Why Can Accurate Models Be Learned from Inaccurate Annotations?

Chongjie Si[1], Yidan Cui[1], Fuchao Yang[2], and Wei Shen[1]✉

[1] MoE Key Lab of Artificial Intelligence, School of Computer Science, Shanghai Jiao Tong University
[2] Nanyang Technological University
{chongjiesi, wei.shen}@sjtu.edu.cn

Abstract. Learning from inaccurate annotations has gained significant attention due to the high cost of precise labeling. However, despite the presence of erroneous labels, models trained on noisy data often retain the ability to make accurate predictions. This intriguing phenomenon raises a fundamental yet largely unexplored question: why models can still extract correct label information from inaccurate annotations remains unexplored. In this paper, we conduct a comprehensive investigation into this issue. By analyzing weight matrices from both empirical and theoretical perspectives, we find that label inaccuracy primarily accumulates noise in lower singular components and subtly perturbs the principal subspace. Within a certain range, the principal subspaces of weights trained on inaccurate labels remain largely aligned with those learned from clean labels, preserving essential task-relevant information. We formally prove that the angles of principal subspaces exhibit minimal deviation under moderate label inaccuracy, explaining why models can still generalize effectively. Building on these insights, we propose LIP, a lightweight plug-in designed to help classifiers retain principal subspace information while mitigating noise induced by label inaccuracy. Extensive experiments on tasks with various inaccuracy conditions demonstrate that LIP consistently enhances the performance of existing algorithms. We hope our findings can offer valuable insights to understand of model robustness under inaccurate supervision.

Keywords: Inaccurate annotation, noisy label learning, partial label learning

1 Introduction

Inaccurate annotations are prevalent in real-world scenarios, arising from the prohibitive costs and challenges associated with obtaining perfectly labeled data [20, 40, 57, 58]. Despite the inherent inaccuracies in annotations, an intriguing phenomenon emerges: models trained on such noisy data often retain the ability to make correct predictions, sometimes even approaching those trained with ground-truth labels in certain cases. This paradox raises a fundamental question

that remains unresolved: **Why can models extract correct label information from inaccurate annotations?**

In this paper, we conduct a comprehensive investigation into this issue. Generally, models encapsulate the knowledge learned during training within their weight matrices. We first investigate how varying degrees of label inaccuracy affect the weight matrix, both from the perspective of singular values and subspace structure. Our findings show that large label inaccuracy accumulates noise in the lower singular components and also alters the structure of the principal subspace, leading to the loss of critical task-specific information. However, **within a certain range, label inaccuracy does not significantly affect, or may even leave unchanged, the principal subspace of the weight matrix**. Subsequently, we provide a theoretical proof to explain the aforementioned phenomenon. We find that within a certain range of label inaccuracy, the angles of the principal subspaces do not exhibit great variation. This ultimately leads us to conclude that models can learn correct information from inaccurate labels because **the principal subspaces of the weights trained on inaccurate labels do not significantly deviate from those derived from accurate labels.**

Furthermore, based on our findings, we propose a plug-in named LIP, as "Label Inaccuracy Processor". It aids existing classifiers in retaining information of principal subspaces while reducing the noises caused by label inaccuracy. This plug-in is lightweight and has been demonstrated through extensive experiments under various configurations of label inaccuracy to effectively enhance the performance of existing algorithms. The main contributions are as follows:

- To the best of our knowledge, it is the first investigation into why model can extract correct label information from inaccurate annotations. We provide both empirical and theoretical proof on this question, which may guide the future exploration of the related areas.
- We propose a lightweight but effective plug-in LIP, which helps existing classifiers retain core information and filter the noises caused by label inaccuracy.
- We conducted extensive evaluations across multiple algorithms on tasks with different inaccurate label settings, demonstrating the clear superiority of our approach.

2 Related Work

2.1 Noisy Label Learning

In real-world applications, perfectly labeled data is often unattainable due to annotation errors, leading to the challenge of noisy label learning (NLL). Deep models tend to overfit noisy labels, severely degrading generalization performance. To address this, researchers have developed various strategies, including noise-robust training and label noise modeling. Noise-Robust Training methods enhance model resilience by designing robust loss functions [9, 28, 46], such as symmetric cross-entropy [46], or by dynamically reweighting samples based on

model confidence [17, 35]. Early stopping [2] prevents deep models from memorizing label noise, while semi-supervised approaches [3, 29] leverage pseudo-labeling to refine noisy samples. Label Noise Modeling explicitly estimates noise transition matrices [10, 33] to correct corrupted labels, while meta-learning approaches [24, 37] utilize clean validation sets to optimize noise handling dynamically. Self-supervised strategies such as DivideMix [23] and Co-teaching [12] separate clean and noisy samples, improving training stability. Despite these advances, why models can learn from inaccurate annotations remains mystery.

2.2 Partial Label Learning

Partial Label Learning (PLL) is a weakly supervised learning framework where each instance is associated with multiple candidate labels, with only one being the ground truth. The key challenge in PLL is disambiguation [7, 8, 30], which can be broadly categorized into averaging-based and identification-based methods. Averaging-based methods [5, 19] assume equal contributions from all candidate labels, averaging their outputs for prediction. For instance, PL-KNN [19] estimates ground truth by leveraging neighboring instances' label confidence but suffers from susceptibility to false positives. In contrast, identification-based methods [8, 44] treat the ground truth as a latent variable, refining its estimation via Expectation-Maximization (EM). PL-AGGD [44] employs feature manifold structures for label confidence estimation, while PL-CL utilizes a complementary classifier to assist disambiguation. Recent deep learning approaches have further advanced PLL [15, 26, 27, 49, 51]. PICO [45] resolves label ambiguity via contrastive learning with embedding prototypes, while PRODEN [26] jointly updates the model and refines label identification. Additionally, [15] employs semantic label representations with weighted calibration rank loss to improve disambiguation.

3 Extracting Accurate Labels from Inaccurate Annotations: Empirical Evidence

As stated in the introduction, it remains unclear why models can learn correct labels from inaccurate annotations. In other words, we lack a fundamental understanding of what enables models to possess this capability. In this section, we will systematically explore this issue, beginning with empirical evidence. We first establish some foundational concepts. Generally, models store the critical information for classification tasks captured from training data in their final classification weights. For stand-alone methods, these weights are typically obtained by solving an optimization problem. For deep-learning-based methods, these weights correspond to the final fully connected (FC) layer used for classification. We denote the weights learned from clean labels as $\mathbf{W}$, and those learned from inaccurate labels as $\mathbf{W}'$.

To investigate this problem, it is crucial to understand **how label inaccuracy impacts the weight matrix**. Therefore, we believe that an exploration

of the relationship between $\mathbf{W}$ and $\mathbf{W}'$ may provide valuable insights into this process. Given that $\mathbf{W}'$, learned from inaccurate labels, also supports classification tasks and often achieves accuracy close to that of $\mathbf{W}$ [26, 38, 45, 52], it suggests an inherent connection between them. What, then, is the nature of this relationship?

3.1 Observations on the Singular Values

We first train a well-established classifier, PRODEN [26], which utilizes a ResNet-34 [14] backbone, on the clean CIFAR-100 dataset [22]. This process yields the classification weights $\mathbf{W}$ of the FC layer. To introduce varying levels of label inaccuracy, we randomly flip the labels of each sample with a predefined probability p, generating datasets with different degrees of annotation noise. This procedure results in two possible scenarios: a sample may acquire additional incorrect labels alongside its true label, or it may entirely lose its correct label. We also enforce a constraint that each sample retains at least one label. We set p to 0.05, 0.1, 0.2, 0.3, and 0.4, and train the same model on these noisy datasets, yielding the corresponding classification weights $\mathbf{W}'$ for each level of label inaccuracy.

To begin with, we perform Singular Value Decomposition (SVD) on both matrices $\mathbf{W}$ and $\mathbf{W}'$ to extract their corresponding singular values $\mathbf{\Sigma}$ and $\mathbf{\Sigma}$' and right singular unitary matrices, referred to as $\mathbf{V}$ and $\mathbf{V}'$. We then analyze the variation in singular values of different weights as inaccuracy increased, and the results are shown in the Fig. 1. First, we observe that the top singular values

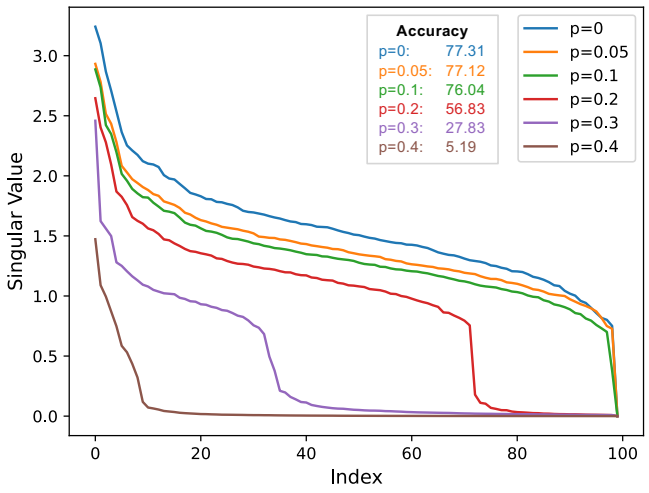

Fig. 1: Singular values under different label inaccuracy (i.e., p). The x-axis denotes the index of singular values. We also provide the corresponding classification accuracy of each setting.

of the weight matrix do not show significant differences within a certain range of label inaccuracy. However, once the label inaccuracy exceeds a certain threshold, the top singular values decline rapidly. This indicates a significant reduction in information accumulation along the principal directions of the weight matrix, suggesting that the model's ability to capture key features is diminished. As noise and uncertainty in the labels increase, these disturbances disrupt the primary structure of the weight matrix, causing the singular value spectrum to flatten. This flattening also reflects a decrease in the model's effective rank, weakening the information retained along the principal directions.

Moreover, increasing label inaccuracy accelerates the decay of singular values, as noise predominantly corrupts lower-rank directions, inhibiting meaningful feature accumulation. When inaccuracy becomes severe, noise overwhelms these subspaces, causing the smallest singular values to rapidly approach zero. This further diminishes the weight matrix's effective rank, rendering the model increasingly unstable and less reliable in these directions.

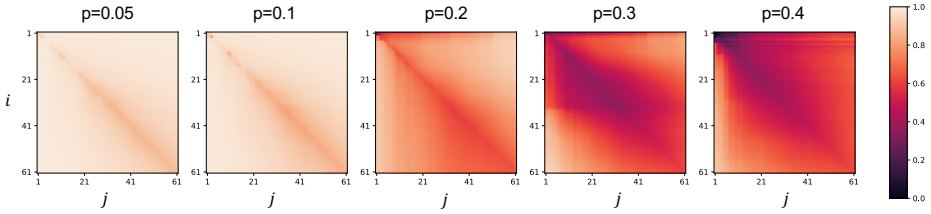

Fig. 2: Subspace similarity between top-i column vectors of $\mathbf{W}$ and top-j of $\mathbf{W}'$. As label inaccuracy increases, the weight matrix is able to preserve the characteristics of the principal subspace within a certain range of inaccuracy, thus retaining the most important task-specific information. However, beyond this range, the principal subspace of the weights begins to change, and even eventually becomes completely uncorrelated, leading to the loss of critical information. For clearer illustrations, we present the analysis for $i, j \in [1, 60]$.

While the analysis of singular values provided important insights into how label inaccuracy affects the weight matrix, it leaves some critical aspects unclear. Specifically, while we can see the reduction in the rank and the weakened information retention in both primary and other directions, it remains ambiguous why the model is still capable of learning from inaccurate labels. In other words, we do not yet fully understand how much of the essential task-specific information is preserved in the weight matrix despite the presence of label inaccuracy. To further explore this, we now turn to the subspace similarities between $\mathbf{W}$ and $\mathbf{W}'$. By examining the subspace structure, we seek to understand how label inaccuracy influences the information encoded in the weight matrix and to identify whether a shared structure enables the extraction of correct information from inaccurate labels.

3.2 Similarities of Subspace

We assess the similarities of the subspace spanned by the top-i singular vectors in $\mathbf{V}$ and that of the top-j singular vectors of $\mathbf{V}'$. Following [16], we compute the normalized subspace similarity based on the Grassmann distance as

$$\phi(\mathbf{V}, \mathbf{V}', i, j) = \frac{\|\mathbf{V}_{:i}^{\top} \mathbf{V}'_{:j}\|_F^2}{\min(i, j)} \in [0, 1]. \tag{1}$$

Here, $\phi(\cdot)$ ranges from 0 to 1, where 1 indicates a complete overlap of subspaces and 0 signifies total separation. $\mathbf{V}_{:i}$ and $\mathbf{V}'_{:j}$ represent the top-i and top-j column vectors of $\mathbf{V}$ and $\mathbf{V}'$, respectively. As shown in Fig. 2, we observed an important phenomenon: as label inaccuracy increases, the overall divergence between the subspaces of $\mathbf{V}$ and $\mathbf{V}'$ becomes more pronounced, with the most notable changes in the **principal subspace** (corresponding to the largest singular values). Since the top singular components capture the most crucial task-specific information, changes in the principal subspace signify the loss of important information, leading to weaker de-noise ability, which can be validated by the classification accuracy in Fig. 1. Therefore, label inaccuracy not only introduces noise in the lower singular components, but also disrupts the principal subspace, leading to the loss of key information.

However, surprisingly, within a certain range of label inaccuracy ($p \leq 0.1$ in our experiments), the principal subspace of the weights remains largely unaffected and even nearly identical. As seen in Fig. 1, the classification results of $\mathbf{W}'$ under this level of inaccuracy are nearly identical to those of $\mathbf{W}$. This similarity between the principal subspaces of $\mathbf{V}$ and $\mathbf{V}'$ indicates that $\mathbf{W}'$ effectively preserves task-specific information, as if it were learned from clean data. **This may be a key reason why the model can still effectively learn from inaccurate annotations.**

It appears we are moving closer to the truth. However, to truly grasp the essence, we must address a pivotal question: Why are their principal subspaces so similar in a certain range of label inaccuracy?

4 Theoretical Analysis

4.1 Label Inaccuracy can be Viewed as a Form of Weight Perturbation

Formally speaking, suppose $\mathcal{X} = \mathbb{R}^q$ denotes the q-dimensional feature space, and $\mathcal{Y} = \{0, 1\}^l$ is the label space with l classes. Given an inaccurate label data set $\mathcal{D} = \{\mathbf{x}_i, \mathcal{S}_i | 1 \leq i \leq n\}$, where $\mathbf{x}_i \in \mathcal{X}$ is the i-th sample and $\mathcal{S}_i \subseteq \mathcal{Y}$ is the corresponding corrupted label set, we learn a classifier $f : \mathcal{X} \to \mathcal{Y}$ based on $\mathcal{D}$. Suppose $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^{\top} \in \mathbb{R}^{n \times q}$ denote the q-dimensional instance matrix with n instances, and $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n]^{\top} \in \{0, 1\}^{n \times l}$ represents the ground-truth label matrix. We aim to learn a weight matrix $\mathbf{W} \in \mathbb{R}^{q \times l}$ to map the

instance matrix to the ground-truth one. Consider the following simple classifier (or loss function):

$$\min_{\mathbf{W}} \|\mathbf{X}\mathbf{W} - \mathbf{G}\|_F^2 + \lambda\|\mathbf{W}\|_F^2, \tag{2}$$

which is a regularized least squares problem, with λ as a hyper-parameter. The closed-form solution is

$$\mathbf{W} = (\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{G}, \tag{3}$$

where $\mathbf{I}$ is the identity matrix. For convenience, we denote $\mathbf{K} = \mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I}$ to simplify following formula expressions.

When handling inaccurate labels, we define the corrupted label set $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]^\mathsf{T} \in \{0,1\}^{n\times l}$, where $y_{ij} = 1$ if the j-th label of $\mathbf{x}_i$ is annotated as its "ground-truth" label. Indeed, we can consider $\mathbf{Y}$ as a perturbation of $\mathbf{G}$, where $\mathbf{Y} = \mathbf{G} + \mathbf{M}$. Here, $\mathbf{M}$ is a perturbation mask composed of elements from $-1, 0, 1$, representing the label inaccuracy. To ensure label consistency, we impose the constraint $\mathbf{M} + \mathbf{G} \geq 0$, meaning that M_{ij} can take the value -1 only where $G_{ij} = 1$, ensuring that incorrectly removed labels originate from initially assigned ones. Therefore, the learned weight matrix $\mathbf{W}'$ under supervision of $\mathbf{Y}$ is

$$\begin{aligned}\mathbf{W}' &= \mathbf{K}^{-1}\mathbf{X}^\mathsf{T}\mathbf{Y} = \mathbf{K}^{-1}\mathbf{X}^\mathsf{T}\mathbf{G} + \mathbf{K}^{-1}\mathbf{X}^\mathsf{T}\mathbf{M} \\ &= \mathbf{W} + \Delta\mathbf{W}.\end{aligned} \tag{4}$$

Consequently, the impact of label inaccuracy on the weight $\mathbf{W}$ can be described as introducing an additional perturbation, denoted as $\Delta\mathbf{W} = \mathbf{K}^{-1}\mathbf{X}^\mathsf{T}\mathbf{M}$.

4.2 Estimation of Upper Bound of Perturbation

We then calculate the upper bound of $\Delta\mathbf{W}$. According to norm inequalities [34], we have

$$\begin{aligned}\|\Delta\mathbf{W}\|_F &\leq \|\mathbf{K}^{-1}\mathbf{X}^\mathsf{T}\|_2\|\mathbf{M}\|_F \\ &\leq \|\mathbf{K}^{-1}\|_2\|\mathbf{X}^\mathsf{T}\|_2\|\mathbf{M}\|_F,\end{aligned} \tag{5}$$

where $\|\cdot\|_F$ represents the Frobenius norm and $\|\cdot\|_2$ denotes the spectral norm of a matrix, respectively. Denote $\lambda_{\min}(\cdot)$ as the minimum eigenvalue of a matrix, and $\sigma_{\max}(\cdot)$ as the maximum singular value of a matrix. Due to $q \ll n$ generally, $\mathbf{X}$ is expected to be full rank. We therefore have

$$\begin{aligned}\|\mathbf{K}^{-1}\|_2 = \|(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}\|_2 &= \frac{1}{\lambda_{\min}(\mathbf{X}^\mathsf{T}\mathbf{X}) + \lambda} \\ \|\mathbf{X}^\mathsf{T}\|_2 = \|\mathbf{X}\|_2 &= \sigma_{\max}(\mathbf{X}).\end{aligned} \tag{6}$$

For simplicity, we define the degree of label inaccuracy as p, where $P(M_{ij} \neq 0) = p$. Consequently, the total number of non-zero elements in $\mathbf{M}$ is pnl, and we have $\|\mathbf{M}\|_F = \sqrt{pnl}$. Therefore, the upper bound of $\Delta\mathbf{W}$ is given by

$$\|\Delta\mathbf{W}\|_F \leq \frac{\sigma_{\max}(\mathbf{X})\sqrt{nl}}{\lambda_{\min}(\mathbf{X}^\mathsf{T}\mathbf{X}) + \lambda}\sqrt{p}. \tag{7}$$

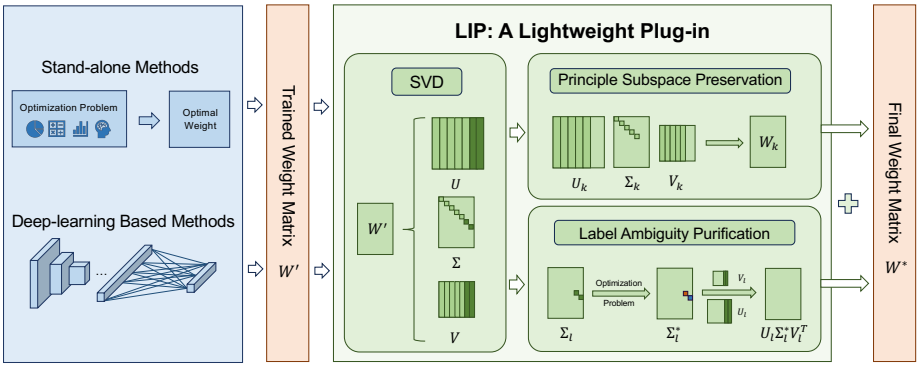

Fig. 3: Framework of LIP. Once a method is trained, the LIP applies post-processing to the trained weights used for classification. The two modules of the LIP, Principle Subspace Preservation and Label Ambiguity Purification, are utilized to retain critical information in the principal subspace and to rectify noise resulting from label inaccuracy, respectively. Given the high processing speed and extreme lightness of these modules, coupled with the absence of a need for training, LIP emerges as an efficient and lightweight plug-in.

4.3 Subspace Perturbation Analysis

To assess the impact of perturbation on the subspace, we apply the Davis-Kahan sine theorem [6], which provides an upper bound on the change in the principal angles between the perturbed and unperturbed subspaces. Specifically, the theorem states that for a matrix perturbation $\Delta\mathbf{W}$, the sine of the angle θ between the subspaces of $\mathbf{W}$ and $\mathbf{W}'$ is bounded by

$$\sin\theta \leq \frac{\|\Delta\mathbf{W}\|_2}{\delta}, \tag{8}$$

where δ denotes the gap between the principle singular values of $\mathbf{W}$ (it can be viewed as a hyper-parameter). Since $\|\Delta\mathbf{W}\|_2 \leq \|\Delta\mathbf{W}\|_F$ [34], the bound becomes

$$\sin\theta \leq \frac{\sigma_{\max}(\mathbf{X})\sqrt{nl}}{\delta(\lambda_{\min}(\mathbf{X}^\top\mathbf{X}) + \lambda)}\sqrt{p}. \tag{9}$$

From this inequality we can draw an important conclusion: when the degree of label inaccuracy p remains within a reasonable range, the angle remains small, and the principal subspace of the weight matrix $\mathbf{W}$ does not undergo significant changes or the changes are controllable. This implies that even in the presence of label inaccuracy, classifiers can still learn information crucial for classification that can be learned from clean labels. Therefore, through experimental results and theoretical analysis, we ultimately determine why models can learn correct information from inaccurate annotations: ***A certain degree of label inaccuracy does not cause the principal subspace of the information learned from clean data to shift severely.***

Furthermore, in the appendix we provide additional analyses, including: (1) our rationale for focusing on the final FC layer and the choice of least-squares fitting (Sec. 8.1); (2) the potential impact of different model architectures—such as Transformers—on our conclusions (Sec. 8.2); and (3) some remarks on our conclusion (Sec. 8.3).

5 LIP: A Lightweight Plug-in

According to our previous analysis, label inaccuracy within a certain range does not significantly impact the critical information in the principal subspace. This naturally leads us to contemplate whether a method can be devised that not only preserves the information in the principal subspace but also reduces or even refine the impact of noise introduced by label inaccuracy. Based on this consideration, we propose a post-processing plug-in, named LIP. LIP primarily consists of two phases: Principle Subspace Preservation (PSP) and Label Ambiguity Purification (LAP). The framework of LIP is shown in Fig. 3.

5.1 Principle Subspace Preservation

Consider the weight matrix $\mathbf{W}'$ learned from datasets with inaccurate labels. For a stand-alone method, $\mathbf{W}'$ represents the ultimate optimization goal; for deep-learning-based methods, $\mathbf{W}'$ corresponds to the weights of the final FC layer used for classification. As demonstrated in Sec. 3-4, the crucial information for classification using inaccurate labels is preserved within the principal subspace of the weights, i.e., the top singular components of $\mathbf{W}'$. Therefore, we initially focus on preserving this information.

We perform SVD on $\mathbf{W}'$ to obtain the corresponding left and right singular vectors $\mathbf{U}$ and $\mathbf{V}$, as well as the singular value matrix $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{\min(q,l)})$ with σ_i arranged in descending order. Let k ($k \leq \min(q, l)$) denote the number of top singular components to be retained, which is a hyper-parameter. Let $\mathbf{U}_k = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k]$, $\mathbf{V}_k = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$ represent the top k singular vectors and $\mathbf{\Sigma}_k = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k)$. Thus, we can express the matrix $\mathbf{W}_k$ which contains the crucial information derived from principle subspace as:

$$\mathbf{W}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^{\mathsf{T}}. \tag{10}$$

5.2 Label Ambiguity Purification

The matrix $\mathbf{W}_k$, which encapsulates the principal subspace's crucial information, has already mitigated the noise introduced by label inaccuracy. However, several studies [13,39,41,42] highlight the importance of the bottom singular components to the weight matrix itself. Notably, as illustrated in Fig. 1, these bottom singular values often appear flattened and near-zero due to label inaccuracy, suggesting that label noise may significantly suppress these important components. Building on this observation, we propose that these components are often misperceived as

noise. Instead of discarding them, we aim to refine these singular values, allowing them to recapture essential information from the training data and reassess the importance of each singular component, thereby enhancing the utilization of the matrix's effective rank.

Specifically, suppose $\mathbf{U}_l = [\mathbf{u}_{k+1}, \mathbf{u}_{k+2}, \ldots, \mathbf{u}_{\min(q,l)}]$ and $\mathbf{V}_l = [\mathbf{v}_{k+1}, \mathbf{v}_{k+2}, \ldots, \mathbf{v}_{\min(q,l)}]$ denote the singular vectors not retained in the PSP phase, and $\mathbf{\Sigma}_l = \mathrm{diag}(\sigma_{k+1}, \sigma_{k+2}, \ldots, \sigma_{\min(q,l)})$ represents the singular values not preserved. With these definitions, we can express the weight matrix $\mathbf{W}'$ as follows:

$$\mathbf{W}' = \mathbf{W}_k + \mathbf{U}_l \mathbf{\Sigma}_l \mathbf{V}_l^\mathsf{T}. \tag{11}$$

To refine the singular values, we retrain them using the training data to extract critical information, which leads to the following optimization problem:

$$\begin{aligned} \min_{\mathbf{\Sigma}_l} \quad & \|\mathbf{X}\mathbf{W}' - \mathbf{Y}\|_F^2 \\ \text{s.t.} \quad & \mathbf{W}' = \mathbf{W}_k + \mathbf{U}_l \mathbf{\Sigma}_l \mathbf{V}_l^\mathsf{T}. \end{aligned} \tag{12}$$

By solving the aforementioned optimization problem, we can determine the optimal singular matrix $\mathbf{\Sigma}_l^*$. Consequently, this enables us to derive the final weight matrix $\mathbf{W}^*$ for classification as

$$\mathbf{W}^* = \mathbf{W}_k + \mathbf{U}_l \mathbf{\Sigma}_l^* \mathbf{V}_l^\mathsf{T}. \tag{13}$$

This matrix not only preserves information in the principal subspace but also effectively addresses and refines the noise associated with label inaccuracy.

5.3 Numerical Solution to Eq. (12)

We first reformulate Eq. (12) as

$$\min_{\mathbf{\Sigma}_l} \quad \|\mathbf{X}(\mathbf{W}_k + \sum_{i=k+1}^{\min(q,l)} \sigma_i \mathbf{u}_i \mathbf{v}_i^\mathsf{T}) - \mathbf{Y}\|_F^2. \tag{14}$$

Taking the gradient of Eq. (14) w.r.t. each σ_j, we have

$$\nabla_{\sigma_j} = 2(\sigma_j \mathbf{u}_j^\mathsf{T} \mathbf{X}^\mathsf{T} \mathbf{X} \mathbf{u}_j + \mathbf{u}_j^\mathsf{T} \mathbf{X}^\mathsf{T} (\mathbf{X}\mathbf{W}_k - \mathbf{Y}) \mathbf{v}_j). \tag{15}$$

The optimal solution of Eq. (14) is achieved when $\nabla_{\sigma_j} = 0$, and accordingly we have

$$\sigma_j = \frac{\mathbf{u}_j^\mathsf{T} \mathbf{X}^\mathsf{T} (\mathbf{Y} - \mathbf{X}\mathbf{W}_k) \mathbf{v}_j}{\mathbf{u}_j^\mathsf{T} \mathbf{X}^\mathsf{T} \mathbf{X} \mathbf{u}_j}. \tag{16}$$

Thus, the corresponding closed-form solution to the optimization problem in Eq. (12) is:

$$\mathbf{\Sigma}_l^* = \frac{\mathrm{diag}(\mathbf{U}_l^\mathsf{T} \mathbf{X}^\mathsf{T} (\mathbf{Y} - \mathbf{X}\mathbf{W}_k) \mathbf{V}_l)}{\mathrm{diag}(\mathbf{U}_l^\mathsf{T} \mathbf{X}^\mathsf{T} \mathbf{X} \mathbf{U}_l)}. \tag{17}$$

Given that this problem has a closed-form solution, LIP does not require training (i.e., it is training-free). Instead, LIP can be directly integrated into any pre-trained classifier, adding only two computational steps to preserve crucial information and refine noise from label inaccuracy.

Table 1: Classification accuracy of each compared approach on the real-world data sets. * denotes these datasets are processed with noise. For any baselines, •/◦ indicates whether the performance of baseline coupled with LIP is statistically superior/inferior to that of the baseline itself based on pairwise t-test at significance level of 0.05. "↑Ratio" represents the average improvement ratio of each baseline by LIP.

Approaches	Data set						Avg.	↑Ratio
	FG-NET*	Lost*	MSRCv2*	Mirflickr*	Soccer Player*	Yahoo!News*		
PL-CL	6.6 ± 1.2	68.3 ± 2.4	45.1 ± 1.5	63.2 ± 1.3	52.1 ± 0.5	60.9 ± 0.4	49.4	5.7%
+ LIP	6.9 ± 1.0 •	72.8 ± 1.6 •	52.1 ± 2.0 •	64.1 ± 1.6 •	53.6 ± 0.5 •	63.9 ± 0.3 •	**52.2**	
PL-CGR	5.9 ± 1.3	71.3 ± 3.0	49.7 ± 2.3	64.5 ± 1.6	53.3 ± 0.9	61.6 ± 0.5	51.1	4.6%
+LIP	7.9 ± 1.3 •	74.3 ± 2.8 •	51.6 ± 2.3 •	65.8 ± 1.8 •	55.9 ± 0.8 •	65.1 ± 0.3 •	**53.4**	
DPCLS	5.7 ± 0.7	69.6 ± 1.8	53.6 ± 1.4	61.2 ± 1.6	54.7 ± 0.7	61.9 ± 0.4	51.1	3.3%
+LIP	7.8 ± 1.0 •	72.7 ± 1.8 •	54.5 ± 1.4 •	62.3 ± 1.6 •	55.7 ± 0.6 •	63.8 ± 0.3 •	**52.8**	
PL-AGGD	6.3 ± 0.9	68.2 ± 3.0	45.0 ± 1.5	63.7 ± 1.2	51.7 ± 0.5	60.7 ± 0.2	49.3	6.3%
+ LIP	6.7 ± 0.7 •	72.9 ± 2.5 •	52.3 ± 1.8 •	64.8 ± 1.4 •	53.6 ± 0.6 •	63.9 ± 0.3 •	**52.4**	
SURE	5.4 ± 1.1	68.3 ± 2.0	43.8 ± 2.3	61.4 ± 1.1	51.2 ± 0.5	59.1 ± 0.3	48.2	3.7%
+ LIP	5.9 ± 1.0 •	70.6 ± 1.8 •	48.1 ± 2.1 •	62.2 ± 1.2 •	51.8 ± 0.4 •	61.6 ± 0.3 •	**50.0**	
LALO	6.0 ± 0.8	66.7 ± 2.0	44.1 ± 1.7	62.1 ± 1.5	51.5 ± 0.4	59.3 ± 0.3	48.3	7.0%
+ LIP	6.7 ± 0.8 •	70.9 ± 1.9 •	51.3 ± 1.4 •	64.4 ± 1.9 •	53.5 ± 0.5 •	63.4 ± 0.4 •	**51.7**	
PL-LEAF	6.4 ± 0.8	65.1 ± 2.9	45.7 ± 2.1	62.6 ± 0.9	51.1 ± 0.6	60.0 ± 0.3	48.5	3.9%
+ LIP	7.3 ± 0.9 •	69.4 ± 2.3 •	48.3 ± 2.1 •	63.6 ± 0.8 •	51.8 ± 0.6 •	61.7 ± 1.0 •	**50.4**	

Table 2: Classification accuracy(%) on CIFAR-100. Experiments are conducted under asymmetric and symmetric conditions.

Approaches	Asymmetric				Symmetric			
	10%	20%	40%	50%	10%	20%	40%	80%
PLS	79.54	73.90	53.17	34.51	79.65	79.13	77.27	45.92
+ LIP	79.75 •	74.34 •	54.42 •	38.49 •	79.87 •	79.69 •	77.38 •	47.43 •
AGCE	67.32	63.57	47.25	27.90	67.61	64.75	59.78	24.05
+ LIP	67.56 •	63.99 •	48.79 •	28.25 •	67.98 •	64.93 •	59.91 •	25.43 •

Table 3: Classification accuracy on CIFAR-100 and CUB-200. "Clean" represents the performance learnt from clean datasets.

Approaches	CIFAR-100			CUB-200		
	$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.02$	$p = 0.04$	$p = 0.06$
PICO	73.51 ± 0.22%	72.18 ± 0.32%	71.62 ± 0.18%	72.56 ± 0.07%	72.27 ± 0.46%	71.88 ± 0.33%
+ LIP	73.83 ± 0.04% •	72.43 ± 0.21% •	71.98 ± 0.11% •	72.90 ± 0.04% •	72.56 ± 0.02% •	72.26 ± 0.03% •
Clean	Clean: 73.88 ± 0.07 %		Clean + LIP: 74.09 ± 0.03% •	Clean: 76.00 ± 0.34%		Clean + LIP: 76.56 ± 0.06% •
PRODEN	77.12 ± 0.13%	76.04 ± 0.16%	56.83 ± 0.05%	74.53 ± 0.03%	74.36 ± 0.15%	72.01 ± 0.18%
+ LIP	77.25 ± 0.03% •	76.42 ± 0.05% •	56.90 ± 0.01% •	74.73 ± 0.01% •	74.51 ± 0.02% •	72.33 ± 0.03% •
Fully Supervised	Clean: 77.31 ± 0.07%		Clean + LIP: 77.48 ± 0.14% •	Clean: 75.70 ± 0.10%		Clean + LIP: 75.91 ± 0.03% •

5.4 Complexity Analysis

We here present the computational complexity of LIP. Specifically, PSP focuses on performing SVD on the weights and retaining the principal components, with

Table 4: Classification accuracy on real-world datasets.

Dataset	CIFAR-10N	CIFAR-100N	Clothing1M
PLS	85.99	55.54	47.59
+LIP	86.69	56.46	48.36

the complexity of $O(\min(q^2l, ql^2))$. LAP focuses on refining the singular values, with the complexity of $O(nql)$. Therefore, the overall computational complexity is $O(nql + \min(q^2l, ql^2))$. In practice, training a classifier requires several hours. **In contrast, LIP can achieve a substantial performance improvement in just one second.** For example, the execution times for CIFAR- 100 and CUB-200 are 9.98 ms and 16.13 ms, respectively.

6 Experiments

6.1 Baselines

To evaluate the effectiveness of LIP, we integrate it with a variety of approaches including PL-CL [20], PL-CGR [55], DPCLS [21], PL-AGGD [44], SURE [8], LALO [7], PL-LEAF [54], PRODEN [26], PICO [45], PLS [1] and AGCE [56]. For more details on the baselines, please refer to the appendix.

6.2 Datasets and Settings

We considered various dataset configurations to explore different forms of label inaccuracy. First, we evaluated six real-world partial label datasets collected from diverse domains and tasks, including FG-NET [31], Lost [4], Soccer Player [53], Yahoo!News [11], MSRCv2 [25], and Mirflickr [18]. In these datasets, each sample, in addition to its correct label, was also assigned several other labels as "ground-truth" labels. Building on this, we further increased the difficulty by randomly selecting 10% of the samples and completely removing their correct labels, thereby creating an even more challenging dataset.

Additionally, we considered two simplified variations of the above setting. In the first variation, each sample either retains its correct label or is entirely mislabeled. For this, following [36], we used CIFAR-100 [22], which contains both asymmetric and symmetric label noise [43]. In the second variation, each sample retains its correct label but also contains additional erroneous labels as ground-truth. For this setting, we conducted experiments on two benchmark datasets: CIFAR-100 and CUB-200 [48]. Following [26, 45], we generated false positive labels by flipping negative labels for each sample with a defined probability p and then combining them with the ground-truth labels to form the final candidate label set. Specifically, for CIFAR-100, p was set to 0.05, 0.1, and 0.2, while for CUB-200, p was set to 0.02, 0.04, and 0.06. For more details on these datasets, please refer to the Appendix. Moreover, we also evaluate the performance of LIP on three real-world datasets, CIFAR-10N [47], CIFAR-100N [50], and Clothing1M [59].

For implementation details, please refer to the appendix.

Table 5: Ablation study of LIP coupled with PL-CL.

PSP	LAP	Data set						Avg.
		FG-NET*	Lost*	MSRCv2*	Mirflickr*	Soccer Player*	Yahoo!News*	
×	×	6.6 ± 1.2	68.3 ± 2.4	45.1 ± 1.5	63.2 ± 1.3	52.1 ± 0.5	60.9 ± 0.4	49.4
✓	×	6.8 ± 1.0	71.9 ± 1.9	50.2 ± 1.6	63.1 ± 1.4	52.5 ± 0.5	61.7 ± 0.2	51.0
✓	✓	6.9 ± 1.0	72.8 ± 1.6	52.1 ± 2.0	64.1 ± 1.6	53.6 ± 0.5	63.9 ± 0.3	52.2

Table 6: Ablation study of LIP coupled with PRODEN.

PSP	LAP	CIFAR-100			CUB-200		
		$p = 0.05$	$p = 0.1$	$p = 0.2$	$p = 0.02$	$p = 0.04$	$p = 0.06$
×	×	77.12 ± 0.13%	76.04 ± 0.16%	56.83 ± 0.05%	74.53 ± 0.03%	74.36 ± 0.15%	72.01 ± 0.18%
✓	×	77.21 ± 0.01%	76.25 ± 0.15%	56.89 ± 0.01%	74.66 ± 0.01%	74.41 ± 0.07%	72.17 ± 0.04%
✓	✓	77.25 ± 0.03%	76.42 ± 0.05%	56.90 ± 0.01%	74.73 ± 0.01%	74.51 ± 0.02%	72.33 ± 0.03%

6.3 Results

Tables 1-4 present the experimental results of LIP, where we can observe

- LIP significantly improves the performance of all these approaches across all cases.
- For state-of-the-art methods like PL-CL, PICO, and PLS, LIP serves as a powerful enhancement, further boosting their already impressive performance. For example, on the MSRCv2 dataset, PL-CL experiences a substantial **15.5%** improvement, which showcases LIP's capacity to elevate even the top-tier models.
- Other algorithms, such as LALO and PL-LEAF, can achieve even superior performance compared to state-of-the-art methods when coupled with LIP. This suggests that LIP not only enhances established approaches but also unlocks the latent potential of these algorithms.
- For deep learning-based approaches, LIP enables them to approach the performance of models trained on clean, fully supervised data, even in the presence of noisy or incomplete labels. Remarkably, LIP is not only beneficial in weakly supervised scenarios but also in fully supervised settings. LIP can still provide performance gains, further refining the model's accuracy and generalization.

6.4 Ablation Study

We conducted an ablation study to assess the contributions of the PSP and LAP modules to the overall performance of LIP. Table 5 and Table 3 in the appendix present the experimental results when LIP is coupled with PL-CL and PRODEN, respectively. It is obvious that both PSP and LAP contribute positively

to the performance of the model across all datasets. PSP effectively preserves the information within the principal subspace, while LAP further refines the noise introduced by label ambiguity. The interaction between these two modules enhances the model's ability when handling inaccurate labels.

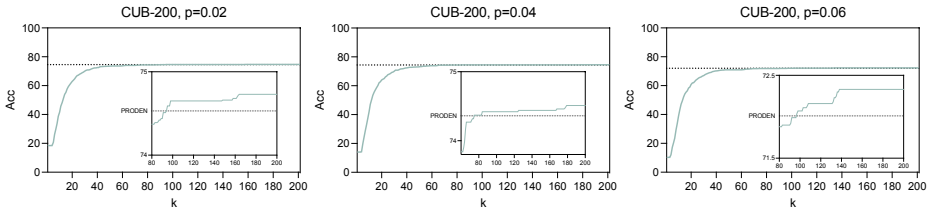

Fig. 4: Sensitivity analysis on k. LIP is coupled with PRODEN on CUB-200 dataset.

6.5 Sensitive Analysis

We also performed a sensitivity analysis on the parameter k of LIP to assess its impact on performance. Specifically, LIP was coupled with PRODEN on the CUB-200 dataset, with the results displayed in Fig. 4. As the value of k increases, the overall performance of the LIP-augmented model consistently improves, eventually stabilizing after a certain threshold. Specifically, once k surpasses 100, the combination of LIP with PRODEN consistently delivers superior results compared to PRODEN alone. This stability of LIP w.r.t. parameter k is a significant advantage, making it highly reliable and robust for practical applications.

7 Conclusion

In this paper, we systematically investigate why models can extract correct label information from inaccurate annotations. By analyzing the weight matrix used for classification, both empirically and theoretically, we reveal that within a certain range, label inaccuracy does not significantly alter the principal subspace of the weight matrix, and in some cases, even leaves it unchanged. This observation suggests that the core structure encoding task-relevant information remains robust to moderate levels of label corruption, enabling effective learning despite annotation errors. Building on these insights, we introduce LIP, a lightweight post-processing plug-in that requires no additional training yet significantly enhances the performance of existing methods. By selectively refining model outputs while preserving critical subspace information, LIP provides a simple yet effective solution to mitigating the impact of label inaccuracy. Our extensive evaluations confirm its effectiveness in improving classification performance. We hope that this work not only deepens the understanding of why models can learn from inaccurate annotations but also inspires new strategies for enhancing robustness in weakly supervised learning paradigms.

Acknowledgements

This work was supported by the NSFC under Grant 62322604 and 62576207.

References

1. Albert, P., Arazo, E., Krishna, T., O'Connor, N.E., McGuinness, K.: Is your noise correction noisy? pls: Robustness to label noise with two stage detection. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 118–127 (2023)
2. Bai, Y., Yang, E., Han, B., Yang, Y., Li, J., Mao, Y., Niu, G., Liu, T.: Understanding and improving early stopping for learning with noisy labels. Advances in Neural Information Processing Systems **34**, 24392–24403 (2021)
3. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: A holistic approach to semi-supervised learning. Advances in neural information processing systems **32** (2019)
4. Cour, T., Sapp, B., Jordan, C., Taskar, B.: Learning from ambiguously labeled images. In: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA. pp. 919–926. IEEE Computer Society (2009). https://doi.org/10.1109/CVPR.2009.5206667, https://doi.org/10.1109/CVPR.2009.5206667
5. Cour, T., Sapp, B., Taskar, B.: Learning from partial labels. J. Mach. Learn. Res. **12**, 1501–1536 (2011). https://doi.org/10.5555/1953048.2021049, https://dl.acm.org/doi/10.5555/1953048.2021049
6. Davis, C., Kahan, W.M.: The rotation of eigenvectors by a perturbation. iii. SIAM Journal on Numerical Analysis **7**(1), 1–46 (1970)
7. Feng, L., An, B.: Leveraging latent label distributions for partial label learning. In: Lang, J. (ed.) Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden. pp. 2107–2113. ijcai.org (2018). https://doi.org/10.24963/ijcai.2018/291, https://doi.org/10.24963/ijcai.2018/291
8. Feng, L., An, B.: Partial label learning with self-guided retraining. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019. pp. 3542–3549. AAAI Press (2019). https://doi.org/10.1609/aaai.v33i01.33013542, https://doi.org/10.1609/aaai.v33i01.33013542
9. Ghosh, A., Kumar, H., Sastry, P.S.: Robust loss functions under label noise for deep neural networks. In: Proceedings of the AAAI conference on artificial intelligence. No. 1 (2017)
10. Goldberger, J., Ben-Reuven, E.: Training deep neural-networks using a noise adaptation layer. In: International conference on learning representations (2017)
11. Guillaumin, M., Verbeek, J., Schmid, C.: Multiple instance metric learning from automatically labeled bags of faces. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I. Lecture Notes in Computer Science, vol. 6311, pp. 634–647. Springer (2010). https://doi.org/10.1007/978-3-642-15549-9_46, https://doi.org/10.1007/978-3-642-15549-9_46

12. Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., Sugiyama, M.: Co-teaching: Robust training of deep neural networks with extremely noisy labels. Advances in neural information processing systems **31** (2018)
13. Han, L., Li, Y., Zhang, H., Milanfar, P., Metaxas, D., Yang, F.: Svdiff: Compact parameter space for diffusion fine-tuning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7323–7334 (2023)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
15. He, S., Feng, L., Lv, F., Li, W., Yang, G.: Partial label learning with semantic label representations. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 545–553 (2022)
16. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
17. Huang, J., Qu, L., Jia, R., Zhao, B.: O2u-net: A simple noisy label detection approach for deep neural networks. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3326–3334 (2019)
18. Huiskes, M.J., Lew, M.S.: The MIR flickr retrieval evaluation. In: Lew, M.S., Bimbo, A.D., Bakker, E.M. (eds.) Proceedings of the 1st ACM SIGMM International Conference on Multimedia Information Retrieval, MIR 2008, Vancouver, British Columbia, Canada, October 30-31, 2008. pp. 39–43. ACM (2008). https://doi.org/10.1145/1460096.1460104, https://doi.org/10.1145/1460096.1460104
19. Hüllermeier, E., Beringer, J.: Learning from ambiguously labeled examples. In: Famili, A.F., Kok, J.N., Sánchez, J.M.P., Siebes, A., Feelders, A.J. (eds.) Advances in Intelligent Data Analysis VI, 6th International Symposium on Intelligent Data Analysis, IDA 2005, Madrid, Spain, September 8-10, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3646, pp. 168–179. Springer (2005). https://doi.org/10.1007/11552253_16, https://doi.org/10.1007/11552253_16
20. Jia, Y., Si, C., Zhang, M.l.: Complementary classifier induced partial label learning. arXiv preprint arXiv:2305.09897 (2023)
21. Jia, Y., Yang, F., Dong, Y.: Partial label learning with dissimilarity propagation guided candidate label shrinkage. Advances in neural information processing systems **36**, 34190–34200 (2023)
22. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
23. Li, J., Socher, R., Hoi, S.C.: Dividemix: Learning with noisy labels as semi-supervised learning. arXiv preprint arXiv:2002.07394 (2020)
24. Li, J., Wong, Y., Zhao, Q., Kankanhalli, M.S.: Learning to learn from noisy labeled data. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5051–5059 (2019)
25. Liu, L., Dietterich, T.: A conditional multinomial mixture model for superset label learning. Advances in neural information processing systems **25** (2012)
26. Lv, J., Xu, M., Feng, L., Niu, G., Geng, X., Sugiyama, M.: Progressive identification of true labels for partial-label learning. In: international conference on machine learning. pp. 6500–6510. PMLR (2020)
27. Lyu, G., Wu, Y., Feng, S.: Deep graph matching for partial label learning. In: Proceedings of the International Joint Conference on Artificial Intelligence. pp. 3306–3312 (2022)

28. Ma, X., Huang, H., Wang, Y., Romano, S., Erfani, S., Bailey, J.: Normalized loss functions for deep learning with noisy labels. In: International conference on machine learning. pp. 6543–6553. PMLR (2020)
29. Nguyen, D.T., Mummadi, C.K., Ngo, T.P.N., Nguyen, T.H.P., Beggel, L., Brox, T.: Self: Learning to filter noisy labels with self-ensembling. arXiv preprint arXiv:1910.01842 (2019)
30. Nguyen, N., Caruana, R.: Classification with partial labels. In: Li, Y., Liu, B., Sarawagi, S. (eds.) Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008. pp. 551–559. ACM (2008). https://doi.org/10.1145/1401890.1401958, https://doi.org/10.1145/1401890.1401958
31. Panis, G., Lanitis, A., Tsapatsoulis, N., Cootes, T.F.: Overview of research on facial ageing using the FG-NET ageing database. IET Biom. **5**(2), 37–46 (2016). https://doi.org/10.1049/iet-bmt.2014.0053, https://doi.org/10.1049/iet-bmt.2014.0053
32. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
33. Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., Qu, L.: Making deep neural networks robust to label noise: A loss correction approach. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1944–1952 (2017)
34. Petersen, K.B., Pedersen, M.S., et al.: The matrix cookbook. Technical University of Denmark **7**(15), 510 (2008)
35. Ren, M., Zeng, W., Yang, B., Urtasun, R.: Learning to reweight examples for robust deep learning. In: Dy, J.G., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018. Proceedings of Machine Learning Research, vol. 80, pp. 4331–4340. PMLR (2018), http://proceedings.mlr.press/v80/ren18a.html
36. Sheng, M., Sun, Z., Cai, Z., Chen, T., Zhou, Y., Yao, Y.: Adaptive integration of partial label learning and negative learning for enhanced noisy label learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 4820–4828 (2024)
37. Shu, J., Zhao, Q., Xu, Z., Meng, D.: Meta transition adaptation for robust deep learning with noisy labels. arXiv preprint arXiv:2006.05697 (2020)
38. Si, C., Jiang, Z., Wang, X., Wang, Y., Yang, X., Shen, W.: Partial label learning with a partner. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 15029–15037 (2024)
39. Si, C., Shi, Z., Zhang, S., Yang, X., Pfister, H., Shen, W.: Unleashing the power of task-specific directions in parameter efficient fine-tuning. arXiv preprint arXiv:2409.01035 (2024)
40. Si, C., Wang, X., Wang, Y., Yang, X., Shen, W.: Appeal: Allow mislabeled samples the chance to be rectified in partial label learning. arXiv preprint arXiv:2312.11034 (2023)
41. Si, C., Wang, X., Yang, X., Xu, Z., Li, Q., Dai, J., Qiao, Y., Yang, X., Shen, W.: Flora: Low-rank core space for n-dimension. arXiv preprint arXiv:2405.14739 (2024)
42. Si, C., Yang, X., Shen, W.: See further for parameter efficient fine-tuning by standing on the shoulders of decomposition. arXiv preprint arXiv:2407.05417 (2024)

43. Sun, Z., Yao, Y., Wei, X.S., Zhang, Y., Shen, F., Wu, J., Zhang, J., Shen, H.T.: Webly supervised fine-grained recognition: Benchmark datasets and an approach. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10602–10611 (2021)
44. Wang, D., Zhang, M., Li, L.: Adaptive graph guided disambiguation for partial label learning. IEEE Trans. Pattern Anal. Mach. Intell. **44**(12), 8796–8811 (2022). https://doi.org/10.1109/TPAMI.2021.3120012, https://doi.org/10.1109/TPAMI.2021.3120012
45. Wang, H., Xiao, R., Li, Y., Feng, L., Niu, G., Chen, G., Zhao, J.: Pico: Contrastive label disambiguation for partial label learning. arXiv preprint arXiv:2201.08984 (2022)
46. Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., Bailey, J.: Symmetric cross entropy for robust learning with noisy labels. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 322–330 (2019)
47. Wei, J., Zhu, Z., Cheng, H., Liu, T., Niu, G., Liu, Y.: Learning with noisy labels revisited: A study using real-world human annotations. arXiv preprint arXiv:2110.12088 (2021)
48. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-ucsd birds 200 (2010)
49. Wu, D.D., Wang, D.B., Zhang, M.L.: Revisiting consistency regularization for deep partial label learning. In: International Conference on Machine Learning. pp. 24212–24225. PMLR (2022)
50. Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: 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)
51. Xu, N., Qiao, C., Geng, X., Zhang, M.L.: Instance-dependent partial label learning. Advances in Neural Information Processing Systems **34**, 27119–27130 (2021)
52. Yi, K., Wu, J.: Probabilistic end-to-end noise correction for learning with noisy labels. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7017–7025 (2019)
53. Zeng, Z., Xiao, S., Jia, K., Chan, T., Gao, S., Xu, D., Ma, Y.: Learning by associating ambiguously labeled images. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013. pp. 708–715. IEEE Computer Society (2013). https://doi.org/10.1109/CVPR.2013.97, https://doi.org/10.1109/CVPR.2013.97
54. Zhang, M., Zhou, B., Liu, X.: Partial label learning via feature-aware disambiguation. In: Krishnapuram, B., Shah, M., Smola, A.J., Aggarwal, C.C., Shen, D., Rastogi, R. (eds.) Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. pp. 1335–1344. ACM (2016). https://doi.org/10.1145/2939672.2939788, https://doi.org/10.1145/2939672.2939788
55. Zhang, Z., Liu, Z., Lu, H.: Partial label learning via cost-guided retraining. In: ECAI 2024, pp. 2170–2177. IOS Press (2024)
56. Zhou, X., Liu, X., Zhai, D., Jiang, J., Ji, X.: Asymmetric loss functions for noise-tolerant learning: Theory and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(7), 8094–8109 (2023)
57. Zhou, Z.H.: A brief introduction to weakly supervised learning. National science review **5**(1), 44–53 (2018)
58. Zhu, X., Goldberg, A.B.: Introduction to semi-supervised learning. Synthesis Lectures on Artificial Intelligence and Machine Learning (2009). https:

//doi.org/10.2200/S00196ED1V01Y200906AIM006, https://doi.org/10.2200/S00196ED1V01Y200906AIM006

59. Zhu, Z., Wang, J., Cheng, H., Liu, Y.: Unmasking and improving data credibility: A study with datasets for training harmless language models. arXiv preprint arXiv:2311.11202 (2023)

8 Discussion on the Analysis

8.1 Why Final FC Layer and Least-squares Classifier?

First, most classification networks decompose into a backbone, which extracts rich feature representations, and a task-specific head. The head is typically realized as a fully connected (FC) layer that maps backbone features to class logits. Crucially, this FC layer functions equivalently to a least-squares classifier: it selects weights to minimize $\|\mathbf{W}^\top \mathbf{x} - \mathbf{y}\|^2$, where $\mathbf{x}$ are the extracted features and $\mathbf{y}$ the one-hot labels. Since our goal is to understand how label noise perturbs the optimization and geometry of the classification weights, focusing on this FC head provides the most direct and analytically tractable setting.

Second, some works point that classification-relevant structure can emerge in intermediate layers [2]. However, conducting a full theoretical analysis across the entire deep network is generally intractable. Instead, we invoke the reasonable abstraction that each layer's weight matrix approximates an ideal linear mapping between its inputs and outputs. Under this abstraction, the learning objective for any given layer reduces to a least-squares problem of the same form as the final FC classifier. Thus, by performing a detailed analysis on the representative classification layer, we capture the core influence of noisy supervision on both learned representations and decision boundaries.

Finally, our framework naturally extends beyond the output layer. In practice, one could train two networks—one on clean labels and one on noisy labels—and then compare the SVD or other structural properties of corresponding weight matrices in intermediate blocks. This demonstrates the broad applicability of our theoretical approach to understanding and mitigating the effects of label noise throughout modern deep architectures.

8.2 Influence of Model Architecture Variations

The observed phenomena are largely independent of the specific network architecture. Our primary analysis focuses on the final fully connected layer, which serves as the mapping from extracted features to the label space. In contrast, the backbone architecture, whether convolutional or transformer-based, primarily contributes to feature extraction and does not directly affect the conclusions drawn from our analysis of classification weights.

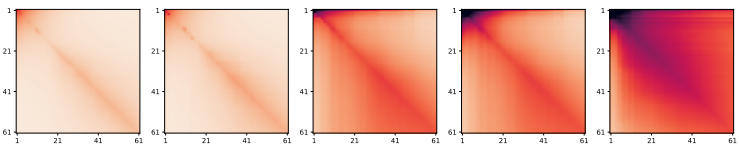

Fig. 5: Subspace similarity between top-i column vectors of $\mathbf{W}$ and top-j of $\mathbf{W}'$ with ViT-B backbone.

To further validate this claim, we conduct additional experiments by replacing the PRODEN backbone with ViT-B on CIFAR-100 and measuring subspace similarity under various noise levels $p = 0.05, 0.1, 0.2, 0.3, 0.4$. The resulting trends shown in Fig. 5 are consistent with those reported in Figure 2 of the main text, confirming that our conclusions hold across architectures.

Moreover, we evaluate ViT-B-based PRODEN and its variant augmented with LIP on CIFAR-100. The results are shown in Table. 7, where we can find that LIP continues to improve performance even under transformer-based settings, further reinforcing the generalizability and effectiveness of our theoretical framework and proposed method.

Table 7: Classification accuracy on real-world datasets.

Setting	$p = 0.05$	$p = 0.1$	$p = 0.2$
PRODEN	81.14	80.39	62.24
+LIP	81.89	81.07	63.13

8.3 Remarks

A natural question is whether our analysis enables estimation of the noise level threshold beyond which the model fails to acquire meaningful knowledge. Theoretically, as indicated by Eq. (9) in the main text, the perturbation of the learned subspace is governed by multiple factors, including the underlying data distribution, dataset size, and the label noise rate p. For a fixed dataset, one can approximate an acceptable range for p by observing the subspace deviation angle—empirically, we consider deviations within $\theta \leq 30^\circ$ to indicate tolerable distortion. Empirically, as shown in Fig. 2 in the main text, we find that when $p > 0.4$, the deviation of the principal subspace becomes notably severe. This suggests a rough upper bound on the label noise under which the model can still preserve the structural integrity of learned representations.

Importantly, a perturbed subspace does not necessarily imply complete failure of learning. As demonstrated in Table 2 in the main text, although model performance degrades significantly when $p > 0.4$—often with accuracy reduced by nearly half—certain discriminative capabilities persist. This indicates that even under substantial noise, the model retains partial alignment with the underlying task, albeit with diminished effectiveness.

In addition to the aforementioned analyses, we can draw several other conclusions or insights.

– The theoretical analysis allows us to explain a common trend observed in many studies [15, 26, 27, 49, 51]: as label inaccuracy increases, there is often a precipitous decline in performance. This occurs because when ρ is too large, the angle of the principal subspace may undergo significant changes,

resulting in a noticeable shift of the principal subspace. This shift entails a loss of crucial information for the task.
– We can posit that the reason why various information and multiple techniques such as regularization or data normalization aid in performance is that they can lower the upper bound of $\sin\theta$, thus preventing significant shifts in the principal subspace.

9 Experiment Details

9.1 Implementation Details

LIP has only one hyper-parameter, k. Empirically, setting k to $\lceil 0.8l \rceil$ yields superior performance. However, given the rapid execution speed of LIP, we can also conduct a grid search on k over the validation set to identify the optimal parameter, and then use the optimal k on the test set. Both PICO and PRODEN utilize ResNet-34 as their backbone architecture. For PRODEN, 500 iterations are conducted on both datasets. For PICO, 500 iterations are run on CUB-200 and 800 iterations on CIFAR-100. For CIFAR-100N, we use ResNet-50 as backbone and an SGD optimizer with a momentum of 0.9 for 300 epochs. All hyper-parameters for baselines are based on their papers. We use MATLAB and PyTorch [32] to implement all the methods. Ten runs of 50%/50% random train/test splits are performed on real-world datasets, and the average accuracy with standard deviation is represented. Five runs are performed with the the average accuracy and standard deviation recorded for CIFAR-100 and CUB-200.

9.2 Details on the Baselines

We provide the details on the baselines here.

– PL-CL [38]: uses the non-candidate labels to induce a complementary classifier to eliminate the false-positive labels [suggested configuration: k=10, λ=0.03, γ, μ, α, and $\beta \in \{0.001, 0.01, 0.1, 0.2, 0.5, 1, 1.5, 2, 4\}$];
– PL-CGR [55]: a cost-guided retraining strategy that guides and corrects disambiguation results while addressing instance-level class imbalance for candidate labels [suggested configuration: $\mu \in \{0.04, 0.05\}$, $\lambda \in \{0.4, 0.5, 0.6\}$, c=0.2];
– DPCLS [21]: leverages dissimilarity propagation to guide the shrinkage of candidate labels [suggested configuration: k=10, λ=0.05, β=0.001, $\alpha \in \{0.001, 0.01\}$];
– PL-AGGD [44]: a approach which constructs a similarity graph and further uses that graph to realize disambiguation [suggested configuration: k=10, T=20, λ=1, μ=1, γ=0.05];
– SURE [8]: a self-guided retraining based approach which maximizes an infinity norm regularization on the modeling outputs to realize disambiguation [suggested configuration: λ and $\beta \in \{0.001, 0.01, 0.05, 0.1, 0.3, 0.5, 1\}$];

Table 8: Characteristics of the real-world data sets.

Data set	#examples	#features	#labels	average #labels	Task Domain
FG-NET [31]	1002	262	78	7.48	facial age estimation
Lost [4]	1122	108	16	2.23	automatic face naming
MSRCv2 [25]	1758	48	23	3.16	object classification
Mirflickr [18]	2780	1536	14	2.76	web image classification
Soccer Player [53]	17472	279	171	2.09	automatic face naming
Yahoo!News [11]	22991	163	219	1.91	automatic face naming

- LALO [7]: an identification-based approach with constrained local consistency to differentiate the candidate labels [suggested configuration: k=10, λ=0.05, μ=0.005];
- PL-LEAF [54]: a *feature-aware disambiguation* based approach that utilizes the feature manifold to generate labeling confidence and perform multi-output regression to train the predictive model [suggested configuration: K=10, C_1=10, C_2=1];
- PICO [45]: a deep-learning framework which utilizes a contrastive learning module and a prototype-based label disambiguation algorithm to identify the ground truth;
- PRODEN [26]: a deep-learning framework which minimizes a proposed estimator of classification to update model and identify true labels.
- PLS [1]: a deep-learning framework that utilizes a state-of-the-art noise detection metric to identify noisy samples and estimate their true label through a consistency regularization method.
- AGCE [56]: a deep-learning framework which presents asymmetric loss functions, enabling the training of a noise-tolerant classifier using noisy labels, provided that clean labels are in the majority.

9.3 Details on the Real-world Data Sets

The characteristics of the real-world data sets in Table 1 in the main text are shown in Tab. 8.

Specifically, for facial age estimation task, the ages annotated by crowd-sourced labelers are considered as each human face's candidate labels. For automatic face naming task, each face scratched from a video or an image is presented as a sample while the names extracted from the corresponding titles or captions are its candidate labels. For object classification task, image segmentations are taken as instances with objects appearing in the same image as candidate labels.