EFFICIENT UTILIZATION OF PRE-TRAINED MODEL FOR LEARNING WITH NOISY LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

In machine learning, when the labels within a training dataset are incorrect, the performance of the trained model gets severely affected. To address this issue, various methods have been researched in the field of *Learning with Noisy Labels*. These methods aim to identify the accurate samples and focus on them, while minimizing the impact of incorrect labels. Recent studies have demonstrated good performance on various tasks using large pre-trained models that extract good features regardless of the given labels. However, to address the noisy label problem, leveraging these pre-trained models have still remained unexplored due to the computational cost of fine-tuning. In this study, we propose an algorithm named `EPL` that utilizes pre-trained models to effectively cleanse the noisy labels and strengthen the robust training. The algorithm follows two main principles: (1) increasing computational efficiency by adjusting the linear classifier alone, and (2) cleaning only the well-clustered classes to avoid creating extra incorrect labels in poorly-clustered classes. We tested and verified that the proposed algorithm shows significant improvement on various benchmarks in comparison to previous methods.

1 INTRODUCTION

Deep neural networks (DNNs) have demonstrated impressive performance on various tasks such as classification He et al. (2016), generation Goodfellow et al. (2020), and object detection He et al. (2017). However, performance of DNNs drops significantly when corrupted annotations are provided. In addition, manually correcting noisy labels or newly obtaining clean labels is difficult due to the large dataset size. To address this problem, researchers have developed various approaches within the field of *Learning with Noisy Labels* (LNL), such as robust training loss Zhang & Sabuncu (2018); Wang et al. (2019), regularizer Cao et al. (2020); Cheng et al. (2023), sample selection Han et al. (2018); Yu et al. (2019), and semi-supervised learning method Li et al. (2020); Liu et al. (2020); Karim et al. (2022). However, recent studies have shown limited performance improvement with increasing algorithm complexity. Thus, a new direction is required to effectively improve LNL performance.

Utilizing pre-trained models (PTMs) can be a promising new direction to address this problem. In recent years, an increasing number of studies have focused on developing large PTMs with high adaptability to various tasks such as natural language processing (NLP) Devlin et al. (2019); Brown et al. (2020) and computer vision (CV) Dosovitskiy et al. (2021); Liu et al. (2022b). These PTMs are easier to access Wolf et al. (2020) and have shown remarkable performance in diversified applications owing to their powerful feature extractors, which can be attributed to their enormous trainable parameters and well-curated large training datasets such as ImageNet-21K; 14 million images across $21,841$ classes.

With their remarkable performance, PTMs potentially can reduce human effort in purifying the noisy labels in the given training datasets Zhu et al. (2022). However, only a few studies have leveraged PTMs for noisy label datasets because of two main reasons: (1) fine-tuning entire PTMs to adapt to new datasets is computationally expensive because of large number of parameters, (2) using PTMs by only adjusting a part of the parameters (*e.g.,* linear probing) often results in poor performance on classes that are dissimilar to the training dataset (*e.g.,* ImageNet-21K) Zhuang et al. (2020); Guo et al. (2019); Lee et al. (2022).

Therefore, our goal is to design an algorithm that follows two philosophies: (1) efficiently utilizing PTMs by only updating a fraction of their parameters (*i.e.,* linear probing), (2) preventing the additional noisy labels by restricting PTMs to purify only for familiar classes.

Contribution. In this study, we propose a simple yet effective method that utilizes PTMs for purifying noisy-labeled datasets. Our main observations and contributions are summarized as follows:

- We provide theoretical evidence that creation of additional noisy labels can be prevented using a model that can make good clusters. In addition, we empirically demonstrated two types of classes that can or cannot be effectively handled by PTMs. We observed that the cleansing performance was lower for classes where PTMs performed limited clustering. (Appendix A, B)
- Based on our findings, we designed a method called **E**fficient utilization of **P**re-trained model for **L**earning with Noisy Labels (`EPL`), which corrects the given corrupted dataset by efficiently leveraging PTMs for the classes which are regarded as confident on their own. This method avoids generating additional label noise. (Section 2)
- We show that our method effectively works in conjunction with existing LNL methods in a variety of datasets, including synthetically noisy-labeled datasets (*e.g.,* CIFAR-10/100, EuroSAT, DTD, and Oxford-IIIT Pet) and real-world datasets (*e.g.,* WebVision, Clothing1M). (Section 3)

2 EPL: EFFICIENT UTILIZATION OF PRE-TRAINED MODEL FOR LEARNING WITH NOISY LABELS

The core philosophy of `EPL` is to "*fully exploit the helpful information of PTM in well-clustered classes without introducing additional label corruption in poorly-clustered classes.*" In this section, we describe the design of our proposed method, `EPL` which consists of (1) linear probing with preventing memorization (2) classwise consistency check, and (3) label correction and running one of the robust training methods. We demonstrate the overall algorithm of `EPL` in Algorithm 1.

2.1 LINEAR PROBING WITH PREVENTING MEMORIZATION

The results presented in Appendix A.2 demonstrate that utilizing linear probing in the presence of noisy labels can improve cleansing performance by simply preventing memorization with noise-robust loss functions. Building on this finding, for linear probing, we employ the ELR loss function Liu et al. (2020) which is defined as follows:

$$\mathcal{L}_{\text{ELR}}(f(\boldsymbol{x}), \tilde{y}) = \mathcal{L}_{\text{CE}}(f(\boldsymbol{x}), \tilde{y}) + \lambda \log(1 - \langle \mathbf{p}, \mathbf{t} \rangle), \tag{1}$$

where $\langle \mathbf{p}, \mathbf{t} \rangle$ is the inner product of the softmax output of the model, $\mathbf{p} := \text{Softmax}(f(\boldsymbol{x}))$, and the moving average (MA) value of the model output $\mathbf{t} \leftarrow \beta\mathbf{t} + (1-\beta)\mathbf{p}$ with MA parameter β, respectively. Note that λ is the weight for the regularization term of the ELR loss function.

2.2 CLASSWISE CONSISTENCY CHECK (C3)

We propose a method called classwise consistency check (C3) to effectively use PTM for noisy label detection without performance degradation by using its results only for well-clustered classes. C3 includes two elements: instance-centric consistency to assess the consistency of predictions for augmented images and model-centric consistency using an ensemble of linear classifiers to prevent incorrect separation owing to memorization of noisy labeled data.

Instance-centric consistency. To determine whether the class is well-clustered with a particular PTM, we evaluate the ability of the PTM to accurately predict augmented versions by extracting the information related to the class. As the decision boundary of a linear classifier is primarily influenced by feature vectors with clean labels, using clean instances for consistency checks is likely to produce more reliable results than using noisy instances. Hence, we divide the training datasets into instances with clean labels and with noisy labels using a Gaussian mixture model (GMM), a method commonly employed in previous research Li et al. (2020); Kim et al. (2021) to differentiate between clean and noisy instances based on their own objectives.

By utilizing K random augmentation operations $\mathcal{A}_k : \mathbb{R}^d \to \mathbb{R}^d$ with $k \in \{1, \dots, K\}$, we are able to generate K variants, randomly augmented datasets from a single original dataset, denoted

algo2e 1 Pseudo code of EPL

Input: Dataset $D = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^N$, Linear classifiers $f_1, \ldots, f_M$, Frozen feature extractor g, Threshold γ
Output: Cleansed dataset $\bar{D}$
Initialize: $\bar{D} = \emptyset, \mathcal{W} = \emptyset$
/* Linear probing */
Train $f_1, \ldots, f_M$ using g and $\mathcal{L}_{\text{ELR}}$ defined at Eq. (1)
/* Select well-clustered classes */
for $c \in \{1, \ldots, C\}$ **do**
  for $m \in \{1, \ldots, M\}$ **do**
    Select a confident samples set D_c by using GMM on outputs of f_m
    Compute $S_c^{\text{ICC}}(f_m)$ score by following Eq. (2)
  end
  if $S_c^{\text{ICC}}(f_m) \geq \gamma \;\; \forall m \in \{1, \ldots, M\}$ **then**
    $\mathcal{W} = \mathcal{W} \cup \{c\}$
  end
end
/* Label Correction */
for $(\boldsymbol{x}_i, \tilde{y}_i) \in D$ **do**
  Label prediction of $\boldsymbol{x}_i$ with model ensemble, $\hat{y}_i = \arg\max_{c \in \{1,\ldots,C\}} \sum_{m=1}^M f_m \circ g(\boldsymbol{x}_i)$

$$\bar{D} = \begin{cases} \bar{D} \cup \{(\boldsymbol{x}_i, \hat{y}_i)\} & \text{if } \hat{y}_i \in \mathcal{W} \\ \bar{D} \cup \{(\boldsymbol{x}_i, \tilde{y}_i)\} & \text{otherwise.} \end{cases}$$

end

as $\mathcal{A}_1(\boldsymbol{x}), \ldots, \mathcal{A}_K(\boldsymbol{x})$. On the set of instances with label c that are potentially clean, obtained by using GMM, denoted as $D_c = \{\boldsymbol{x} | (\boldsymbol{x}, \tilde{y}) \in D, \tilde{y} = c\}$, we define $S_c^{\text{ICC}}(f)$ as the instance-centric consistency (ICC) score for class c, calculated as $S_c^{\text{ICC}}(f) = \frac{1}{K|D_c|} \cdot \sum_{\boldsymbol{x} \in D_c} \mathcal{C}(f, \boldsymbol{x})$ where

$$\mathcal{C}(f, \boldsymbol{x}) = \sum_{k=1}^{K} \mathbf{1}_{\{f(\mathcal{A}_k(\boldsymbol{x}))=c\}}, \tag{2}$$

where f denotes the linear classifier which is trained on D. If $S_c^{\text{ICC}}(f) \geq \gamma$, we consider class c as a well-clustered class where $\gamma \in [0, 1]$ is the threshold hyperparameter for separating well/poorly-clustered classes.

Model-centric consistency. If the representation power of the PTM is limited to depict a particular class, specifically when the size of the PTM and its pre-training dataset is small, the model memorizes the noisy labels rather than training the clean instances. In this situation, the results of class separation based on the ICC score are different according to the initialization of the linear classifier, which can result in wrong separation of well-clustered classes and performance degradation of the target model. To mitigate the risk of inaccurate separation caused by a limited feature extractor and a single linear classifier, we use a model-centric consistency check that employs an ensemble of multiple linear classifiers with varying initializations. Through its implementation, a class is considered well-clustered only if all classifiers in the ensemble are in agreement regarding its clusterability.

2.3 Label Correction and Running Robust Methods

On separation of well- and poorly-clustered classes using the C3 module, we can reliably obtain the corrected labels by utilizing the PTM in well-clustered classes. Therefore, we re-label the samples whose predicted labels are in the well-clustered classes as the predicted labels. As the samples in the poorly-clustered classes are not handled by PTM, they must be purified. Hence, we run the LNL algorithms, such as ELR+ Liu et al. (2020), DivideMix Li et al. (2020), and UNICON Karim et al. (2022), to mitigate the degradation from the noisy labels in the poorly-clustered classes.

Table 1: Comparison with state-of-the-art LNL algorithms in test accuracy (%) on CIFAR-10 and CIFAR-100 datasets with symmetric, asymmetric, and instance noise. +EPL denotes the performance when the noisy labeled dataset is cleansed based on the proposed method with the corresponding PTM. † indicates reported results from the original work. The best results sharing the noisy fraction (the noisy fraction and method) are highlighted in bold (underline). We report both the best/last performance for each experiment.

Method	CIFAR-10				CIFAR-100			
	Symm. 0.6	*Symm.* 0.9	*Asym.* 0.4	*Inst.* 0.4	*Symm.* 0.6	*Symm.* 0.9	*Asym.* 0.4	*Inst.* 0.4
CE	77.8 / 40.9	43.7 / 16.0	85.5 / 74.5	74.7 / 53.7	45.0 / 23.2	11.4 / 3.6	46.3 / 39.6	43.3 / 35.4
GCE	83.6 / 75.4	46.4 / 21.4	81.3 / 71.4	10.0 / 10.0	57.0 / 49.5	14.4 / 11.0	51.3 / 45.6	1.0 / 1.0
ELR+	93.5 / 93.1	78.7 / 76.0	91.3 / 85.7	65.3 / 64.3	69.6 / 69.0	33.4 / 32.4	73.9 / 73.6	57.4 / 56.1
+ EPL (ConvNeXt-XL)	95.3 / 95.3	95.3 / 95.2	93.4 / 93.3	94.6 / 94.5	75.9 / 75.9	67.2 / 66.8	76.4 / 76.4	68.4 / 68.1
+ EPL (ViT-L/14-CLIP)	95.4 / 95.3	95.4 / 95.1	92.8 / 92.6	93.1 / 92.7	73.7 / 73.3	41.8 / 41.6	75.9 / 75.8	61.4 / 60.7
+ EPL (ViT-B/16)	95.0 / 94.8	94.9 / 94.7	92.6 / 91.2	92.7 / 92.2	75.1 / 75.0	61.7 / 61.5	75.2 / 75.0	61.9 / 61.8
+ EPL (ViT-L/16)	95.5 / 95.4	95.5 / 95.3	93.9 / 93.9	95.4 / 95.3	75.4 / 75.2	71.7 / 71.7	**77.1 / 76.7**	69.5 / 69.4
DivideMix	94.8 / 94.6	76.0† / 75.4†	93.4† / 92.1†	92.2 / 90.5	71.8 / 71.2	31.5† / 31.0†	60.8 / 54.6	63.8 / 63.4
+ EPL (ConvNeXt-XL)	95.1 / 94.9	95.0 / 94.8	94.4 / 93.9	95.2 / 94.9	74.5 / 74.0	68.4 / 67.6	74.4 / 74.3	72.9 / 72.9
+ EPL (ViT-L/14-CLIP)	95.0 / 94.7	94.8 / 94.5	94.3 / 93.9	94.9 / 93.9	74.2 / 73.7	54.0 / 53.1	71.8 / 71.3	67.5 / 66.8
+ EPL (ViT-B/16)	95.4 / 94.9	95.0 / 94.7	94.2 / 93.7	95.2 / 95.1	73.9 / 72.1	66.9 / 66.6	73.7 / 73.2	68.4 / 68.0
+ EPL (ViT-L/16)	95.4 / 95.2	95.3 / 94.9	94.7 / 94.4	95.5 / 95.3	74.2 / 73.6	71.3 / 70.5	74.7 / 74.2	72.5 / 72.1
UNICON	95.0 / 94.3	89.8 / 89.0	94.1 / 93.7	94.6 / 94.4	74.5 / 73.5	43.8 / 42.6	73.1 / 71.4	73.8 / 71.4
+ EPL (ViT-L/16)	**96.1 / 96.0**	**96.0 / 95.8**	**96.0 / 95.8**	**95.9 / 95.8**	**76.2 / 74.7**	**72.3 / 70.5**	76.5 / 74.7	**77.8 / 76.3**

3 EXPERIMENTS

In this section, we present empirical evaluation, which demonstrates the superior performance and computational efficiency of our proposed algorithm for robust training under the presence of noisy labels. We first described the LNL benchmarks and implementations in detail (Section 3.1). Then, we described the experimental results on a substantial synthetic (CIFAR-10/100, EuroSAT, DTD, Oxford-IIIT-Pet) and real-world (Clothing1M, WebVision) noisy labeled datasets in Section 3.2. Moreover, we conduct additional experiments to obtain a better understanding of EPL, and this analysis is provided in Section 3.3.

3.1 EXPERIMENTAL SETUP

Datasets. We first evaluate EPL on the most commonly used noisy labeled image classification tasks: CIFAR-10/-100, Clothing1M Xiao et al. (2015), and WebVision Li et al. (2017). For CIFAR datasets, we injected uniform randomness into a fraction of labels for symmetric noise and flipped labels to specific classes for asymmetric noise by following Liu et al. (2020). To set up instance-dependent noise, we followed the noise generation of Cheng et al. (2021).

In addition to commonly used benchmarks, we also performed experiments on other synthetic, noisy labeled datasets which were created by introducing artificial noise to EuroSAT Helber et al. (2019), DTD Cimpoi et al. (2014), and Oxford-IIIT Pet Parkhi et al. (2012). In these additional datasets, the noise generation is the same as in CIFAR datasets. A detailed explanation of datasets is described in Appendix C.1.

Table 2: Comparison with state-of-the-art LNL algorithms in test accuracy (%) on EuroSAT, DTD, and Oxford-IIIT Pet datasets with symmetric, asymmetric, and instance noise. +EPL denotes the performance when the noisy labeled dataset is cleansed based on the proposed method with the corresponding PTM. The best results sharing the noisy fraction (the noisy fraction and method) are highlighted in bold (underline). We report both the best/last performance for each experiment.

Method	EuroSAT			DTD			Oxford-IIIT Pet		
	Symm. 0.6	*Asym.* 0.4	*Inst.* 0.4	*Symm.* 0.6	*Asym.* 0.4	*Inst.* 0.4	*Symm.* 0.6	*Asym.* 0.4	*Inst.* 0.4
CE	70.8 / 70.0	75.9 / 71.4	77.2 / 75.4	67.7 / 66.2	66.4 / 66.2	69.8 / 67.7	52.3 / 47.1	59.5 / 58.5	65.5 / 65.1
ELR+	73.4 / 72.7	77.3 / 76.6	79.6 / 79.6	74.1 / 72.1	73.7 / 73.2	71.8 / 70.2	73.0 / 72.6	72.1 / 71.5	78.5 / 78.5
+ EPL (ConvNeXt-XL)	86.0 / 86.0	81.3 / 79.3	87.6 / 87.6	78.9 / 78.1	76.9 / 75.8	76.6 / 76.4	78.0 / 77.9	76.4 / 76.3	82.2 / 82.1
+ EPL (ViT-L/16)	75.6 / 75.2	78.6 / 78.4	84.6 / 84.0	76.6 / 75.9	75.1 / 74.3	72.4 / 72.3	78.3 / 78.3	76.5 / 76.5	82.3 / 82.3
DivideMix	82.1 / 81.9	86.5 / 85.4	83.5 / 82.6	77.4 / 77.1	76.8 / 76.1	73.5 / 73.1	73.9 / 67.1	64.9 / 59.5	71.2 / 70.1
+ EPL (ConvNeXt-XL)	**93.6 / 92.0**	**92.0 / 91.4**	**93.3 / 92.7**	**82.6 / 82.2**	83.1 / 82.7	80.8 / 80.4	77.7 / 76.0	73.0 / 70.5	81.1 / 80.3
+ EPL (ViT-L/16)	88.6 / 85.7	87.2 / 86.9	92.9 / 92.8	79.8 / 79.5	80.6 / 80.1	75.9 / 75.2	77.9 / 76.1	73.2 / 71.6	79.2 / 77.8
UNICON	81.3 / 80.4	85.6 / 84.0	84.0 / 82.9	80.9 / 80.9	77.3 / 77.2	75.0 / 74.8	79.6 / 79.6	75.2 / 74.4	80.5 / 80.1
+ EPL (ConvNeXt-XL)	90.3 / 90.2	89.6 / 89.3	87.0 / 86.4	82.5 / 82.3	**83.6 / 82.5**	**81.4 / 80.9**	81.2 / 81.2	80.1 / 79.5	84.3 / 84.0
+ EPL (ViT-L/16)	89.5 / 89.3	86.1 / 85.7	87.1 / 87.0	81.7 / 81.5	79.9 / 79.7	76.1 / 75.5	**82.0 / 81.7**	**80.7 / 80.4**	**85.3 / 85.3**

Table 3: Comparison with LNL algorithms in test accuracy (%) on Clothing1M and WebVision. +EPL denotes the performance with our proposed method through the corresponding PTM. † indicates reported results from the original work. The best results sharing the dataset (the dataset and method) are highlighted in bold (underline). For WebVision and ILSVRC12, we report both top-1/top-5 accuracies.

	Clothing1M			WebVision (WebVision)			WebVision (ILSVRC12)		
	DivideMix	ELR+	UNICON	DivideMix	ELR+	UNICON	DivideMix	ELR+	UNICON
Baseline	74.76†	74.81†	74.98†	77.32† / 91.64†	77.78† / 91.68†	77.60† / 93.44†	75.20† / 90.84†	70.29† / 89.76†	75.29† / 93.72†
+ EPL (ViT-L/14-CLIP)	75.12	**75.21**	75.18	77.53 / 92.89	77.94 / 92.62	77.75 / 93.74	75.47 / 91.74	73.11 / 90.21	75.93 / 93.79
+ EPL (ConvNext-XL)	75.04	75.13	75.14	**78.77 / 93.31**	78.43 / 93.41	78.23 / 93.70	**76.51 / 92.54**	74.28 / 90.70	76.32 / 93.81
+ EPL (ViT-L/16)	75.02	75.16	75.07	78.26 / 92.71	78.35 / 92.76	78.04 / 93.65	76.36 / 92.76	73.57 / 90.51	76.08 / 93.86

Implementations. We integrate EPL with ELR+ Liu et al. (2020), DivideMix Li et al. (2020), and UNICON Karim et al. (2022). To verify our proposed method effectiveness, we compared the test performance of the same approaches with and without applying EPL. For hyperparameter, we trained five linear classifiers for a total of 15 training epochs across all datasets and PTMs. Furthermore, we applied the C3 module with $\gamma = 0.8$ at the end of the first epoch to differentiate between well/poorly-clustered classes. We applied various types of PTMs for each dataset: ConvNeXt Liu et al. (2022b), ViT-CLIP Radford et al. (2021), ViT Dosovitskiy et al. (2021). ViT-CLIP refers to the ViT architecture that has been pre-trained using the CLIP method. Other PTMs are pre-trained with ImageNet-1K or ImageNet-21K datasets. All pre-trained weights of PTMs are from HuggingFace Wolf et al. (2020). We describe the detailed implementation in Appendix C.3.

3.2 Experimental Results

In this section, we report the performance of the method compared to the benchmark-simulated (CIFAR, EuroSAT, DTD, Oxford-IIIT-Pet) and real-world (Clothing1M, WebVision) datasets.

CIFAR datasets. For CIFAR datasets, we combine our proposed method with various algorithms: ELR+, DivideMix, and UNICON. In Table 1, EPL with all types of PTMs consistently improves the performance regardless of the LNL method, noise rates, and noise distributions without applying complicated methodological modification. ConvNeXt-XL and ViT-L/16 are most effective because of their well-constructed feature extractor with a large number of parameters and pre-training dataset size.

Additional synthetic datasets. To verify the effectiveness of EPL, we conducted experiments on additional synthetic datasets: EuroSAT, DTD, and Oxford-IIIT-Pet. Table 2 summarizes the performance of existing methods and the performance gain when integrated with EPL. The datasets are fine-grained or out-of-domain, which the PTMs have not encountered during the pre-training phase. Nevertheless, the C3 module in EPL effectively separates the well/poorly-clustered classes, enabling the model to maintain or even improve its performance without any decrease.

Real-world datasets. To evaluate EPL performance on real-world datasets where noisy labels can occur easily, we conduct experiments on Clothing1M Xiao et al. (2015) and WebVision Li et al. (2017). Table 3 summarizes the performance of various LNL methods and the gain in performance when integrated with EPL. ViT-L/14-CLIP showed the largest performance gain for Clothing1M, while ConvNeXt-XL and ViT-L/16 showed larger increments in WebVision. These results indicate the importance of the similarity in the noisy target dataset and pre-training dataset for training the PTM.

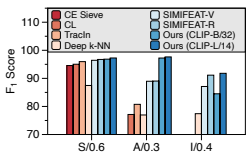


Fig. 1: Detection perf.

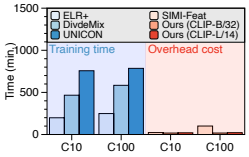


Fig. 2: Comp. overhead

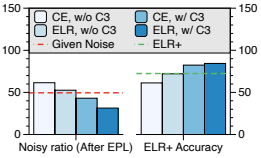


Fig. 3: Component study

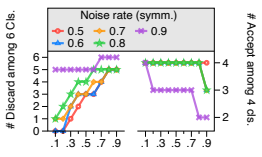


Fig. 4: γ sensitivity

3.3 ANALYSIS

We designed our analyses to answer the following questions. (1) Does EPL perform better than other noisy label detection methods? (2) How much additional computation does EPL require? (3) Which part of EPL is important for improved performance? (4) Does EPL divide well/poorly-clustered classes better? These analyses provide additional explanations to understand EPL.

Detection performance. To verify the impact of EPL, we examined the detection performance on CIFAR-10 with various noise rates of symmetric noise. We compared six detection methods, including CE sieve, confident learning (CL; Northcutt et al. 2021), TracIn Pruthi et al. (2020), Deep KNN Bahri et al. (2020), and SimiFeat-V/R Zhu et al. (2022). As shown in Figure 1, except for ViT-B/32-CLIP on instance-dependent noise with 0.4 noise rates, our proposed method outperformed the other noisy label detection methods. While we achieved the highest performance with the larger model ViT-L/14-CLIP, the EPL with ViT-L/14-CLIP required a smaller cost than the previous SoTA method SimiFeat with ViT-B/32-CLIP, as shown in Figure 2.

Computational efficiency. We verified the computational efficiency of the proposed algorithm. All experiments were conducted on CIFAR-10/100 with symmetric label noise under noise rate 0.4 according to Liu et al. (2020). As shown in Figure 2, while the baselines required several hours for training, our proposed approach only required relatively 5%-15% additional computational costs. From the results, our proposed method was computationally efficient due to the simplicity of linear probing.

Component analysis. Using the CIFAR-10 dataset with 0.5 noisy ratio and the ViT-B/32-CLIP model, we ran experiments to evaluate the performance of each component. We compared four cases: (1) PTM is trained on CE and uses the predictions of all samples as the cleansed labels; (2) PTM is trained on ELR and uses the predictions of all samples as the cleansed labels; (3) PTM is trained on CE and uses predictions filtered by the C3 module as the cleansed labels; and (4) PTM is trained on ELR and uses predictions filtered by the C3 module as the cleansed labels. Here, the fourth case can be considered as complete EPL setting. The results, as shown in Figure 3, indicate that using PTM labels learned by CE and ELR without the C3 module leads to increased noisy ratio, negatively impacting ELR+ training performance. However, using the C3 module improved the ELR+ performance, and the best performance was observed when both ELR loss and C3 modules were used. Our proposed algorithm, EPL, was able to empower the LNL algorithm.

Parameter sensitivity. For evaluating the sensitivity of hyperparameters, we determined the number of classes that were discarded/accepted by our C3 module among six poorly-clustered and four well-clustered classes in Figure 6. As described in Figure 4, our C3 module discards poorly-clustered classes as γ increases. However, when γ increases, the number of accepted classes among four classes is reduced. However, almost all well-clustered classes are accepted by the C3 module except for the case of noise rate of 0.9. Through the results, we verified that EPL was robust to the selection of γ except in the extreme noise case.

4 CONCLUSION

This study proposes an algorithm, referred to as EPL, which addresses the problem of noisy labels through large PTMs without updating the feature extractor. Our findings indicate that linear probing of PTMs can effectively detect noisy labels for well-clustered classes; however, it may inadvertently introduce additional label corruption for poorly-clustered classes. To address this problem, a combination of data-centric and model-centric consistency modules was used to separate the training dataset into well/poorly-clustered classes and PTMs were applied only to noisy label identification and correction for well-clustered classes. This proposed approach improves performance on synthetic and real-world benchmarks and effectively distinguishes well/poorly-clustered classes with very few computational costs.

REFERENCES

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. *arXiv preprint arXiv:2202.03555*, 2022.

Dara Bahri, Heinrich Jiang, and Maya Gupta. Deep k-nn for noisy labels. In *International Conference on Machine Learning*, pp. 540–550. PMLR, 2020.

Mikhail Belkin, Daniel J Hsu, and Partha Mitra. Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate. *Advances in neural information processing systems*, 31, 2018.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Kaidi Cao, Yining Chen, Junwei Lu, Nikos Arechiga, Adrien Gaidon, and Tengyu Ma. Heteroskedastic and imbalanced deep learning with adaptive regularization. *arXiv preprint arXiv:2006.15766*, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=2VXyy9mIyU3`.

Hao Cheng, Zhaowei Zhu, Xing Sun, and Yang Liu. Mitigating memorization of noisy labels via regularization between representations. In *Submitted to The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=6qcYDVlVLnK`. under review.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris. Spottune: transfer learning through adaptive fine-tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4805–4814, 2019.

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. *Advances in neural information processing systems*, 31, 2018.

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*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

Nazmul Karim, Mamshad Nayeem Rizve, Nazanin Rahnavard, Ajmal Mian, and Mubarak Shah. Unicon: Combating label noise through uniform selection and contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9676–9686, 2022.

Taehyeon Kim, Jongwoo Ko, JinHwan Choi, Se-Young Yun, et al. Fine samples for learning with noisy labels. *Advances in Neural Information Processing Systems*, 34:24137–24149, 2021.

Jongwoo Ko, Bongsoo Yi, and Se-Young Yun. Alasca: Rethinking label smoothing for deep learning under label noise. *arXiv preprint arXiv:2206.07277*, 2022.

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

Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin. Robust inference via generative classifiers for handling noisy labels. In *International Conference on Machine Learning*, pp. 3763–3772. PMLR, 2019.

Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. *arXiv preprint arXiv:2210.11466*, 2022.

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

Shikun Li, Xiaobo Xia, Shiming Ge, and Tongliang Liu. Selective-supervised contrastive learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 316–325, 2022.

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.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 100–114, Dublin, Ireland and Online, May 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.deelio-1.10. URL https://aclanthology.org/2022.deelio-1.10.

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.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022b.

Curtis G Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. URL https://openreview.net/forum?id=XccDXrDNLek.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33: 19920–19930, 2020.

Xingye Qiao, Jiexin Duan, and Guang Cheng. Rates of convergence for large-scale nearest neighbor classification. *Advances in neural information processing systems*, 32, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. URL https://openreview.net/forum?id=Zkj_VcZ6ol.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

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.

Alexander B Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

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*, pp. 322–330, 2019.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Xiaobo Xia, Tongliang Liu, Bo Han, Mingming Gong, Jun Yu, Gang Niu, and Masashi Sugiyama. Sample selection with uncertainty of losses for learning with noisy labels. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xENf4QUL4LW.

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*, pp. 2691–2699, 2015.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

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*, pp. 7164–7173. PMLR, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy8gdB9xx.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. Differentiable prompt makes pre-trained language models better few-shot learners. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=ek9a0qIafW.

Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747*, 2020.

Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 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*, pp. 1657–1667, 2022.

Songzhu Zheng, Pengxiang Wu, Aman Goswami, Mayank Goswami, Dimitris Metaxas, and Chao Chen. Error-bounded correction of noisy labels. In *International Conference on Machine Learning*, pp. 11447–11457. PMLR, 2020.

Xiong Zhou, Xianming Liu, Junjun Jiang, Xin Gao, and Xiangyang Ji. Asymmetric loss functions for learning with noisy labels. In *International conference on machine learning*, pp. 12846–12856. PMLR, 2021.

Zhaowei Zhu, Zihao Dong, and Yang Liu. Detecting corrupted labels without training a model to predict. In *International Conference on Machine Learning*, pp. 27412–27427. PMLR, 2022.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

- Supplementary - Efficient Utilization of Pre-trained Model for Learning with Noisy Labels

Due to the page limitation of the main manuscript, we provide detailed information in this supplementary document as follows. (1) In Appendix A, we provide our brief motivation for utilizing PTMs under the presence of label noise. (2) In Appendix B, we describe our theoretical backups, which are briefly described in Appendix A. (3) In Appendix C, we summarize the experimental setup, including datasets and preprocessing, and explain the PTMs used in our experiments.

A Pre-trained Models under Label Noise

Our fundamental concept is to construct a robust training framework to label noise through the extremely efficient utilization of large PTMs. Other possible design choices for using large PTMs exist but with huge computational costs. For instance, by following the philosophy of C2D Zheltonozhskii et al. (2022), PTMs can be used as a good initial point, *i.e.,* fine-tuning the large PTMs such as ViT-L/16 Dosovitskiy et al. (2021), for applying DivideMix Li et al. (2020). However, such approaches are computationally expensive because of the large size of the PTMs. Hence, we focus on (1) efficiently applying large PTMs for cleansing the given noisy labeled dataset and (2) applying the cleansed dataset to the previous works Li et al. (2020); Liu et al. (2020); Karim et al. (2022).

To achieve this aim, we first study the characteristics of the PTMs under the label-corrupted dataset. We present two pieces of theoretical evidence. First, when a noisy labeled dataset is cleaned, the probability of creating additional label corruption is limited by the smallest error in estimating either the corrupted or the true conditional distribution of the target dataset. Second, when PTMs make class-wise clusters, the error of estimating the true conditional distribution established in the first evidence is significantly reduced. Subsequently, we illustrate two empirical observations: (1) the power of PTMs for cleansing noisy labels and (2) the limitation of efficient linear probings when PTMs cannot extract proper features for some poorly-clustered classes.

A.1 Theoretical Motivations

Before introducing our theoretical motivations, we formally describe the notations and problem formulations that are focused on. Subsequently, theoretical motivations are derived based on our formulations.

Notation. We focus on binary classification. Assuming the data points and labels lie in $\mathcal{X} \times \mathcal{Y}$, where the feature space $\mathcal{X} \subset \mathbb{R}^d$ and label space $\mathcal{Y} = \{0, 1\}$. A single data point $\boldsymbol{x}$ and its label y follow the joint probability distribution $(\boldsymbol{x}, y) \sim \mathcal{D}$ that can be factored as $\mathcal{D}(\boldsymbol{x}, y) = \Pr(y|\boldsymbol{x}) \Pr(\boldsymbol{x})$. Here, we define the true conditional probability $\eta(\boldsymbol{x}) = \Pr(y = 1|\boldsymbol{x})$.

Noisy label problem. In practice, noisy label $\tilde{y}$ instead of true label y can be possibly obtained. Here, assuming noisy label $\tilde{y}$ is generated based on the true label y and a transition probability $\tau_{ij} = \Pr(\tilde{y} = j|y = i)$ that $\tau_{ij} < 0.5$ when $i \neq j$ for feasibility. We aim to find a predictor Bayes classifier $h^* = \arg\min_{h:\mathcal{X}\to\mathcal{Y}} \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}} [h(\boldsymbol{x}) \neq y]$ using the corrupted training dataset $D = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^N$. By the definition, the Bayes optimal classifier can be calculated with $\eta(\boldsymbol{x})$ as follows: $h^*(\boldsymbol{x}) = \mathbb{1}_{\{\eta(\boldsymbol{x})>1/2\}}$. Based on our assumption, the conditional probability of noisy label $\tilde{\eta}(\boldsymbol{x}) = \Pr(\tilde{y} = 1|\boldsymbol{x})$ can be defined as follows:

$$\tilde{\eta}(\boldsymbol{x}) = (1 - \tau_{10})\, \eta(\boldsymbol{x}) + \tau_{01}\, [1 - \eta(\boldsymbol{x})]$$

Hereinafter, we state the assumption and present our theorem. The first theorem establishes a connection between a classifier $f : \mathcal{X} \to \mathcal{Y}$ with the probability of a noisy label $\tilde{y}$ being correct.

Assumption 1 (Tsybakov 2004)**.** *There exist constants $W, \alpha > 0$, and $t \in (0, \frac{1}{2}]$, such that for all $t \leq t_0$,*

$$\Pr\left[\left|\eta(\boldsymbol{x}) - \frac{1}{2}\right| \leq t\right] \leq W t^{\alpha}$$

Tsybakov condition, which is also referred to as the margin assumption, indicates that the region surrounding the decision boundary, represented by $\{\boldsymbol{x} \in \mathcal{X} | \eta(\boldsymbol{x}) = 1/2\}$, has a bounded volume. This assumption is widely used in prior research Belkin et al. (2018); Qiao et al. (2019), and has been substantiated through empirical evidence in the study by Zheng et al. (2020).

Assumption 2 (Correct Label)**.** *Given $\boldsymbol{x}$, its correct label is the Bayes optimal classifier prediction $h^*(\boldsymbol{x})$.*

Our goal is to recover the correct label, $h^*(\boldsymbol{x})$, for each data point $\boldsymbol{x}$, rather than y. It is important to note that $h^*(\boldsymbol{x})$ is determined solely by the true label distribution $\eta(\boldsymbol{x})$, whereas y is just a sample from this distribution. For the trained model f, we define the estimation errors for the corrupted and true conditional probability as $\zeta_N := \|f - \tilde{\eta}\|_\infty$ and $\zeta_T := \|f - \eta\|_\infty$.

Theorem 1. *Suppose that Asssumption 1 and 2 are satisfied with constant $C, \alpha > 0$, and $t_0 \in (0, \frac{1}{2}]$. Assume $\zeta_N \leq t_0(1-\tau)$ or $\zeta_T \leq t_0$. For $\Delta = \frac{1-|\tau_{10}-\tau_{01}|}{2}$, we have:*

$$\Pr_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\tilde{y} = h^*(\boldsymbol{x}), f_{\tilde{y}}(\boldsymbol{x}) < \Delta\right] \leq W\left[\min\left(\frac{\zeta_N}{1-\tau}, \zeta_T\right)\right]^{\alpha},$$

where $\tau = \tau_{01} + \tau_{10}$ and $f_{\tilde{y}}(\boldsymbol{x})$ is the predicted probability of the label being $\tilde{y}$ by classifier f.

Theorem 1 implies that if the classifier f is sufficiently close to either true (η) or corrupted conditional distribution ($\tilde{\eta}$), then the probability that a given label $\tilde{y}$ is correct is bounded if the classifier has lower confidence in it. It implies that when we re-label the given training dataset by utilizing f, the probability of generating additional noisy labels is upper-bounded by the RHS of the theorem. Note that while the theorem stated in Zheng et al. (2020) only focused on ζ_N, we extend the theorem for the utilization of ζ_T. Through our extension, Theorem 1 can achieve the tighter upper bound by reducing one of the values of ζ_T and ζ_N.

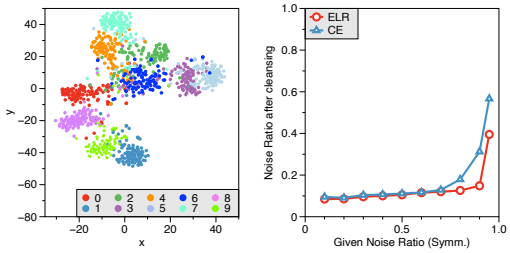
Reducing ζ_T. As stated in Theorem 1, a solution to reduce the upper bound is to make the classifier approximate the true conditional distribution. We suggest additional theoretical results that can achieve lower ζ_T with well-defined features through application of PTMs. Here, we first apply linear discriminant analysis (LDA) assumption which is widely referred in previous works Lee et al. (2019); Kim et al. (2021).

Theorem 2 (Informal)**.** *Suppose that f is a linear classifier and the LDA assumption for $\boldsymbol{x} \in \mathcal{X}$ holds. For the decision boundary $b = \frac{1}{2}\left(\frac{\sum_{i=1}^{N} \mathbb{1}_{\{\tilde{y}_i=0\}} f(\boldsymbol{x}_i)}{\sum_{i=1}^{N} \mathbb{1}_{\{\tilde{y}_i=0\}}} + \frac{\sum_{i=1}^{N} \mathbb{1}_{\{\tilde{y}_i=1\}} f(\boldsymbol{x}_i)}{\sum_{i=1}^{N} \mathbb{1}_{\{\tilde{y}_i=1\}}}\right)$, the lower bounds for the precision and recall for true conditional distribution can be derived as follows:*

$$\texttt{PRECISION} \geq \left(1 + \frac{\Phi(\mathcal{O}(-\Delta/\sigma))}{\Phi(\mathcal{O}(\Delta/\sigma))}\right)^{-1}$$
$$\texttt{RECALL} \geq \Phi\left(\mathcal{O}(\Delta/\sigma)\right),$$

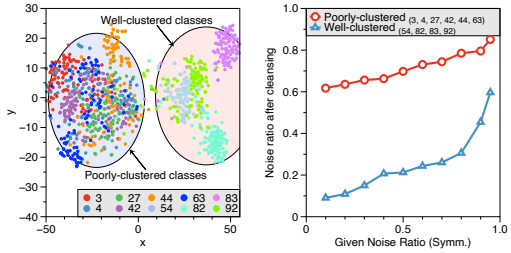
where Φ is the cumulative distribution function of $\mathcal{N}(0,1)$, Δ is function mean difference between two classes, and σ is classwise standard deviation of function values.

The formal statement and proof are in Appendix B.2. Theorem 2 demonstrates that well-clustered features have a larger difference in mean of distribution (Δ) and the smaller the variance of the cluster (σ), the larger the lower bound of recall and precision. As we have large lower bounds for precision and recall for true conditional distribution, we can approximate the true conditional distribution (η) with small estimation errors (ζ_T) by predicting most of the instances to its true labels. Because the PTMs can extract class-relevant features Radford et al. (2021), and using them improves efficiency and accuracy by leveraging well-defined feature extractors to construct a classifier that closely approximates the true conditional distribution with linear probing. This effectively corrects noisy labels.



(a) t-SNE CIFAR-10 (b) CIFAR-10 Cleansing

Fig. 5: CIFAR-10 analysis: (a) t-SNE Van der Maaten & Hinton (2008) plot and (b) noisy label cleansing performance through linear probing. It shows well-clustering and sufficient cleansing performance for all classes.



(a) t-SNE CIFAR-100 (b) CIFAR-100 Cleansing

Fig. 6: CIFAR-100 analysis: (a) t-SNE Van der Maaten & Hinton (2008) plot and (b) noisy label cleansing performance through linear probing. It shows there exists some classes are difficult to be well-clustered and cleansed.

A.2 Empirical Observations

In this section, we provide empirical observations about the theoretical evidence in Section A.1 that PTMs can help purify the given noisy labeled datasets only for the well-clustered classes.

Effectiveness of PTMs on well-clustered classes. We conduct exploratory experiments on noisy label cleansing performance using CIFAR-10 datasets with varying rates of symmetric noise for examining the results when PTMs meet label noise. To achieve this aim, we utilized ViT-B/32-CLIP model Radford et al. (2021) for PTM and trained the linear classifier on CIFAR-10 dataset similar to that of Zhu et al. (2022). As described in Figure 5, the PTM has sufficient power for clustering each class on CIFAR-10 and cleansing noisy labels. Based on this, we can also verify the philosophy of our theoretical evidence as "PTMs with linear probing can empower the cleansing performance."

When PTMs meet poorly-clustered classes. However, PTM has difficulty in learning when the data used for pre-training and linear probing are dissimilar Zhuang et al. (2020); Yosinski et al. (2014); Lee et al. (2022). To confirm this phenomenon, we experimented on the CIFAR-100 dataset with a similar setting to CIFAR-10. The only difference was that we plotted 10 randomly selected classes. As shown in Figure 6, some classes (54, 82, 83, 92; called well-clustered classes) in the CIFAR-100 dataset can create their own clusters, while others (3, 4, 27, 42, 44, 63) tend to mix without creating their own group (called poorly-clustered classes). This implies that the advantages of PTMs mentioned in Section A.1 cannot be leveraged for specific classes. This empirical observation implies that the Tsybakov assumption is difficult to be satisfied, and the lower bounds of precision and recall (Theorem 2) are difficult to be enlarged by small Δ and large σ values.

A.3 Distinguishing Well/Poorly-clustered Classes

In the previous subsection, we observed a significant difference in correction performance between well-clustered and poorly-clustered classes, due to the difficulty in clusterability. Therefore, the key factor in using PTMs for cleaning noisy labels is to find a method that maximizes the positive impact from the well-clustered classes and minimizes the negative effects from the poorly-clustered classes. Therefore, to fully leverage the potential of PTMs for cleansing noisy labels in target datasets, differentiating between "well-clustered classes" and "poorly-clustered classes" is necessary. In Proposition 1, we present our basic concept for distinguishing these two categories.

Proposition 1. *For the similar data points in the same class $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ that satisfy $\eta^*(\boldsymbol{x}_1) = \eta^*(\boldsymbol{x}_2)$ for arbitrary conditional distribution η^*, if $\|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\| > 2\zeta_N$, we can say that such classes are poorly-clustered that Theorem 1 would not hold.*

The proof can be found in Appendix B.1.2. As stated in Proposition 1, when the outputs of similar inputs in certain classes are dissimilar, linear probing of PTM is ineffective for cleansing noisy labels in those classes. Considering the aforementioned limitation of PTM in cleansing noisy labels in certain classes, we designed our algorithm to achieve both computational efficiency and a reliable performance enhancement while utilizing PTMs in the presence of label noise.

B THEORETICAL JUSTIFICATION

B.1 THEORETICAL GUARANTEES FOR MOTIVATIONS

Lemma 1 (Zheng et al. 2020)**.** *If a classifier g depends linearly on η, i.e, $g(x) = a\eta + b$ with $a, b > 0$. Set $\Delta = \min(\frac{a}{2} + b, 1 - b - \frac{a}{2})$. We have*

$$\Pr_{(x,y)\sim D}[\tilde{y} = h^*(\boldsymbol{x}), g_{\tilde{y}}(\boldsymbol{x}) < \Delta] = 0 \tag{3}$$

Proof. To calculate $\Pr_{(x,y)\sim D}[\tilde{y} = h^*(\boldsymbol{x}), g_{\tilde{y}}(\boldsymbol{x}) < \Delta]$, we enumerate two cases:

Case 1: $\tilde{y} = 1$. Observe $h^*(\boldsymbol{x}) = 1$ iff $\eta(\boldsymbol{x}) > 1/2$; $g_{\tilde{y}}(\boldsymbol{x}) = g(\boldsymbol{x}) = a\eta(\boldsymbol{x}) + b < \Delta$ iff $\eta(\boldsymbol{x}) < \frac{\Delta - b}{a}$. We have:

$$\Pr_{(x,y)\sim D}[\tilde{y} = h^*(\boldsymbol{x}), g_{\tilde{y}}(\boldsymbol{x}) < \Delta] = \Pr[\frac{1}{2} < \eta(\boldsymbol{x}) < \frac{\Delta - b}{a}].$$

We next show that this probability if 0 for the chosen $\Delta = \min(\frac{a}{2} + b, 1 - b - \frac{a}{2})$. If $\Delta = \frac{a}{2} + b$, the probability is zero as $\frac{\Delta - b}{a} = \frac{1}{2}$. Otherwise, $\Delta = 1 - b - \frac{a}{2}$. We know that $1 - b - \frac{a}{2} < \frac{a}{2} + b$. Therefore, $1 - 2b < a$. In this case,

$$\frac{\Delta - b}{a} = \frac{1 - 2b}{a} - \frac{1}{2} < 1 - \frac{1}{2} = \frac{1}{2}.$$

Thus, we have $\Pr\left[\frac{1}{2} < \eta(\boldsymbol{x}) < \frac{\Delta - b}{a}\right] = 0$.

Case 2: $\tilde{y} = 0$. Observe that $h^*(\boldsymbol{x}) = 0$ iff $\eta(\boldsymbol{x}) \leq 1/2$; $g_{\tilde{y}}(\boldsymbol{x}) = 1 - g(x) = 1 - [a\eta(\boldsymbol{x}) + b] < \Delta$ iff $\eta(\boldsymbol{x}) > L := \frac{1 - b - \Delta}{a}$, we have:

$$\Pr_{(x,y)\sim D}[\tilde{y} = h^*(\boldsymbol{x}), g_{\tilde{y}}(\boldsymbol{x}) < \Delta] = \Pr[\frac{1}{2} < \eta(\boldsymbol{x}) < \frac{\Delta - b}{a}].$$

Similar to Case 1, by checking when $\Delta = \frac{a}{2} + b$ and when $\Delta = 1 - b - \frac{a}{2}$, we can verify that $\Pr\left[\frac{1 - b - \Delta}{a} < \eta(\boldsymbol{x}) < \frac{1}{2}\right] = 0$.

This proves Equation 1 and completes the proof.

□

Lemma 2 (Zheng et al. 2020)**.** *Let $\Delta = \frac{1 - \tau_{10} - \tau_{01}}{2}$. Let $\tilde{\eta} = \tilde{\eta}$ and $\tilde{\eta}_0 = 1 - \tilde{\eta}$. Then, We have*

$$\Pr_{(x,y)\sim D}[\tilde{y} = h^*(\boldsymbol{x}), \tilde{\eta}(\boldsymbol{x}) < \Delta] = 0 \tag{4}$$

Proof. Recall $\eta(\boldsymbol{x}) = (1 - \tau_{01} - \tau_{10})\eta(\boldsymbol{x}) + \tau_{01}$, in which τ_{01} and τ_{10} are transition probabilities. We can directly prove this lemma using Lemma 1 by setting $g = \eta$ with $a = 1 - \tau_{01} - \tau_{10}$ and $b = \tau_{01}$. □

B.1.1 PROOF OF THEOREM 1

Proof. Case 1 Zheng et al. (2020): When $\tilde{y} = 1$, $f_{\tilde{y}}(\boldsymbol{x}) = f(\boldsymbol{x}) \geq \tilde{\eta}(\boldsymbol{x}) - \zeta_N$. Then, we have

$$\Pr[\tilde{y} = h^*(\boldsymbol{x}), f_{\tilde{y}}(\boldsymbol{x}) < \Delta] \leq \Pr[\tilde{y} = h^*(\boldsymbol{x}), \tilde{\eta}(\boldsymbol{x}) - \zeta_N < \Delta]$$

By substituting Δ with $\Delta + \zeta_N$ into Equation 1, we have :

$$\begin{aligned}\Pr[\tilde{y} = h^*(\boldsymbol{x}) = 1, \tilde{\eta}(\boldsymbol{x}) - \zeta_N < \Delta] &= \Pr[\tilde{y} = h^*(\boldsymbol{x}) = 1, \eta(\boldsymbol{x}) < \Delta + \zeta_N] \\ &= \Pr\left[\frac{1}{2} < \eta(\boldsymbol{x}) < \frac{\Delta + \zeta_N - \tau_{01}}{1 - \tau}\right]\end{aligned}$$

Similar to Lemma 1, by discussing the cases when $\Delta = \frac{1+\tau_{10}-\tau_{01}}{2}$ and when $\Delta = \frac{1+\tau_{01}-\tau_{10}}{2}$, we can show that $\frac{\Delta-\tau_{01}}{1-\tau} < \frac{1}{2}$. Based on the Tsybakov condition, we have

$$\Pr\left[\frac{1}{2} < \eta(\boldsymbol{x}) < \frac{\Delta-\tau_{01}}{1-\tau} + \frac{\zeta_N}{1-\tau}\right] \leq \Pr\left[\frac{1}{2} < \eta(\boldsymbol{x}) < \frac{1}{2} + \frac{\zeta_N}{1-\tau}\right] \leq C(\frac{\zeta_N}{1-\tau})^\lambda$$

This implies that:

$$\Pr\left[\tilde{y} = h^*(\boldsymbol{x}) = 1, f_{\tilde{y}}(\boldsymbol{x}) < \Delta\right] \leq C(\frac{\zeta_N}{1-\tau})^\lambda$$

Similar to case 1 of Lemma 1, by using Equation 4 for the case when $\tilde{y} = 0$, we can prove that

$$\begin{aligned}\Pr\left[\tilde{y} = h^*(\boldsymbol{x}) = 0, f_{\tilde{y}}(\boldsymbol{x}) < \Delta\right] &\leq \Pr\left[\tilde{y} = h^*(\boldsymbol{x}) = 0, 1-\tilde{\eta}(\boldsymbol{x}) - \zeta_N < \Delta\right] \\ &= \Pr\left[\frac{1-\tau_{01}-\Delta}{1-\tau} - \frac{\zeta_N}{1-\tau} < \eta(\boldsymbol{x}) < \frac{1}{2}\right] \\ &\leq \Pr\left[\frac{1}{2} - \frac{\zeta_N}{1-\tau} < \eta(\boldsymbol{x}) < \frac{1}{2}\right] \leq C(\frac{\zeta_N}{1-\tau})^\lambda\end{aligned}$$

Case 2: When $\tilde{y} = 1$, $f_{\tilde{y}}(\boldsymbol{x}) = f(\boldsymbol{x}) \geq \eta(\boldsymbol{x}) - \zeta_C$. Then, we have

$$\Pr\left[\tilde{y} = h^*(\boldsymbol{x}), f_{\tilde{y}}(\boldsymbol{x}) < \Delta\right] \leq \Pr\left[\tilde{y} = h^*(\boldsymbol{x}), \eta(\boldsymbol{x}) - \zeta_C < \Delta\right] = \Pr\left[\frac{1}{2} < \eta(\boldsymbol{x}) < \Delta + \zeta_C\right]$$

Since $\Delta = \frac{1-|\tau_{10}-\tau_{01}|}{2} < \frac{1}{2}$, we have

$$\Pr\left[\frac{1}{2} < \eta(\boldsymbol{x}) < \Delta + \zeta_C\right] = \Pr\left[\frac{1}{2} < \eta(\boldsymbol{x}) < \frac{1}{2} + \zeta_C\right] \leq C(\zeta_C)^\lambda$$

For the case when $\tilde{y} = 0$, we have

$$\begin{aligned}\Pr\left[\tilde{y} = h^*(\boldsymbol{x}) = 0, f_{\tilde{y}}(\boldsymbol{x}) < \Delta\right] &\leq \Pr\left[\tilde{y} = h^*(\boldsymbol{x}) = 0, 1-\eta(\boldsymbol{x}) - \zeta_C < \Delta\right] \\ &= \Pr\left[\frac{1}{2} - \zeta_C < \eta(\boldsymbol{x}) < \frac{1}{2}\right] \leq C(\zeta_C)^\lambda\end{aligned}$$

Since both case 1 and case 2 holds, we have

$$\Pr_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\tilde{y} = h^*(\boldsymbol{x}), f_{\tilde{y}}(\boldsymbol{x}) < \Delta\right] \leq C\left[\min(\frac{\zeta_N}{1-\tau}, \zeta_C)\right]^\lambda$$

□

B.1.2 Proof of Proposition 1

Proof. When assumption for Theorem 1 holds, *i.e.,* $\eta(\boldsymbol{x})$ satisfies the Tsybakov condition with constants $C, \lambda > 0$, $t_0 \in (0, \frac{1}{2}]$ and $\zeta_N \leq t_0(1-\tau_{10}-\tau_{01})$ and $\zeta_C \leq t_0$. Then, we have:

$$\begin{aligned}\|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\| &= \|(f(\boldsymbol{x}_1) - \tilde{\eta}(\boldsymbol{x}_1)) - (f(\boldsymbol{x}_2) - \tilde{\eta}(\boldsymbol{x}_2)) + (\tilde{\eta}(\boldsymbol{x}_1) - \tilde{\eta}(\boldsymbol{x}_2))\| \\ &\leq \|(f(\boldsymbol{x}_1) - \tilde{\eta}(\boldsymbol{x}_1))\| + \|(f(\boldsymbol{x}_2) - \tilde{\eta}(\boldsymbol{x}_2))\| + \|\tilde{\eta}(\boldsymbol{x}_1) - \tilde{\eta}(\boldsymbol{x}_2)\| \\ &\leq \zeta_N + \zeta_N + \|\tilde{\eta}(\boldsymbol{x}_1) - \tilde{\eta}(\boldsymbol{x}_2)\|\end{aligned}$$

With $\|\tilde{\eta}(\boldsymbol{x}_1) - \tilde{\eta}(\boldsymbol{x}_2)\| = 0$ by the definition of similar data points, we have $\|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\| \leq 2\zeta_N$. Hence, with proof by contrapositive, when $\|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\| > 2\zeta_N$, the assumption for Theorem 1 does not hold. This implies that Theorem 1 does not valid if such classes do not pass the C3 module. □

B.2 PROOF OF THEOREM 2

Since the feature distribution comprises two Gaussian distributions, the projected distribution is also a mixture of two Gaussian distributions. By LDA assumption, its decision boundary with probability 0.5 is the same as the average of the means of two clusters. Here, we introduce our additional theoretical motivation that corrects noisy labels by simple linear probing. Here, we introduce the formal statement and proof for Theorem 2.

Theorem 3 (Formal version of Theorem 2)**.** *Suppose that f is a linear classifier and $\Pr(y=0) = \Pr(y=1)$. Furthermore, we assume that the output distribution is comprised of two Gaussian distributions with mean μ_0 and μ_1 for class $y=0$ and $y=1$, respectively, common standard deviation σ by LDA assumption. For the decision boundary $b = \frac{1}{2}\left(\frac{\sum_{i=1}^{N} \mathbb{1}_{\{\tilde{y}_i=0\}} f(\boldsymbol{x}_i)}{\sum_{i=1}^{N} \mathbb{1}_{\{\tilde{y}_i=0\}}} + \frac{\sum_{i=1}^{N} \mathbb{1}_{\{\tilde{y}_i=1\}} f(\boldsymbol{x}_i)}{\sum_{i=1}^{N} \mathbb{1}_{\{\tilde{y}_i=1\}}}\right)$, with the probability of $1-\delta$, the lower bounds for the precision and recall for true conditional distribution can be derived as follows:*

$$\texttt{RECALL} \geq \Phi\left(\frac{\Delta - 2\mathcal{C}\sqrt{\left(\frac{1}{N_0}+\frac{1}{N_1}\right)\log\left(\frac{2}{\delta}\right)}}{2\sigma}\right)$$

$$\texttt{PRECISION} \geq \left(1 + \Phi\left(\frac{-\Delta - 2\mathcal{C}\sqrt{\left(\frac{1}{N_0}+\frac{1}{N_1}\right)\log(2/\delta)}}{2\sigma}\right) \Bigg/ \Phi\left(\frac{\Delta - 2\mathcal{C}\sqrt{\left(\frac{1}{N_0}+\frac{1}{N_1}\right)\log(2/\delta)}}{2\sigma}\right)\right)^{-1}$$

Proof. Here, we define $N_0 = \sum_{i=1}^{N} \mathbb{1}_{\{\tilde{y}_i=0\}}$ and $N_1 = \sum_{i=1}^{N} \mathbb{1}_{\{\tilde{y}_i=1\}}$. We also define the mean of $f(\boldsymbol{x})$ for $y=0$ and $y=1$ as μ_0 and μ_1, mean of corrupted conditional distribution for $\tilde{y}=0$ and $\tilde{y}=1$ as $\tilde{\mu}_0$ and $\tilde{\mu}_1$. By the central limit theorem (CLT), we have $\frac{\sum_{i=1}^{N} \mathbb{1}_{\{\tilde{y}_i=0\}} f(\boldsymbol{x}_i)}{N_0} \sim \mathcal{N}\left(\tilde{\mu}_0, \frac{\tilde{\sigma}_0^2}{N_0}\right)$ and $\frac{\sum_{i=1}^{N} \mathbb{1}_{\{\tilde{y}_i=1\}} f(\boldsymbol{x}_i)}{N_1} \sim \mathcal{N}\left(\tilde{\mu}_1, \frac{\tilde{\sigma}_1^2}{N_1}\right)$ where $\tilde{\sigma}_0^2$ and $\tilde{\sigma}_1^2$ are standard deviation for the corrupted distribution of given label 0 and 1. Thus, we have $\frac{\sum_{i=1}^{N} \mathbb{1}_{\{\tilde{y}_i=0\}} f(\boldsymbol{x}_i)}{N_0} + \frac{\sum_{i=1}^{N} \mathbb{1}_{\{\tilde{y}_i=1\}} f(\boldsymbol{x}_i)}{N_1} \sim \mathcal{N}\left(\tilde{\mu}_0 + \tilde{\mu}_1, \frac{\tilde{\sigma}_0^2}{N_0} + \frac{\tilde{\sigma}_1^2}{N_1}\right)$. Moreover, by the definition of transition probability $\tau_{ij} = \Pr(\tilde{y}=j|y=i)$, we have $\tilde{\mu}_0 = \tau_{00}\mu_0 + \tau_{10}\mu_1$ and $\tilde{\mu}_1 = \tau_{01}\mu_0 + \tau_{11}\mu_1$. Thus, we have

$$\tilde{\mu}_0 + \tilde{\mu}_1 = (\tau_{00}\mu_0 + \tau_{10}\mu_1) + (\tau_{01}\mu_0 + \tau_{11}\mu_1) = (\tau_{00}+\tau_{01})\mu_0 + (\tau_{10}+\tau_{11})\mu_1 = \mu_0 + \mu_1$$

By the concentration inequality on standard Gaussian distribution, we have

$$\Pr\left(\left|\frac{\sum_{i=1}^{N} \mathbb{1}_{\{\tilde{y}_i=0\}} f(\boldsymbol{x}_i)}{N_0} + \frac{\sum_{i=1}^{N} \mathbb{1}_{\{\tilde{y}_i=1\}} f(\boldsymbol{x}_i)}{N_1} - (\mu_0+\mu_1)\right| > \psi\right) < 2\exp\left(-\frac{\psi^2}{2}\cdot\frac{1}{\tilde{\sigma}_0^2/N_0 + \tilde{\sigma}_1^2/N_1}\right) \tag{5}$$

Therefore, with probability $1-\delta$

$$\frac{\mu_0+\mu_1}{2} - \mathcal{C}\sqrt{\left(\frac{1}{N_0}+\frac{1}{N_1}\right)\log\left(\frac{2}{\delta}\right)} \leq b \leq \frac{\mu_0+\mu_1}{2} + \mathcal{C}\sqrt{\left(\frac{1}{N_0}+\frac{1}{N_1}\right)\log\left(\frac{2}{\delta}\right)}, \tag{6}$$

where $\mathcal{C} > 0$ is a constant. Then, by using the Equation 6, we can derive the lower bound for the recall as follows:

$$
\begin{aligned}
\texttt{RECALL} = \Pr(f(\boldsymbol{x}) > b | y = 1) &\geq \Pr\left(f(\boldsymbol{x}) > \frac{\mu_0 + \mu_1}{2} + \mathcal{C}\sqrt{\left(\frac{1}{N_0} + \frac{1}{N_1}\right)\log\left(\frac{2}{\delta}\right)} \middle| y = 1 \right) \\
&= \Pr\left(\frac{f(\boldsymbol{x}) - \mu_1}{\sigma} > \frac{-\Delta + 2\mathcal{C}\sqrt{\left(\frac{1}{N_0} + \frac{1}{N_1}\right)\log\left(\frac{2}{\delta}\right)}}{2\sigma} \right) \\
&= \Pr\left(\mathcal{N}(0,1) > \frac{-\Delta + 2\mathcal{C}\sqrt{\left(\frac{1}{N_0} + \frac{1}{N_1}\right)\log\left(\frac{2}{\delta}\right)}}{2\sigma} \right) \\
&= \Phi\left(\frac{\Delta - 2\mathcal{C}\sqrt{\left(\frac{1}{N_0} + \frac{1}{N_1}\right)\log\left(\frac{2}{\delta}\right)}}{2\sigma} \right)
\end{aligned}
$$

For precision, with the assumption of the balanced dataset (*i.e.,* $\Pr(y = 0) = \Pr(y = 1)$), we have a lower bound for precision as follows:

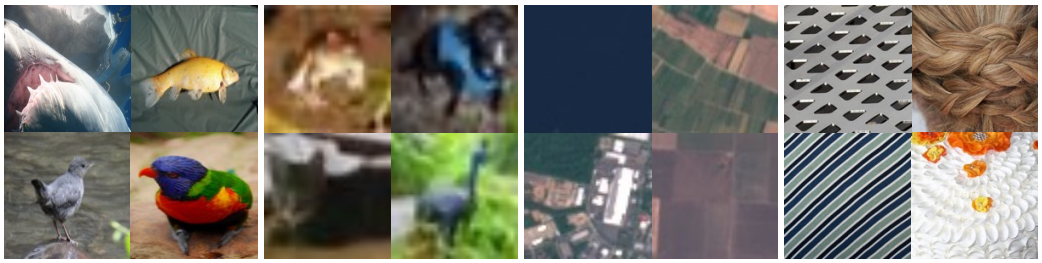
$$
\begin{aligned}
\texttt{PRECISION} &= \Pr(y = 1 | f(\boldsymbol{x}) > b) \\
&= \frac{\Pr(f(\boldsymbol{x}) > b | y = 1) P(y = 1)}{\sum_{i \in \{0,1\}} \mathbb{P}(f(\boldsymbol{x}) > b | y = i) P(y = i)} \\
&\geq \frac{\Pr\left(f(\boldsymbol{x}) > \frac{\mu_1 + \mu_0}{2} + \mathcal{C}\sqrt{\left(\frac{1}{N_0} + \frac{1}{N_1}\right)\log\left(\frac{2}{\delta}\right)} \middle| y = 1 \right) \Pr(y = 1)}{\sum_{i \in \{0,1\}} \Pr\left(f(\boldsymbol{x}) > \frac{\mu_0 + \mu_1}{2} - \mathcal{C}\sqrt{\left(\frac{1}{N_0} + \frac{1}{N_1}\right)\log\left(\frac{2}{\delta}\right)} \middle| y = i \right) \Pr(y = i)} \\
&= \frac{\Pr\left(f(\boldsymbol{x}) > \frac{\mu_0 + \mu_1}{2} + \mathcal{C}\sqrt{\left(\frac{1}{N_0} + \frac{1}{N_1}\right)\log\left(\frac{2}{\delta}\right)} \middle| y = 1 \right) \Pr(y = 1)}{\sum_{i \in \{0,1\}} \Pr\left(f(\boldsymbol{x}) > \frac{\mu_0 + \mu_1}{2} - \mathcal{C}\sqrt{\left(\frac{1}{N_0} + \frac{1}{N_1}\right)\log\left(\frac{2}{\delta}\right)} \middle| y = i \right) \Pr(y = i)} \\
&\geq \frac{1}{1 + \frac{\Pr\left(f(\boldsymbol{x}) > \frac{\mu_0 + \mu_1}{2} + \mathcal{C}\sqrt{\left(\frac{1}{N_0} + \frac{1}{N_1}\right)\log(2/\delta)} \middle| y = 0 \right) \Pr(y = 0)}{\Pr\left(f(\boldsymbol{x}) > \frac{\mu_0 + \mu_1}{2} - \mathcal{C}\sqrt{\left(\frac{1}{N_0} + \frac{1}{N_1}\right)\log(2/\delta)} \middle| y = 1 \right) \Pr(y = 1)}} \\
&= \frac{1}{1 + \Phi\left(\frac{-\Delta - 2\mathcal{C}\sqrt{\left(\frac{1}{N_0} + \frac{1}{N_1}\right)\log(2/\delta)}}{2\sigma} \right) \Big/ \Phi\left(\frac{\Delta - 2\mathcal{C}\sqrt{\left(\frac{1}{N_0} + \frac{1}{N_1}\right)\log(2/\delta)}}{2\sigma} \right)}
\end{aligned}
$$

□

The use of well-constructed feature clusters, characterized by large mean differences and small standard deviations, can significantly improve the cleansing performance of noisy labels. This theorem is further validated by the successful application of PTMs to obtaining such clusters.

C Implementation Details

In this section, we describe the datasets and statistics that we used. We evaluate a total of six datasets (CIFAR-10/100, EuroSAT, DTD, Oxford-IIIT PET, Clothing 1M, and Webvision) with various noisy label configurations.



(a) ImageNet (b) CIFAR (c) EuroSAT (d) DTD

(e) Oxford-IIIT Pet (f) Clothing1M (g) WebVision

Fig. 7: Example Images for each dataset.

C.1 Dataset Description

CIFAR-10/100 Krizhevsky et al. (2009). The CIFAR-10 dataset contains 60000 32x32 color images divided into 10 classes, with 50000 for training and 10000 for testing. The CIFAR-100 dataset is similar but has 100 classes and is grouped into 20 superclasses, with 500 images for training and 100 for testing per class. Each image in the CIFAR-100 dataset has both a "fine" label and a "coarse" label. Examples are depicted in Figure 7b.

EuroSAT Helber et al. (2019). The EuroSAT dataset is based on Sentinel-2 satellite images, and it includes $27,000$ covering 13 spectral bands. It has 10 classes with $27,000$ images. We randomly divided the $27,000$ in samples into two groups, *i.e.,* `train:valid = 0.9:0.1`. Examples are depicted in Figure 7c.

DTD Cimpoi et al. (2014). The Describable Texture Dataset (DTD) is a collection of textural images in the wild. It contains $5,640$ images in 47 categories. Each class has 120 images. As frequently set, we divide the per-class 120 images into train/valid/test equally, *i.e.,* 40 for each split. Examples are depicted in Figure 7d.

Oxford-IIIT Pet Parkhi et al. (2012). The Oxford-IIIT pet dataset is a 37 of classes pet image dataset with roughly 200 images for each class. The images have large variations in scale, pose, and lighting. Every image has a breed-specific ground truth annotation. We train with $3,680$ images and test with the remaining $3,669$ images. Examples are depicted in Figure 7e.

Clothing 1M Xiao et al. (2015). Clothing 1M contains 1M clothing images in 14 classes and an estimated noise level of 38.5% Song et al. (2022). It is a dataset with noisy labels since the data is collected from several online shopping websites and includes many mislabeled samples. This dataset also contains 50k, 14k, and 10k images with clean labels for training, validation, and testing, respectively. Examples are depicted in Figure 7f.

WebVision Li et al. (2017). The WebVision dataset is designed to facilitate research on learning visual representation from noisy web data. It is a large-scale web images dataset that contains more than 2.4 million images crawled from the Flickr website and Google Images search. Examples are depicted in Figure 7g.

C.2 DATA PREPROCESSING

C.2.1 NOISY LABEL GENERATION FOR SYNTHETIC DATASETS

We synthetically generate symmetric and asymmetric noise by following the method proposed by Liu et al. (2020). To create symmetric noise, we randomly select a fraction of labels and alter them with uniform randomness. For asymmetric noise, we use a mapping technique on the CIFAR-10 dataset, where we change certain classes to similar but distinct classes (*i.e.,* TRUCK $\rightarrow$ AUTOMOBILE, BIRD $\rightarrow$ AIRPLANE, DEER $\rightarrow$ HORSE, CAT $\rightarrow$ DOG). On the CIFAR-100 dataset, we divide the dataset into 20 super-classes of size five and alter each class to the next class within the same super-class. In addition, we generate instance-based noise using the method proposed by Cheng et al. (2021) and compare its performance to that of existing label noise learning methods. The noise rate (global flipping rate) is defined as ϵ and we sample flip rates from a truncated normal distribution with a mean of ϵ, standard deviation as 0.1, and a range of 0 to 1. Instance-dependent label noise parameters W_I are sampled from the standard normal distribution, with the size of W_I being $L \times C$, where L is the length of each feature and C is the number of classes.

C.2.2 DATA AUGMENTATION

All image preprocessing is done using the officially released preprocessor that comes with the corresponding PTMs (ViT, CLIP, and ConvNeXt). In addition, we perform image augmentation to generate similar images in order to implement instance-centric consistency. The additional image augmentation for each dataset is as follows: Random Horizontal Flip $\rightarrow$ Random Affine (CIFAR-10/100) and Random Resized Crop $\rightarrow$ Random Horizontal Flip $\rightarrow$ Color Jitter (EuroSAT, DTD, Oxford-IIIT Pet, Clothing1M, WebVision).

C.3 DETAILED IMPLEMENTATION

Here, we describe the implementation of LNL methods after purifying the noisy labels through our proposed `EPL`.

CIFAR-10/100 for LNL methods For data augmentation, we follow the default settings which are widely used in various existing works Liu et al. (2020); Li et al. (2020). For CIFAR-10/100, each side of the image is padded with 4 pixels, and a 32×32 crop is randomly selected from the padded image or its horizontal flip. We finally normalize the image with the following means and standard deviations, sequentially: CIFAR-10 $\{\texttt{mean} : (0.4913, 0.4821, 0.4465), \texttt{standard deviation} : (0.2470, 0.2434, 0.2615)\}$

To implement DivideMix Li et al. (2020), ELR+ Liu et al. (2020), and UNICON Karim et al. (2022), we use PreAct ResNet18 He et al. (2016). For a fair comparison, we use the same values as their reported values for all hyper-parameter of ELR+. For DivideMix and UNICON, we use the same values as their reported hyper-parameter values except the λ_u which is unsupervised loss function weights for corresponding baseline methods. This is because the optimal values for λ_u are largely varying the fraction of noisy labels and ours significantly reduce the fraction of noisy labels. Under a small fraction of noisy labels, we have to reduce the value of λ_u.

Additional synthetic datasets for LNL methods. We use RandomResizeCrop under resize parameter 224 and cropping scale $(0.08, 1.0)$ with BICUBIC interpolation method, random horizontal flip, and color jitter with brightness 0.4 contrast 0.4 and saturation 0.4 parameters. We normalize each image with $\{\texttt{mean} : (0.485, 0.456, 0.406), \texttt{standard deviation} : (0.229, 0.224, 0.225)\}$ which is typically used for 224 size images, comes from ImageNet training recipe.

We utilize ResNet18 He et al. (2016) for three additional synthetic datasets. More precisely, we utilize a pre-trained model, offered by `Torchvision` for the DTD and Pet datasets and train from scratch for the EuroSAT dataset. Detail hyperparameters are described in Table 4. For LNL model-specific hyperparameters, such as β and λ for ELR+, we utilize the original hyperparameters for Clothing1M experiments.

Clothing1M. As following the previous works Li et al. (2020); Liu et al. (2020), we preprocess the raw images as following augmentation procedure: Resize with $256 \rightarrow$ Random Crop with parameter $224 \rightarrow$ Random Horizontal Flip. We normalize each image with $\{\texttt{mean} :$

Table 4: Hyperparameters for EuroSAT, DTD and Oxford-IIIT Pet datasets to run ELR+, DivideMix and UNICON.

	EuroSAT	DTD	Pet
Model	ResNet18 (Scratch)	ResNet18 (ImageNet Pre-trained)	
Optimizer	SGD optimizer		
Learning rate	0.01		
Momentum	0.9		
Weight Decay	1×10^{-4}		
Scheduler	0.1 decay at 25 epoch	Cosine Annealing Scheduler (single step)	
Epochs	50	15	
Batch size	128	64	

$(0.6959, 0.6537, 0.6371)$, `standard deviation` : $(0.3113, 0.3192, 0.3214)\}$. To implement the baselines, we apply ResNet50 He et al. (2016) as backbone networks for all baseline methods. For a fair comparison, we use the same values as their reported values for all hyper-parameter of each baseline method. For a fair comparison, we use the same values as their reported values for all hyper-parameter of each baseline method.

WebVision. As following the previous works Li et al. (2020); Liu et al. (2020), we preprocess the raw images as following augmentation procedure: Random Crop with $227 \rightarrow$ Random Horizontal Flip. We normalize each image with $\{$`mean` : $(0.485, 0.456, 0.406)$, `standard deviation` : $(0.229, 0.224, 0.225)\}$ which is typically used for ImageNet datasets. Here, we apply InceptionResNetV2 Szegedy et al. (2017) for backbone networks. For a fair comparison, we use the same values as their reported values for all hyper-parameter of each baseline method.

C.4 Implementation Trick

To reduce the computational complexity of extracting features by using PTMs, we implement small tricks. It is based on the fact that the only part being updated at the linear probing phase is the linear classifier. Therefore, we initially extract all features before training the linear classifier and just train the linear classifier (which consists of one fully connected layer) at the linear probing phase. It can dramatically reduce the `EPL` overhead.

C.5 Pre-trained Model Description

In this section, we introduce the PTMs that we mainly use to implement our proposed method, `EPL`. All pre-trained weights are officially released in HuggingFace Wolf et al. (2020) and we are easily accessible to these models.

Vision Transformer (ViT; Dosovitskiy et al. 2021). The Vision Transformer utilizes a Transformer-inspired architecture to classify images by dividing them into fixed-size patches, linear embedding each patch, and incorporating position embeddings. These resulting vectors are then processed by a standard Transformer encoder and a learnable "classification token" is added to the sequence for classification. We mainly use ViT-L/16 which is pre-trained on ImageNet-22K, and ViT-B/16 and ViT-B/32 which are pre-trained on ImageNet-1K. Examples of ImageNet dataset are depicted in Figure 7a.

Contrastive Language-Image Pre-training (CLIP; Radford et al. 2021). CLIP is a highly efficient method for learning image representations through natural language supervision. It employs a simplified version of ConVIRT Zhang et al. (2020), and utilizes a joint training approach for both an image encoder and a text encoder, with the goal of correctly predicting the pairing of a batch of (image, text) training examples. In our experiment, we utilize ViT-CLIP which applies CLIP pretraining method to ViT architecture.

ConvNeXt Liu et al. (2022b). ConvNeXt is a transformation of standard ResNet into the design of a Vision Transformer through the gradual process of modernization by uncovering several crucial components that contribute to the performance difference. This exploration resulted in a family of

pure ConvNet models, known as ConvNeXt. These models are entirely built from standard ConvNet modules and have been found to be competitive in terms of accuracy and scalability when compared to Transformers. We mainly use ConvNeXt-XL which is pre-trained on ImageNet-22K. Examples of ImageNet dataset are depicted in Figure 7a.

D RELATED WORK

Learning with Noisy Labels. After Zhang et al. (2017) showed that DNNs easily memorize randomly labeled training data, numerous studies have addressed the memorization problem under label noise. Existing methods mainly address this problem by (1) detecting corrupted instances and only using label information of clean examples Han et al. (2018); Li et al. (2020); Cheng et al. (2021); Kim et al. (2021); Xia et al. (2022) (2) designing loss functions or regularization terms with robust behaviors and provable tolerance to label noise Zhang & Sabuncu (2018); Wang et al. (2019); Liu et al. (2020); Zhou et al. (2021); Ko et al. (2022). Recently, majority of the research Zheltonozhskii et al. (2022); Karim et al. (2022); Li et al. (2022) is focused on applying self-supervised approaches to construct robust feature extractors on label noise. Zheltonozhskii et al. (2022) proposed C2D to run semi-supervised approaches Li et al. (2020); Liu et al. (2020) with the initial parameters from the SimCLR Chen et al. (2020) and showed significant performances. However, these approaches may be over-complicated requiring hyperparameter tuning for different datasets, as well as significant computation resources.

Pre-trained Models. Recently, several studies have demonstrated that PTMs, which are trained on the large image Ridnik et al. (2021) or text corpora, can learn universal visual or language representations that are useful for downstream computer vision or natural language processing tasks. This has eliminated the need to train a new model from scratch. With the advancement of computational power and development of deep models such as GPT-3 Brown et al. (2020), Vision Transformer Dosovitskiy et al. (2021), and ConvNext Liu et al. (2022b), the capabilities of PTMs have greatly improved. Utilizing PTMs has been considered as an effective solution for multi-modal models such as CLIP Radford et al. (2021) and Data2Vec Baevski et al. (2022), which can effectively represent various types of domains.

As researchers make pre-trained weights of PTMs available to the open-source community, there is growing interest in finding ways to effectively use these pre-trained weights. For example, a lot of recent research is being focused on utilizing large pre-trained models for prompt learning Zhang et al. (2022) or in-context learning Liu et al. (2022a), with the goal of achieving good results in few-shot learning scenarios. However, relatively only a few studies have explored using these PTMs in a robust learning framework to handle label noise. Recently, Zhu et al. (2022) suggested the applying self-supervised PTMs Chen et al. (2020) or large PTM Radford et al. (2021) for detecting corrupted labels without training, however, these methods applied the KNN technique, which requires heavy computational consumption Li et al. (2022).