.78**

Each image in CIFAR-10N is associated with three kinds of labels: aggregation of three annotations by majority voting (Aggregate), random selection of one from all annotations (Random 1, 2, 3), and the worst annotation (Worst). The quality of these labels decreases in the mentioned order. We used Aggregate, Random1, and Worst in our experiments.

Baselines. We evaluated CLID-MU by integrating it into three widely-used SSL methods: UDA [41], FixMatch [25], and Flexmatch [48]. We compare it with the following methods, each also naturally integrated into these SSL frameworks: (1) implicit regularization methods proven to have strong performance in dealing with noisy labels, including ELR [18] and MixUp [52]; and (2) WNet-MAE [5], an explicit regularization method that can operate in scenarios without access to clean labeled data.

Results. Table 5 shows the results on CIFAR-10 with various noise types. It can be observed that WNet-CLID *outperforms the compared methods by a large margin across all three SSL methods*, particularly under high noise ratios. ELR and MixUp are less effective under challenging settings, such as symmetric noise at 50% and with (noisy) human labels. The performance of WNet-MAE degrades with higher noise ratios because it relies on MAE loss, which is heavily dependent on the quality of the labels. In contrast, our WNet-CLID succeeds in eliminating the influence of noisy labels when training the meta-model. The superior performance and robustness under real-world noise demonstrate that it has greater potential to be applied in practical SSL scenarios.

5.4 Sensitivity Analysis

Effect of temperature scaling. One critical hyperparameter in CLID is the temperature scaling factor τ. This parameter governs the sharpness of the similarity scores. Since feature embeddings are derived after a ReLU layer, their values are constrained to the range $[0, 1]$. When $\tau > 1$, the embedding graph becomes more uniform, whereas a smaller τ amplifies the similarity scores, resulting in a sharper distribution. We evaluate the robustness of CLID-MU across τ values within $\{0.1, 0.3, 0.5, 0.7, 1, 1.5\}$ using the meta-learning method VRI. The experiments are conducted on the CIFAR-100 dataset with 60% symmetric noise and 40% instance-dependent noise. Figure 5 shows that both the best accuracy and ensemble accuracy are relatively stable across different values of temperature scaling. The ensemble accuracy reaches its highest at $\tau = 0.5$ and $\tau = 0.3$ for 60% symmetric noise and $\tau = 0.1$ for 40% instance-dependent noise. Notably, the corresponding average CLID scores of the top-5 model snapshots are the lowest, suggesting that the hyperparameter τ can be tuned based on average CLID scores to optimize performance.

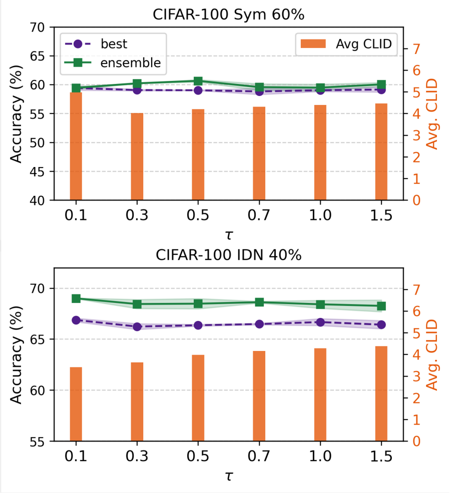


Figure 5: Test accuracy across different values of temperature scaling (τ) and the corresponding average CLID score of the top-5 model snapshots.

Effect of batch size for CLID. We investigated whether a larger batch size for the unsupervised CLID metric could improve training robustness. Using the CIFAR-10N dataset with 4000 labeled data

Table 6: Test accuracy with various Meta-Train batch sizes in semi-supervised learning (Flexmatch) on CIFAR-10N. Results are mean and std dev. across 3 data folds.

Batch size	Aggregate	Random1	Worst
50	89.85 ± 0.22	88.39 ± 1.12	80.28 ± 3.52
100	89.71 ± 0.48	88.27 ± 1.15	81.58 ± 2.78
300	89.83 ± 0.48	88.51 ± 1.01	81.60 ± 3.33
500	89.88 ± 0.44	88.46 ± 1.02	81.67 ± 2.19

points, we tested FlexMatch with WNet-CLID by varying the meta-train batch size from 50 to 500. As shown in Table 6, performance remained stable for batch sizes under low noise ratios (Aggregate and Random1). However, in more challenging settings (Worst), performance improved when the batch size was larger than 50. We observed no significant performance difference for batch sizes of 100 or greater.

6 Conclusion

In this paper, we propose Cross-Layer Information Divergence Based Meta Update Strategy (CLID-MU) for learning with noisy labels (LNL) without access to a clean labeled set. Unlike prior works that use supervised loss as meta-loss to evaluate model performance, CLID-MU effectively utilizes unlabeled data to measure the cross-layer information divergence (CLID) and then leverages CLID to evaluate the model performance during the Meta-Train step. We evaluate our CLID-MU method on benchmark datasets under synthetic and real-world noises across numerous data settings, including learning with noisy labels and semi-supervised learning with noisy labels. Our comprehensive experimental results demonstrate that our CLID-MU achieves superior performance compared to state-of-the-art methods. Further, CLID-MU is orthogonal to other LNL approaches, such as MixUp and label correction, and can be readily combined with them to enhance their performance. Future work involves exploring CLID for different layers beyond the label space and the feature space of the last encoder block.

Acknowledgments

This work was supported by USDA NIFA (AFRI Award No. 2020-67021-32459) and NSF (Grants IIS-1910880, CSSI-2103832, NRT-HDR-2021871).

References

[1] Yingyi Chen, Xi Shen, Shell Xu Hu, and Johan AK Suykens. 2021. Boosting co-teaching with compression regularization for label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2688–2692.

[2] Erik Englesson and Hossein Azizpour. 2021. Generalized jensen-shannon divergence loss for learning with noisy labels. *Advances in Neural Information Processing Systems* 34 (2021), 30284–30297.

[3] Vasilii Feofanov, Malik Tiomoko, and Aladin Virmaux. 2023. Random matrix analysis to balance between supervised and unsupervised learning under the low density separation assumption. In *International Conference on Machine Learning*. PMLR, 10008–10033.

[4] Aritra Ghosh, Himanshu Kumar, and P. Shanti Sastry. 2017. Robust Loss Functions under Label Noise for Deep Neural Networks. In *AAAI Conference on Artificial Intelligence*. https://api.semanticscholar.org/CorpusID:6546734

[5] Aritra Ghosh and Andrew Lan. 2021. Do we really need gold samples for sample weighting under label noise?. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 3922–3931.

[6] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems* 31 (2018).

[7] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), 770–778. https://api.semanticscholar.org/CorpusID:206594692

[8] Dennis Hofmann, Peter VanNostrand, Huayi Zhang, Yizhou Yan, Lei Cao, Samuel Madden, and Elke Rundensteiner. 2022. A demonstration of autood: a self-tuning anomaly detection system. *Proceedings of the VLDB Endowment* 15, 12 (2022), 3706–3709.

[9] Dennis M Hofmann, Peter M VanNostrand, Lei Ma, Huayi Zhang, Joshua C DeOliveira, Lei Cao, and Elke A Rundensteiner. 2025. Agree to Disagree: Robust Anomaly Detection with Noisy Labels. *Proceedings of the ACM on Management of Data* 3, 1 (2025), 1–24.

[10] Ruofan Hu, Dongyu Zhang, Dandan Tao, Huayi Zhang, Hao Feng, and Elke Rundensteiner. 2023. Uce-fid: Using large unlabeled, medium crowdsourced-labeled, and small expert-labeled tweets for foodborne illness detection. In *2023 IEEE International Conference on Big Data (BigData)*. IEEE, 5250–5259.

[11] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. 2017. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109* (2017).

[12] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning*. PMLR, 2304–2313.

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

[14] Junnan Li, Richard Socher, and Steven CH Hoi. 2020. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394* (2020).

[15] Muyang Li, Xiaobo Xia, Runze Wu, Fengming Huang, Jun Yu, Bo Han, and Tongliang Liu. 2024. Towards realistic model selection for semi-supervised learning. In *Forty-first International Conference on Machine Learning*.

[16] Xuefeng Li, Tongliang Liu, Bo Han, Gang Niu, and Masashi Sugiyama. 2021. Provably end-to-end label-noise learning without anchor points. In *International conference on machine learning*. PMLR, 6403–6413.

[17] Yifan Li, Hu Han, S. Shan, and Xilin Chen. 2023. DISC: Learning from Noisy Labels via Dynamic Instance-Specific Selection and Correction. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023), 24070–24079. https://api.semanticscholar.org/CorpusID:260068694

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

[19] Sheng Liu, Zhihui Zhu, Qing Qu, and Chong You. 2022. Robust training under label noise by over-parameterization. In *International Conference on Machine Learning*. PMLR, 14153–14172.

[20] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Monazam Erfani, and James Bailey. 2020. Normalized Loss Functions for Deep Learning with Noisy Labels. In *International Conference on Machine Learning*. https://api.semanticscholar.org/CorpusID:220042066

[21] Deep Patel and PS Sastry. 2023. Adaptive sample selection for robust learning under label noise. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 3932–3942.

[22] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1944–1952.

[23] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *International conference on machine learning*. PMLR, 4334–4343.

[24] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-Weight-Net: Learning an Explicit Mapping For Sample Weighting. In *Neural Information Processing Systems*. https://api.semanticscholar.org/CorpusID:173188221

[25] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems* 33 (2020), 596–608.

[26] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. 2019. Selfie: Refurbishing unclean samples for robust deep learning. In *International conference on machine learning*. PMLR, 5907–5915.

[27] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE transactions on neural networks and learning systems* 34, 11 (2022), 8135–8153.

[28] Haoliang Sun, Qi Wei, Lei Feng, Yupeng Hu, Fan Liu, Hehe Fan, and Yilong Yin. 2024. Variational Rectification Inference for Learning with Noisy Labels. *International Journal of Computer Vision* (2024), 1–20.

[29] Mitchell Keren Taraday and Chaim Baskin. 2023. Enhanced Meta Label Correction for Coping with Label Corruption. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 16295–16304.

[30] Yuanpeng Tu, Boshen Zhang, Yuxi Li, Liang Liu, Jian Li, Jiangning Zhang, Yabiao Wang, Chengjie Wang, and Cai Rong Zhao. 2023. Learning from Noisy Labels with Decoupled Meta Label Purifier. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023), 19934–19943. https://api.semanticscholar.org/CorpusID:256846984

[31] Deng-Bao Wang, Yong Wen, Lujia Pan, and Min-Ling Zhang. 2021. Learning from noisy labels with complementary loss functions. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 10111–10119.

[32] Qi Wang, Yue Ma, Kun Zhao, and Yingjie Tian. 2020. A comprehensive survey of loss functions in machine learning. *Annals of Data Science* (2020), 1–26.

[33] Renzhen Wang, Xixi Jia, Quanziang Wang, Yichen Wu, and Deyu Meng. 2022. Imbalanced Semi-supervised Learning with Bias Adaptive Classifier. In *International Conference on Learning Representations*. https://api.semanticscholar.org/CorpusID:257279756

[34] Wei Wang, Dong-Dong Wu, Jindong Wang, Gang Niu, Min-Ling Zhang, and Masashi Sugiyama. 2025. Realistic Evaluation of Deep Partial-Label Learning Algorithms. *arXiv preprint arXiv:2502.10184* (2025).

[35] Yidong Wang, Hao Chen, Yue Fan, Wang Sun, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, et al. 2022. Usb: A unified semi-supervised learning benchmark for classification. *Advances in Neural Information Processing Systems* 35 (2022), 3938–3961.

[36] Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. 2021. Learning with noisy labels revisited: A study using real-world human annotations. *arXiv preprint arXiv:2110.12088* (2021).

[37] Yang Wu, Huayi Zhang, Yizheng Jiao, Lin Ma, Xiaozhong Liu, Jinhong Yu, Dongyu Zhang, Dezhi Yu, and Wei Xu. 2024. ROSE: A Reward-Oriented Data Selection Framework for LLM Task-Specific Instruction Tuning. *arXiv preprint arXiv:2412.00631* (2024).

[38] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. 2020. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems* 33 (2020), 7597–7610.

[39] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. 2019. Are anchor points really indispensable in label-noise learning? *Advances in neural information processing systems* 32 (2019).

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

[41] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh Thang Luong, and Quoc V. Le. 2019. Unsupervised Data Augmentation for Consistency Training. *Advances in Neural Information Processing Systems* 2020-Decem (4 2019). https://arxiv.org/abs/1904.12848v6

[42] Youjiang Xu, Linchao Zhu, Lu Jiang, and Yi Yang. 2021. Faster meta update strategy for noise-robust deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 144–153.

[43] Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. 2020. Dual t: Reducing estimation error for transition matrix in label-noise learning. *Advances in neural information processing systems* 33 (2020), 7260–7271.

[44] Linya Yi, Sheng Liu, Qi She, Alex McLeod, and Boyu Wang. 2022. On Learning Contrastive Representations for Learning with Noisy Labels. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), 16661–16670. https://api.semanticscholar.org/CorpusID:247223119

[45] LIN Yong, Renjie Pi, Weizhong Zhang, Xiaobo Xia, Jiahui Gao, Xiao Zhou, Tongliang Liu, and Bo Han. 2022. A holistic view of label noise transition matrix in deep learning and beyond. In *The Eleventh International Conference on Learning Representations*.

[46] Suqin Yuan, Lei Feng, and Tongliang Liu. 2025. Early stopping against label noise without validation data. *arXiv preprint arXiv:2502.07551* (2025).

[47] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide Residual Networks. *ArXiv* abs/1605.07146 (2016). https://api.semanticscholar.org/CorpusID:15276198

[48] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. 2021. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems* 34 (2021), 18408–18419.

[49] Dongyu Zhang, Ruofan Hu, and Elke Rundensteiner. 2024. CoLafier: Collaborative Noisy La bel Purifier With Local Intrinsic Dimensionality Guidance. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*. SIAM, 82–90.

[50] Huayi Zhang, Lei Cao, Samuel Madden, and Elke Rundensteiner. 2021. Lancet: labeling complex data at scale. *Proceedings of the VLDB Endowment* 14, 11 (2021).

[51] Huayi Zhang, Lei Cao, Peter VanNostrand, Samuel Madden, and Elke A Rundensteiner. 2021. Elite: Robust deep anomaly detection with meta gradient. In *Proceedings of the 27th ACM SIGKDD*. 2174–2182.

[52] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017).

[53] Huayi Zhang, Binwei Yan, Lei Cao, Samuel Madden, and Elke Rundensteiner. 2024. Metastore: Analyzing deep learning meta-data at scale. Association for Computing Machinery (ACM).

[54] Xinyu Zhang, Hao Jia, Taihong Xiao, Ming-Ming Cheng, and Ming-Hsuan Yang. 2020. Semi-Supervised Learning with Meta-Gradient. In *International Conference on Artificial Intelligence and Statistics*. https://api.semanticscholar.org/CorpusID:220404311

[55] Yihua Zhang, Prashant Khanduri, Ioannis Tsaknakis, Yuguang Yao, Mingyi Hong, and Sijia Liu. 2024. An Introduction to Bilevel Optimization: Foundations and applications in signal processing and machine learning. *IEEE Signal Processing Magazine* 41, 1 (2024), 38–59.

[56] Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. 2021. Learning with feature-dependent label noise: A progressive approach. *arXiv preprint arXiv:2103.07756* (2021).

[57] Zizhao Zhang and Tomas Pfister. 2021. Learning Fast Sample Re-weighting Without Reward Data. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), 705–714. https://api.semanticscholar.org/CorpusID:237431056

[58] Zhilu Zhang and Mert Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems* 31 (2018).

[59] Zhilu Zhang and Mert Rory Sabuncu. 2018. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. *ArXiv* abs/1805.07836 (2018). https://api.semanticscholar.org/CorpusID:29164161

[60] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Scholkopf. 2003. Learning with Local and Global Consistency. In *Neural Information Processing Systems*. https://api.semanticscholar.org/CorpusID:508435

[61] Dawei Zhu, Xiaoyu Shen, Marius Mosbach, Andreas Stephan, and Dietrich Klakow. 2023. Weaker Than You Think: A Critical Look at Weakly Supervised Learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 14229–14253. doi:10.18653/v1/2023.acl-long.796

A Proof for Theorem 1

Theorem 1. *The exponential loss L^{exp} is bounded by*

$$L^{exp} \leq \Pi_{k=1}^{K} R_k^{1/K},$$

where $R_k = \frac{1}{n}\sum_{i=1}^{n} exp(-f_{y_i}^{k}(x_i))$. The upper bound of L^{exp} decreases as K increases.

Proof.

$$\begin{aligned} L^{exp} &= \frac{1}{n}\sum_{i=1}^{n}\exp\left(-\frac{1}{K}\sum_{k=1}^{K} f_{y_i}^{k}(x_i)\right) \\ &= \frac{1}{n}\sum_{i=1}^{n}\prod_{k=1}^{K}\exp\left(-\frac{1}{K} f_{y_i}^{k}(x_i)\right) \end{aligned} \tag{11}$$

Since $\exp(-\frac{1}{K}f_{y_i}^{k}(x_i)) \leq \exp(-f_{y_i}^{k}(x_i))^{\frac{1}{K}}$, we have

$$\prod_{k=1}^{k}\exp\left(-\frac{1}{K}f_{y_i}^{k}(x_i)\right) \leq \prod_{k=1}^{k}\exp\left(-f_{y_i}^{k}(x_i)\right)^{\frac{1}{K}} \tag{12}$$

Thus, the exponential loss can be bounded as:

$$\begin{aligned} L^{exp} &\leq \prod_{k=1}^{K}\left(\frac{1}{n}\sum_{i=i}^{n}\exp(-f_{y_i}^{k}(x_i))\right)^{\frac{1}{K}} \\ &= \prod_{k=1}^{K} R_k^{1/K} \end{aligned} \tag{13}$$

Since $\exp(-f_{y_i}^{k}(x_i)) \leq 1$, we have $R_k \leq 1$, thus the upper bound will decrease as K increases. □

B Pseudocode

Algorithm 1 CLID-based Meta-update Strategy

```
Input: Noisy labeled data $\tilde{D}$, meta dataset $D^{meta}$, Classification model: $f(\cdot;\theta)$, Meta model: $\Omega(\cdot;\psi)$.
Parameters: Labeled data batch size: $n$, Meta data batch size: $m$, maximum iteration: $T$, temperature scaling: $\tau$, learning rate for classification model: $\alpha$, learning rate for meta model: $\gamma$, number of snapshots to retain: $K$
Initialize: $t = 0$, $\mathcal{M} \leftarrow []$. //M is a bounded list of top-K snapshots and the corresponding CLID scores
while $t < T$ do
    $\{(x_i, \tilde{y}_i)\}_{i=1}^{n} \leftarrow$ BatchSampler($\tilde{D}, n$)
    $X^{meta} \leftarrow$ BatchSampler($D^{meta}, m$)
    $L_i(\theta) = l(f(x_i;\theta); \tilde{y}_i)$    //cross-entropy loss
    //Virtual-Train Step:
    $\hat{\theta}^{t+1}(\psi) = \theta^t - \alpha\frac{1}{n}\sum_{i=1}^{n}\Omega(L_i;\psi^t)\nabla_\theta L_i(\theta)\mid_{\theta^t}$
    //CLID-based Meta-Train Step:
    $\psi^{t+1} = \psi^t - \gamma\nabla_\psi L^{clid}(X^{meta};\hat{\theta}^{t+1}(\psi))\mid_{\psi^t}$
    //Actual-Train Step:
    $\theta^{t+1} = \theta^t - \alpha\frac{1}{n}\sum_{i=1}^{n}\Omega(L_i;\psi^{t+1})\nabla_\theta L_i(\theta)\mid_{\theta^t}$
    if EpochEnd(t) then
        $c^t = L^{clid}(D^{meta}, \theta^{t+1})$ //Evaluate the snapshot
        if $|\mathcal{M}| < K$ then
            $\mathcal{M} \leftarrow (\theta^t, c^t)$
        else
            $(\theta_{\max}, c_{\max}) \leftarrow \arg\max_{(\theta,c)\in\mathcal{M}} c$
            if $c^t < c_{\max}$ then
                $\mathcal{M} \leftarrow (\theta^t, c^t)$
            end if
        end if
    end if
    $t = t + 1$
end while
return $\mathcal{M}$
```

C Implementation Details

Comparison with meta-learning methods. We do the implementations following the original work WNet and VRI. For VRI, we use PresNet-18 for all noise settings and train the model for 150 epochs. For WNet, we use WRN-28-10 for all noise settings and train the models for 100 epochs. We employ the Cosine Annealing strategy with a 10-epoch period to adjust the learning rate of the classification network. The initial learning rates are 0.02 for PreResNet-18 and 0.05 for WRN-28-10. For the meta-model, we use a learning rate of 0.01 for VRI and $1e^{-5}$ for WNet. Across all experiments, we set the temperature scaling factor (τ) to 0.5, the meta-dataset size to 1000, and the number of model snapshots K to 5. We use a batch size of 100 for both the training set and the meta-dataset.

Comparison with SOTA. All experiments on CIFAR-10N and CIFAR-100N are conducted using ResNet-34, following prior works. For VRI-CLID, we employ the Cosine Annealing strategy with a 10-epoch period to adjust the learning rate of the classification network, starting with an initial learning rate (lr) of 0.02. Across all experiments, we set the temperature scaling factor (τ) to 0.5, the meta-dataset size to 1000, and the number of model snapshots K to 5. The batch size is 100 for both the training set and the meta-dataset.

For experiments integrating DivideMix, the initial learning rate is set to 0.03 and decays to 1/10 of its value at 120 and 180 epochs, with a total training duration of 300 epochs. We use a batch size of 128 for the training set and 100 for the meta-dataset.

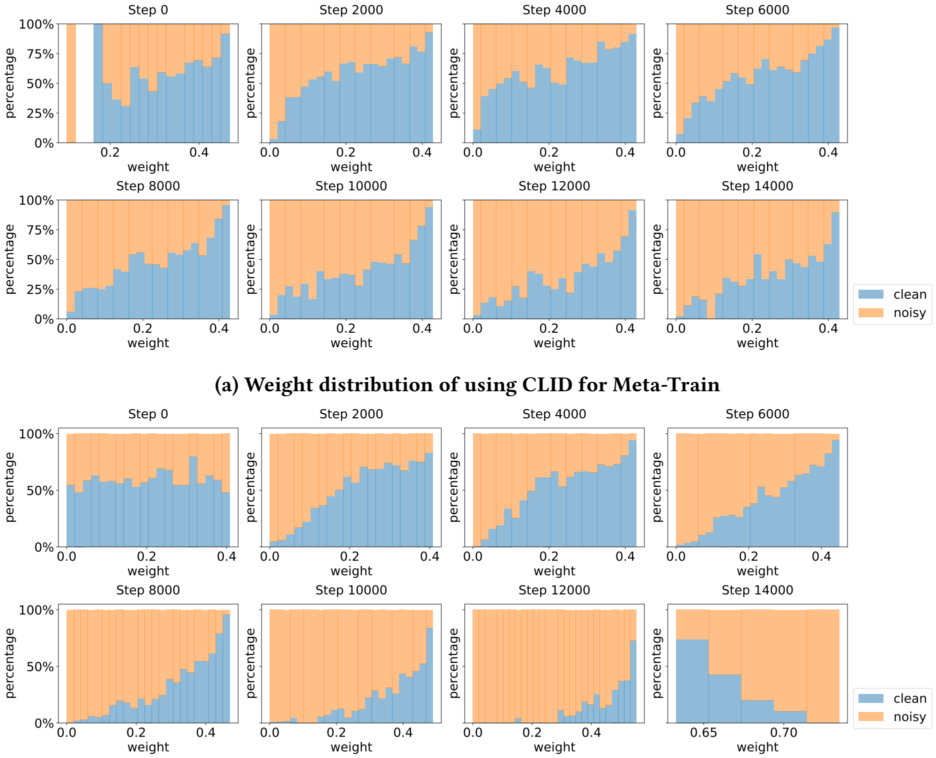
For experiments on Animal-10N, we use VGG19 to remain consistent with prior works. The initial learning rate is set to 0.1, and we apply the Cosine Annealing strategy with a 160-epoch period for learning rate adjustment. We use a batch size of 128 for the training set and 100 for the meta-dataset.

For experiments on Clothing1M, we use the pre-trained Resnet-50 model to remain consistent with prior works. The learning rate is fixed at 0.0005, and the learning rate for the meta-model is fixed at 0.01. All the models were trained for 10 epochs.

Semi-supervised learning experiments. Following the semi-supervised learning benchmarks [35], we used a WRN-28-2 model [47] for all noise settings. The 50,000 training data is split into 4,000 labeled samples and 46,000 unlabeled samples. For CLID-MU, we sampled 1,000 instances from the unlabeled set to construct the meta-dataset, while for WNet-MAE, the meta-dataset was sampled from the labeled set. The classification model was trained using SGD with a momentum of 0.999, a weight decay of $5e^{-4}$, a fixed learning rate of 0.03, and a batch size of 100. The meta-model is trained with a weight decay of $5e^{-4}$ and a batch size of 100 for the meta-dataset. The hyperparameter β and meta model learning rate γ for UDA, FixMatch, and FlexMatch are set to 7,1,7 and $1e^{-5}$,$1e^{-4}$,$1e^{-5}$, respectively.

D Computational Complexity

CLID-MU introduces additional computational overhead compared to baseline methods. Empirically, training with CLID-MU on a single NVIDIA A100 GPU requires approximately 140 seconds per epoch using a PreResNet18 backbone, whereas baseline methods complete an epoch in roughly 14 seconds. To mitigate this overhead, we propose several optimization strategies for future work. (1) Instead of computing all pairwise similarities within a batch, we can first construct a sparse class probability graph by connecting each node only to its top-K most similar nodes. The corresponding sparse embedding graph is then built using those connections. This reduces the computational complexity from $O(m^2)$ to $O(Km)$, where m is the batch size and $K \ll m$. To further accelerate this step, Approximate Nearest Neighbor (ANN) techniques can be employed using the library FAISS, reducing the complexity to $\sim O(mlogm)$. (2) Another optimization is to reduce the frequency of meta-model updates by computing the CLID loss once every N steps instead of at every iteration. (3) Meta-model updates may be terminated once the CLID loss converges, thereby eliminating redundant computations in the later stages of training.



(a) Weight distribution of using CLID for Meta-Train

(b) Weight distribution of using MAE for Meta-Train

Figure 6: The weight distribution of clean and noisy samples in the experiment of FlexMatch on CIFAR-10N (Worst).

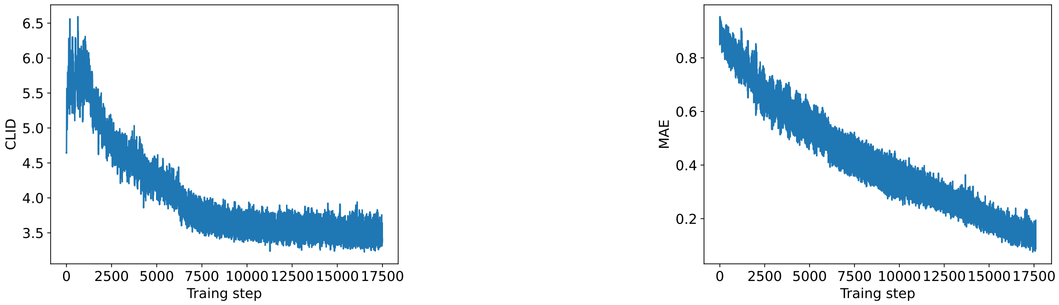


Figure 7: Trend of meta loss used in the Meta-Train step in the experiment of FlexMatch on CIFAR-10N Worst label. Left: our CLID-based Meta-update strategy. Right: MAE-based Meta-update strategy using noisy labeled data as the meta dataset.

E Weight Distribution

Using the semi-supervised learning experiment with FlexMatch as an example, we divided the weights into equal-length bins and visualized the percentage distribution of clean and noisy samples in each bin, as shown in Figure 6. The weights generated by CLID-MU are more stable, with most of the larger weights being assigned to clean samples. In contrast, the weights generated by the MAE-based Meta-update tend to increasingly assign higher weights to noisy samples as training progresses.

This phenomenon is not due to the larger learning rate used in training with MAE, nor is it a result of overfitting. By examining the trend of the meta loss in Figure 7, we observed that the MAE loss has not yet converged, indicating that MAE struggles to effectively measure model performance under complex noise patterns.