PNP: Robust Learning from Noisy Labels by Probabilistic Noise Prediction

Zeren Sun[1], Fumin Shen[2], Dan Huang[3], Qiong Wang[1], Xiangbo Shu[1], Yazhou Yao[1]*, Jinhui Tang[1]
[1]Nanjing University of Science and Technology, Nanjing, China
[2]University of Electronic Science and Technology of China, Chengdu, China
[3]China Research and Development Academy of Machinery Equipment, Beijing, China

Abstract

*Label noise has been a practical challenge in deep learning due to the strong capability of deep neural networks in fitting all training data. Prior literature primarily resorts to sample selection methods for combating noisy labels. However, these approaches focus on dividing samples by order sorting or threshold selection, inevitably introducing hyper-parameters (*e.g.*, selection ratio / threshold) that are hard-to-tune and dataset-dependent. To this end, we propose a simple yet effective approach named PNP (**P**robabilistic **N**oise **P**rediction) to explicitly model label noise. Specifically, we simultaneously train two networks, in which one predicts the category label and the other predicts the noise type. By predicting label noise probabilistically, we identify noisy samples and adopt dedicated optimization objectives accordingly. Finally, we establish a joint loss for network update by unifying the classification loss, the auxiliary constraint loss, and the in-distribution consistency loss. Comprehensive experimental results on synthetic and real-world datasets demonstrate the superiority of our proposed method. The source code and models have been made available at* https://github.com/NUST-Machine-Intelligence-Laboratory/PNP.

1. Introduction

Although deep neural networks (DNNs) have attained impressive achievements, surpassing traditional methods in various vision tasks [3, 15, 28, 31, 38, 46], their requirement for large-scale high-quality human-labeled training samples (*e.g.*, ImageNet [4] and COCO [21]) can often pose a bottleneck when applied to real-world scenarios. Precise annotation is always labor-expensive and time-consuming, especially when domain-specific expert knowledge is necessary (*e.g.*, fine-grained visual categorization [13, 24, 41]). To alleviate this issue, one promising alternative is to resort to web images for training deep networks [19, 22, 34, 35, 39, 43, 45, 47–49, 53, 54]. However, noisy labels are inevitable in web images [34]. It has been demonstrated that noisy labels can impair the performance of deep networks since the over-parameterization equips DNNs with not only large learning capacities but also strong memorization power [16, 52]. Consequently, it is of great significance to develop robust models for learning from noisy labels.

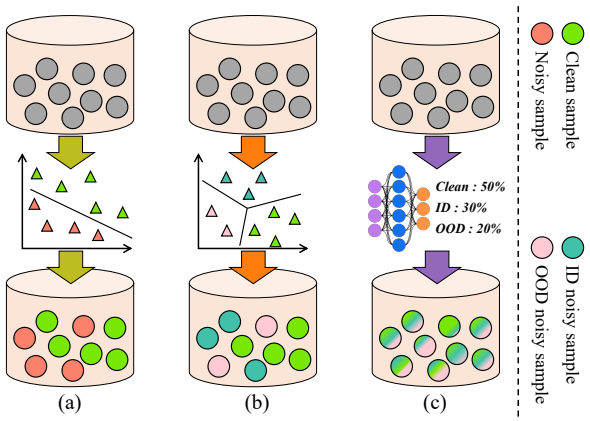

Figure 1. Early sample selection methods (**a**) tend to divide samples into two subsets (*i.e.*, clean and noisy), neglecting differences between ID and OOD noisy samples. Some recent methods (**b**) endeavor to identify clean, ID noisy, and OOD noisy samples. However, these methods usually suffer from tuning dataset-dependent threshold hyper-parameters (denoted as decision boundaries in (a) and (b)). In contrast, our proposed approach (**c**) seeks to model noisy labels in a probabilistic manner. PNP employs a dedicated predictor to estimate the probability distribution of noise type.

Noisy labels in real-world datasets can be categorized into two types: open-set and closed-set [48]. In the closed-set scenario, the true label of a noisy sample comes from a known label space $\mathcal{Y}_{known}$ present in the training data. Conversely, in the open-set scenario, true labels of samples are outside $\mathcal{Y}_{known}$. In other words, closed-set noisy samples are in-distribution (ID) ones, while open-set noisy samples are out-of-distribution (OOD) ones. A large body of prior literature primarily focuses on closed-set scenarios,

*Corresponding author.

assuming that only in-distribution noise exists. However, the ID-noise-only assumption may not hold true in real-world applications. Recently, an increasing number of researchers have been attracted to the open-set noisy problem, which is also the primary focus of this work.

There are mainly two common strategies to tackle noisy labels: loss correction [5, 9, 26, 29, 33, 39, 50, 56] and sample selection [1, 6, 11, 25, 42, 51]. Classic loss correction methods either attempt to estimate the noise transition matrix [2, 5, 9, 26, 33] or seek to regularize losses based on network predictions [29, 56]. Unfortunately, the noise transition matrix is challenging to estimate, while prediction-based loss correction suffers from error accumulation.

Sample selection methods essentially follow an intuitive but straightforward idea: eliminating noisy data and training with the cleaner subset. Researchers have recently witnessed that deep networks tend to fit clean and simple patterns before memorizing noisy labels [16, 52]. Accordingly, many approaches have been proposed to exploit this observation and regard low-loss samples as clean ones. For example, Co-teaching [6] maintains two networks simultaneously and enables them to select low-loss samples for their peer networks. Early sample selection methods usually split samples into two subsets: clean and noisy, neglecting the difference between in-distribution noisy and out-of-distribution noisy labels. More recently, CRSSC [34] and Jo-SRC [48] are proposed to divide samples into three groups: clean ones, in-distribution noisy ones, and out-of-distribution noisy ones, and treat them differently. The former employs a two-step sample selection process to categorize samples into three groups, while the latter proposes global sample selection criteria to distinguish different types of noise. Despite that promising results have been observed, existing methods inevitably involve hard-to-tune and dataset-dependent threshold hyper-parameters for selecting samples, posing a limit to the reliability and scalability of these methods in various larger real-world scenarios.

To address aforementioned issues, we propose a simple yet effective approach, named PNP (**P**robabilistic **N**oise **P**rediction), to probabilistically model label noise in an end-to-end manner. Specifically, we simultaneously train two networks, in which one (*i.e.*, label predictor network) predicts the category of the input data while the other (*i.e.*, noise predictor network) predicts the noise type (*i.e.*, clean / ID noisy / OOD noisy). The clean, ID noisy, OOD noisy samples can be naturally identified according to the prediction from the noise predictor network. To enable effective learning of the noise predictor network, we propose to optimize it in a regression manner, using JS divergence between prediction-label pairs and prediction-prediction pairs. Finally, we impose a consistency regularization on in-distribution data to further advance the learning of our label predictor network and noise predictor network. A comparison between our PNP and existing sample selection methods is visualized in Fig. 1. Our major contributions are:

(1) We propose a simple yet effective approach, named PNP, to combat noisy labels. PNP simultaneously predicts the category label and noise type for all training samples. By adopting distinct loss functions for different samples, PNP can robustly learn from noisy training data.

(2) PNP employs an auxiliary regression loss for empowering the model to learn to predict the noise type of each sample. JS divergence between prediction-label pairs and prediction-prediction pairs is adopted to approximate the ground-truth noise type. Furthermore, consistency between different views of in-distribution data is encouraged to reinforce the recognition ability.

(3) We evaluate two paradigms of sample selection in our method: PNP-hard (hard selection) and PNP-soft (soft selection). We validate the effectiveness and superiority of our method by providing extensive experimental results on both synthetic and real-world noisy datasets. Moreover, comprehensive ablation studies are established to verify each component of our approach.

2. Related Works

Prior works on learning from noisy labels can be briefly categorized into three families:

Label. Early methods primarily focus on correcting corrupted labels. For example, F-correction [26] proposes to adopt a two-step method for estimating the noise transition matrix. S-model [5] proposes to adopt an additional softmax layer to model the noise transition matrix. For these approaches, a well-estimated noise transition matrix is critical in achieving superior and robust performance. However, the noise transition matrix is difficult to estimate, especially in complicated scenarios (*e.g.*, real-world noisy datasets).

Sample. From the perspective of sample, the core idea is to perform sample re-weighting or sample selection. Sample re-weighting methods primarily seek to assign different weights to training samples. For example, Ren *et al*. [30] propose a meta-learning algorithm to weight training data differently. However, this line of work tends to involve a complicated optimization process and require a small set of clean validation data. Different from sample re-weighting, sample selection methods aim to select correctly-labeled samples for training. Researchers have demonstrated that low-loss samples are more likely to possess correct labels. For example, Co-teaching [6] trains two networks and lets them select low-loss samples for each other. JoCoR [42] employs a joint loss to select low-loss data, encouraging agreement between networks. CRSSC [34] adopts a loss-based selection and a confidence-based selection to identify clean, ID noisy, and OOD noisy samples. Jo-SRC [48] exploits the Jensen-Shannon (JS) divergence and prediction disagreement to globally select different types of noisy data.

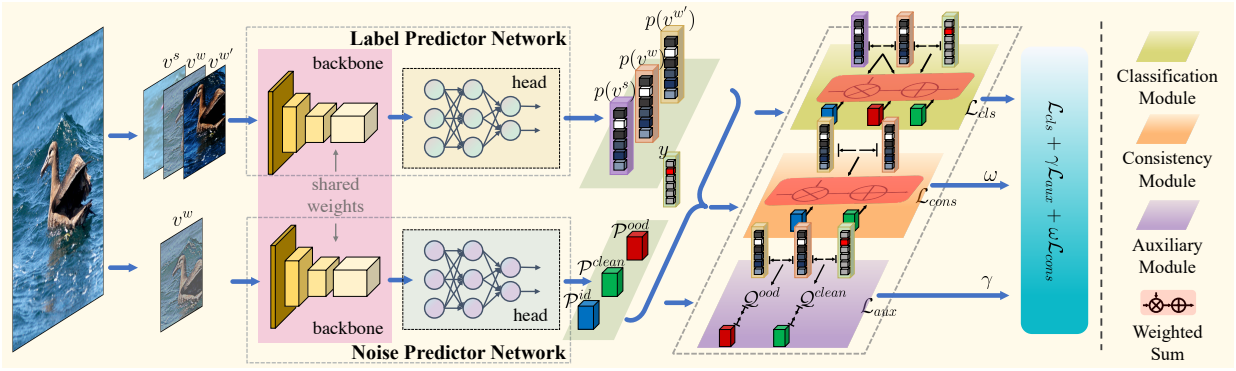

Figure 2. The overall framework of PNP. Each input image x_i is fed into two networks in parallel. The noise predictor network accordingly predicts the probability of x_i being clean ($\mathcal{P}^{clean}$) / ID ($\mathcal{P}^{id}$) / OOD ($\mathcal{P}^{ood}$). Meanwhile, x_i is augmented into two weakly and one strongly augmented views before fed into the label predictor network, leading to three label predictions $p(v_i^w)$, $p(v_i^{w'})$, and $p(v_i^s)$. Afterward, the classification loss $\mathcal{L}_{cls}$ is computed based on the estimated noise type and the selection paradigm (hard / soft) in the classification module. The constraint loss $\mathcal{L}_{aux}$ is attained resorting to the approximated ground-truth noise type in the auxiliary module. The consistency loss $\mathcal{L}_{cons}$ is obtained by encouraging the (label) prediction agreement between different views of in-distribution samples in the consistency module. Finally, our model is updated by back-propagating a joint loss, which is essentially a weighted sum of the above three losses.

Loss. Besides above two types of methods, some existing works concentrate on employing robust loss functions [23, 29, 40, 56]. For example, the bootstrapping loss [29] adds a perceptual loss term to the conventional classification loss. GCE [56] integrates the mean absolute loss and the cross-entropy loss. However, these methods tend to yield unsatisfactory performance in real-world cases.

3. The Proposed Method

Preliminaries. Given a N-sample C-class dataset $\mathcal{D} = \{(x_i, y_i)|1 \leq i \leq N\}$, in which x_i denotes th i-th training sample and $y_i \in \{0, 1\}^C$ is its annotated label. We denote the true label of x_i as y_i^*. Conventionally, we implicitly assume all annotated labels are accurate (*i.e.*, $y_i = y_i^*$) and thus optimize the model by minimizing the empirical loss

$$\mathcal{L} = \mathbb{E}_{\mathcal{D}}[l_{ce}(x_i, y_i)] = \frac{1}{N}\sum_{i=1}^{N} l_{ce}(x_i, y_i), \tag{1}$$

in which

$$l_{ce}(x_i, y_i) = -\sum_{c=1}^{C} y_i^c log(p^c(x_i, \Theta)). \tag{2}$$

Θ denotes the model parameters. $p^c(x_i, \Theta)$ denotes the predicted softmax probability of the i-th training sample x_i over its c-th class. (For simplicity, we use the notation p_i^c hereinafter.) Nevertheless, the clean-label assumption may be too restrictive for real-world scenarios and noisy labels are inevitable in many real-world datasets. In this paper, we focus on the scenario where annotated labels are not guaranteed to be correct. Due to the memorization effect [16], noisy labels are prone to leading to inferior performance when used for network training. Thus, it is urgent to design noise-robust methods for addressing noisy labels.

3.1. Probabilistic Noise Modeling

One of the most common strategies for tackling label noise is to find clean samples based on a pre-designed selection process. Owing to the behavior of DNNs in learning simple patterns before fitting noisy labels, previous works have witnessed promising results by selecting low-loss samples as clean ones. However, these methods tend to involve complicated hyper-parameters tuning. For example, Co-teaching [6] and JoCoR [42] require to estimate the noise ratio, while CRSSC [34] and Jo-SRC [48] need to choose a proper selection threshold. Unfortunately, these hyper-parameters (*e.g.*, noise ratio and selection threshold) are usually hard to tune and dataset-dependent.

To alleviate the aforementioned issue, we propose to directly model label noise in an end-to-end probabilistic manner. Specifically, we propose to train two parallel networks. The first network, termed as label predictor network (LPN), is trained to predict the category label:

$$p(x_i) = \sigma(h(f(x_i, \Psi), \Phi_L)) \in \mathbb{R}^C, \tag{3}$$

where Φ_L denotes parameters of the prediction head of LPN. Ψ denotes parameters of the backbone. $f(\cdot, \Psi)$ and $h(\cdot, \Phi_L)$ are mapping functions of the backbone and the prediction head. $\sigma(\cdot)$ is the softmax function. Conversely, the second network, termed as noise predictor network (NPN), is trained to predict the noise type:

$$t(x_i) = \sigma(g(f(x_i, \Psi), \Phi_N)) \in \mathbb{R}^3, \tag{4}$$

in which Φ_N denotes parameters of the prediction head of NPN. $g(\cdot, \Phi_N)$ is the mapping function of this prediction head. In our implementation, the prediction head of NPN is a multi-layer perception (MLP) network with one hidden layer. Here, we define $t^{(0)}(x_i)$, $t^{(1)}(x_i)$, and $t^{(2)}(x_i)$ as the likelihood of x_i belonging to the clean, ID, and OOD set, respectively. For simplicity, we henceforward denote $\mathcal{P}_i^{clean} = t^{(0)}(x_i)$, $\mathcal{P}_i^{id} = t^{(1)}(x_i)$, and $\mathcal{P}_i^{ood} = t^{(2)}(x_i)$. It should be noted that, in our implementation, to reduce the resource consumption and enable an end-to-end joint optimization, LPN and NPN share the same backbone feature extractor but differ in their prediction heads. By probabilistically modeling label noise, we can conveniently identify and tackle different types of noisy samples accordingly.

3.2. Classification Losses for Different Noise

The NPN predicts the noise type of each sample by estimating its "likelihood" of being clean / ID / OOD. We adopt different loss functions for different types of noisy samples. For clean samples, we employ the cross-entropy loss along with an entropy regularization term:

$$l_{clean}(x_i, y_i) = -\sum_{c=1}^{C} y_i^c log(p_i^c) - \sum_{c=1}^{C} p_i^c log(p_i^c), \quad (5)$$

For in-distribution / out-of-distribution noisy samples, inspired by unsupervised consistency training [44], we propose to treat outputs of strongly and weakly augmented inputs as predictions and targets, respectively. More specifically, for an ID noisy sample x_i, we feed its two augmented views (*i.e.*, a strongly augmented one v_i^s and a weakly augmented one v_i^w) into our network. The LPN accordingly produces predictions $p(v_i^s)$ and $p(v_i^w)$, which are then leveraged to compute the cross-entropy loss:

$$l_{id}(x_i) = l_{ce}(p(v_i^s), \varepsilon(p(v_i^w), \tau)), \quad (6)$$

in which

$$\varepsilon(z, T) = \frac{exp(z/T)}{\sum_{z^*} exp(z^*/T)}. \quad (7)$$

Similarly, for an OOD noisy sample x_i, we also employ its two augmented views for computing the classification loss:

$$l_{ood}(x_i) = l_{ce}(p(v_i^s), \varepsilon(p(v_i^w), 1/\tau)). \quad (8)$$

Here, inspired by Jo-SRC [48], we empirically set $\tau = 0.1$, making $\varepsilon(\cdot, \cdot)$ a sharpening operation in Eq. (6) but a flattening operation in Eq. (8).

Discussion. The motivation of employing Eqs. (6) and (8) for noisy samples is three-folded. Firstly, by optimizing losses computed from Eqs. (6) and (8), we implicitly enhance consistency between strongly and weakly augmented views of each noisy sample, leading to a smoother model and an improved sample efficiency. Secondly, strong augmentations tend to provide more diverse and natural views, benefiting the generalization performance. Lastly, although first terms (*i.e.*, predictions) in l_{ce} are identical between Eqs. (6) and (8), the second terms (*i.e.*, targets) are constructed distinctively based on the nature of ID and OOD noisy samples. For ID noisy samples, predictions from a well-trained model tend to be more reliable than given annotations. Therefore, we employ a sharpening operation to advance training by enforcing more confident predictions. On the contrary, OOD noisy samples usually confuse models due to their out-of-task ground-truth categories. By imposing a flattening operation, their predictions will fit an approximately uniform distribution, leading to a boosted robustness and generalization performance.

3.3. Constraint of Probabilistic Noise Modeling

We propose to train an additional predictor (*i.e.*, NPN) for estimating the noise type of each sample. However, the NPN is difficult to optimize due to the absence of ground-truth supervision. In this work, we propose to approximate the ground-truth noise type for each sample and accordingly train the NPN. Specifically, we follow Jo-SRC [48] and adopt the Jensen-Shannon (JS) divergence [20] to approximate the probability $\mathcal{Q}^{clean}$ of a sample x_i being clean:

$$\mathcal{Q}_i^{clean} = \mathcal{Q}^{clean}(x_i) = 1 - D_{JS}(p(v_i^w) \| y_i), \quad (9)$$

where $D_{JS}(\cdot \| \cdot)$ is the JS divergence function. Moreover, inspired by [48], we employ prediction divergence to estimate the "likelyhood" $\mathcal{Q}^{ood}$ of a sample being OOD. Different from [48], to enable a smoother optimization, we design

$$\mathcal{Q}_i^{ood} = \mathcal{Q}^{ood}(x_i) = D_{JS}(p(v_i^w) \| p(v_i^{w'})), \quad (10)$$

in which $v_i^{w'}$ denotes another weakly augmented view of x_i.

Once approximations of the ground-truth noise type are obtained, the following auxiliary constraint loss is adopted to optimize the NPN:

$$l_{aux}(x_i) = |\mathcal{P}_i^{clean} - \mathcal{Q}_i^{clean}| + |\mathcal{P}_i^{ood} - \mathcal{Q}_i^{ood}|. \quad (11)$$

Discussion. (1) Although Eq. (11) only provides a weak constraint due to the ground-truth approximation, the optimization of this auxiliary loss drives the estimation of noise type to its correct direction. (2) The optimization of Eq. (11) is actually a regression task. Therefore, the loss function could be any applicable regression loss (*e.g.*, Mean Absolute Error, Mean Squared Error, *etc.*). For simplicity, we empirically employ Mean Absolute Error (MAE) loss in our implementation. (3) Jo-SRC [48] uses prediction disagreement to measure the prediction divergence, producing a $0/1$ "likelyhood". Conversely, we employ the JS divergence to estimate the prediction disagreement so that our NPN can be optimized in a smoother manner.

3.4. Consistency of In-distribution Data

Intuitively, a well-trained model should predict consistently on different variations of in-distribution samples but contradictorily on those of out-of-distribution data. Due to the employment of prediction divergence in detecting out-of-distribution samples, we propose to impose a consistency regularization loss (*i.e.*, Eq. (12)) on in-distribution data.

$$l_{cons}(x_i) = D(p(v_i^w) \| p(v_i^{w'})) + D(p(v_i^{w'}) \| p(v_i^w)). \quad (12)$$

$D(\cdot\|\cdot)$ denotes the Kullback-Leibler (KL) divergence. The consistency regularization not only implicitly enhances representation learning, but also explicitly empowers our model to better discriminate ID noise and OOD noise.

3.5. PNP-hard and PNP-soft

The overall workflow of our PNP method is shown in Fig. 2. Our algorithm is trained in a two-step manner. Starting with a warm-up period, our network is trained with the original noisy labels by optimizing Eq. (1) for a few epochs. This step facilitates us with a reasonable model for subsequent robust learning. After the warm-up step, we start our PNP training by optimizing the following objective loss function in an end-to-end manner:

$$\mathcal{L} = \mathcal{L}_{cls} + \gamma \mathcal{L}_{aux} + \omega \mathcal{L}_{cons}, \quad (13)$$

where γ and ω are designed to balance different loss terms.

In this work, we evaluate two paradigms of sample selection: hard selection and soft selection. Following the idea of hard sample selection [34, 48], PNP-hard employs different loss functions on different types of samples:

$$\begin{cases} \mathcal{L}_{cls} = \mathbb{E}_{\mathcal{D}}[\mathbb{1}_{\mathcal{P}_i^{clean} \geq \max(\mathcal{P}_i^{id}, \mathcal{P}_i^{ood})} l_{clean}(x_i, y_i) \\ \qquad + \mathbb{1}_{\mathcal{P}_i^{id} > \max(\mathcal{P}_i^{clean}, \mathcal{P}_i^{ood})} l_{id}(x_i) \\ \qquad + \mathbb{1}_{\mathcal{P}_i^{ood} > \max(\mathcal{P}_i^{id}, \mathcal{P}_i^{clean})} l_{ood}(x_i)], \\ \mathcal{L}_{cons} = \mathbb{E}_{\mathcal{D}}[\mathbb{1}_{\mathcal{P}_i^{ood} < \max(\mathcal{P}_i^{id}, \mathcal{P}_i^{clean})} l_{cons}(x_i)]. \end{cases} \quad (14)$$

$\mathbb{1}_A$ is a indicator function, which equals 1 if A is true, and 0 otherwise. Contrarily, PNP-soft adopts soft sample selection, re-weighting losses based on predictions of noise type:

$$\begin{cases} \mathcal{L}_{cls} = \mathbb{E}_{\mathcal{D}}[\mathcal{P}_i^{clean} l_{clean}(x_i, y_i) \\ \qquad + \mathcal{P}_i^{id} l_{id}(x_i) + \mathcal{P}_i^{ood} l_{ood}(x_i)], \\ \mathcal{L}_{cons} = \mathbb{E}_{\mathcal{D}}[(\mathcal{P}_i^{clean} + \mathcal{P}_i^{id}) l_{cons}(x_i)]. \end{cases} \quad (15)$$

Comparison between PNP-hard and PNP-soft. PNP-hard is intuitive and straightforward, assigning each sample a discrete tag that reveals its noise type. Different loss functions are accordingly employed based on the estimated noise type. On the contrary, PNP-soft adopts a re-weighting schema when computing losses. While hard selection can concretely identify the noise type, it may amplify the risk of incorrect predictions, leading to a potential overfitting problem. Conversely, PNP-soft is beneficial by guaranteeing that at least part of the loss is correctly optimized even if the noise type is wrongly predicted. However, PNP-soft may suffer from the underfitting issue. Empirically, PNP-hard achieves better performance if the noise situation is insignificant and a trustworthy NPN can be attained. When the training data is heavily corrupted, PNP-soft would be superior, owing to its robustness against errors from NPN.

4. Experiments

4.1. Experiment Setup

Datasets. We evaluate our PNP approach on two synthetic datasets (*i.e.*, CIFAR100N and CIFAR80N) and four real-world datasets (*i.e.*, Web-Aircraft, Web-Bird, Web-Car, and Food101N). CIFAR100N and CIFAR80N stem from CIFAR100 [14]. Specifically, we follow Jo-SRC [48] to create the closed-set noisy dataset CIFAR100N and the open-set noisy dataset CIFAR80N. We adopt two classic noise structures: symmetric and asymmetric. Web-Aircraft, Web-Bird, and Web-Car are sub-datasets of WebFG-496 [36], which is a webly supervised fine-grained datasets. Food101N [17] is large-scale real-world noisy dataset.

Evaluation Metric. For assessing the performance of our proposed PNP approach, we adopt the test accuracy as our evaluation metric. Reported results are averaged performance of five repeated experiments under identical settings.

Implementation Details. We adopt a seven-layer DNN [48] for CIFAR100N and CIFAR80N. Adam optimizer [12] is employed during training. We set the initial learning rate as 0.001 and the batch size as 128. We warmup the network for 10 epochs. The learning rate starts to decay linearly after 80 epochs of training. The entire training lasts for 200 epochs. For obtaining further performance gains, we adopt the label smoothing regularization (LSR) [38] technique when calculating clean samples' classification losses (*i.e.*, Eq. (5)). The LSR parameter ϵ is empirically set to 0.6. γ and ω are set as 1.0 in default. For Web-Aircraft, Web-Bird, and Web-Car, we leverage ResNet-50 [8] pre-trained on ImageNet as our backbone to compare PNP with other state-of-the-art methods. We update network parameters using SGD optimizer [37] with a momentum of 0.9. The initial learning rate and batch size are 0.0005 and 16, respectively. The warm-up stage lasts for 10 epochs and we train networks for 120 epochs. We start decay learning rate after 10 epochs in a cosine annealing manner. γ and ω are also set as 1.0 in default. For Food101N, we follow settings in Jo-SRC [48] and employ pre-trained ResNet-50 for comparison. Default values of γ and ω are 1.0 and 0.2.

Methods	Publication	CIFAR100N			CIFAR80N		
		Sym − 20%	*Sym* − 80%	*Asym* − 40%	*Sym* − 20%	*Sym* − 80%	*Asym* − 40%
Standard	-	35.14 ± 0.44	4.41 ± 0.14	27.29 ± 0.25	29.37 ± 0.09	4.20 ± 0.07	22.25 ± 0.08
Decoupling [25]	NeurIPS 2017	33.10 ± 0.12	3.89 ± 0.16	26.11 ± 0.39	43.49 ± 0.39	10.01 ± 0.29	33.74 ± 0.26
Co-teaching [6]	NeurIPS 2018	43.73 ± 0.16	15.15 ± 0.46	28.35 ± 0.25	60.38 ± 0.22	16.59 ± 0.27	42.42 ± 0.30
Co-teaching+ [51]	ICML 2019	49.27 ± 0.03	13.44 ± 0.37	33.62 ± 0.39	53.97 ± 0.26	12.29 ± 0.09	43.01 ± 0.59
JoCoR [42]	CVPR 2020	53.01 ± 0.04	15.49 ± 0.98	32.70 ± 0.35	59.99 ± 0.13	12.85 ± 0.05	39.37 ± 0.16
Jo-SRC [48]	CVPR 2021	58.15 ± 0.14	23.80 ± 0.05	38.52 ± 0.20	65.83 ± 0.13	29.76 ± 0.09	53.03 ± 0.25
PNP-hard	-	**64.25** ± 0.12	**30.26** ± 0.15	**56.01** ± 0.31	**65.87** ± 0.23	**30.79** ± 0.16	**56.17** ± 0.42
PNP-soft	-	**63.27** ± 0.14	**31.32** ± 0.19	**60.25** ± 0.21	**67.00** ± 0.18	**34.36** ± 0.18	**61.23** ± 0.17

Table 1. Average test accuracy (%) on CIFAR100N and CIFAR80N over the last 10 epochs ("*Sym*" and "*Asym*" denote the symmetric and asymmetric label noise, respectively).

Baselines. To evaluate our PNP approach on synthetic datasets, we follow Jo-SRC [48] and compare PNP-hard / PNP-soft with state-of-the-art sample selection methods: Decoupling [25], Co-teaching [6], Co-teaching+ [51], JoCoR [42], and Jo-SRC [48]. For evaluating on Web-Aircraft, Web-Bird, and Web-Car, we additionally compare PNP with other state-of-the-art methods (*e.g.*, SELFIE [32], PENCIL [50], AFM [27], CRSSC [34], Self-adaptive [10], DivideMix [18], PLC [55], and Peer-learning [36]). We follow Jo-SRC [48] when evaluating our approach on Food101N. We compare our approach with CleanNet [17], DeepSelf [7], and Jo-SRC [48]. Finally, we denote "Standard" as the baseline case in which we train a deep network using noisy datasets directly. We implement all above methods using PyTorch for performing fair comparison.

4.2. Evaluation on Synthetic Noisy Datasets

We first evaluate PNP on synthetic datasets. By varying the structure and ratio of label noise, we can better understand the effectiveness of PNP in different noise situations.

Results on CIFAR100N. Starting from evaluating our approach in closed-set scenarios, we present the comparison in test accuracy with state-of-the-art approaches on CIFAR100N in Tab. 1. Results of existing methods are drawn from Jo-SRC [48] and those of our method are obtained under the same experimental settings. From Tab. 1, we can observe that both PNP-hard and PNP-soft consistently achieve the leading performance. While existing state-of-the-art approaches almost fail in the most inferior case (*i.e.*, *Sym*-80%), our PNP-hard and PNP-soft still achieve the most appealing performances. We can observe that PNP-hard outperforms PNP-soft only when the noise structure and ratio is *Sym*-20%. This verifies our argument that hard selection (PNP-hard) will achieve better results only when the noise situation is insignificant. In other cases, PNP-soft consistently performs better than PNP-hard. It should be noted that real-world noisy labels are mostly asymmetric. Tab. 1 reveals that our PNP-hard / PNP-soft performs notably better than state-of-the-art methods in the case of *Asym*-40%. The remarkable superiority of our method in asymmetric noise indicates that PNP will achieve satisfactory results in real-world noisy datasets.

Results on CIFAR80N. CIFAR80N is specifically created to simulate the real-world (open-set) noisy scenario. The comparison between our method with state-of-the-art approaches is also provided in Tab. 1. Results of existing methods are directly from Jo-SRC [48], and performances of our method are obtained under the same experimental settings. From Tab. 1, we can have the following observations: (1) Our PNP-hard / PNP-soft method consistently outperforms state-of-the-art approaches across different noise scenarios. Our model can achieve the best performance even when facing severe label noise (*i.e.*, *Sym*-80%). (2) PNP-soft exhibits better performance than PNP-hard in all noisy cases. We believe this results from the complicated noisy labels existed in the open-set noisy dataset CIFAR80N. (3) PNP-hard and PNP-soft obtain impressive performance boost in the case of *Asym*-40%, validating our design for open-set real-world (asymmetric) problems. These observations firmly validate the effectiveness and superiority of our proposed method in open-set noisy cases.

4.3. Evaluation on Real-world Noisy Datasets

Beyond the above evaluations, we conduct experiments on real-world noisy datasets, including three medium-scale web-image-based fine-grained datasets and one large-scale food dataset, to verify the effectiveness of PNP.

Results on Web-Aircraft / Bird / Car. Web-Aircraft, Web-Bird, and Web-Car are three real-world web image datasets for fine-grained vision categorization. Within each dataset, more than 25% of training samples are associated with unknown (asymmetric) noisy labels. Even worse, these datasets do not provide any label verification information, making it a practical and challenging label noise problem. Tab. 2 illustrates a comparison between our method with state-of-the-art methods. From this table, the leading performance obtained by our method can be witnessed. PNP-hard and PNP-soft both outperform state-of-

Methods	Publications	Backbone	Performances (%)		
			Web-Aircraft	Web-Bird	Web-Car
Standard	-	ResNet50	60.80	64.40	60.60
Decoupling [25]	NeurIPS 2017	ResNet50	75.91	71.61	79.41
Co-teaching [6]	NeurIPS 2018	ResNet50	79.54	76.68	84.95
Co-teaching+ [51]	ICML 2019	ResNet50	74.80	70.12	76.77
SELFIE [32]	ICML 2019	ResNet50	79.27	77.20	82.90
PENCIL [50]	CVPR 2019	ResNet50	78.82	75.09	81.68
JoCoR [42]	CVPR 2020	ResNet50	80.11	79.19	85.10
AFM [27]	ECCV 2020	ResNet50	81.04	76.35	83.48
CRSSC [34]	ACM MM 2020	ResNet50	82.51	81.31	87.68
Self-adaptive [10]	NeurIPS 2020	ResNet50	77.92	78.49	78.19
DivideMix [18]	ICLR 2020	ResNet50	82.48	74.40	84.27
Jo-SRC [48]	CVPR 2021	ResNet50	82.73	81.22	88.13
PLC [55]	ICLR 2021	ResNet50	79.24	76.22	81.87
Peer-learning [36]	ICCV 2021	ResNet50	78.64	75.37	82.48
PNP-hard	-	ResNet50	**85.03**	81.20	**89.93**
PNP-soft	-	ResNet50	**85.54**	**81.93**	**90.11**

Table 2. Comparison with state-of-the-art approaches in test accuracy (%) on Web-Aircraft, Web-Bird, and Web-Car.

Method	Backbone	Test accuracy
Stardard	ResNet-50	84.51
CleanNet ω_{hard} [17]	ResNet-50	83.47
CleanNet ω_{soft} [17]	ResNet-50	83.95
DeepSelf [7]	ResNet-50	85.11
Jo-SRC [48]	ResNet-50	86.66
PNP-hard	ResNet-50	**87.31**
PNP-soft	ResNet-50	**87.50**

Table 3. Comparison with state-of-the-art approaches in test accuracy (%) on Food101N.

Clean	ID	OOD	AUX	CONS	PNP	
					hard	soft
✓					42.10	47.13
✓	✓				49.34	52.11
✓	✓	✓			50.69	54.09
✓	✓	✓	✓		52.90	57.35
✓	✓	✓		✓	51.30	60.20
✓	✓	✓	✓	✓	58.54	62.18

Table 4. Impacts of different ingredients in test accuracy (%) on CIFAR80N (*Asym*-40%). Results at the best epochs are presented.

the-art methods on Web-Aircraft and Web-Car by a considerably large margin (2.30% / 2.81% on Web-Aircraft and 1.80% / 1.98% on Web-Car). Although PNP-hard achieves a slightly lower result than CRSSC [34] and Jo-SRC [48] on the Web-Bird dataset, PNP-soft still exhibits the best performance. Besides the superior performance, PNP-soft consistently surpasses PNP-hard on all three datasets. This behavior once again confirms our argument that PNP-soft is more robust against complicated noisy labels than PNP-hard.

Results on Food101N. Food101N is another real-world noisy dataset, consisting of 101 different food categories and over 310k training samples. This dataset also contains a large proportion of noisy labels. Tab. 3 presents the experimental results of our methods compared with state-of-the-art approaches. As illustrated in Tab. 3, PNP-hard and PNP-soft both achieve superior test accuracy than existing methods, supporting our claim that PNP is effective in alleviating noisy labels in large-scale real-world applications.

4.4. Ablation Study

4.4.1 Influence of Different Ingredients

Tab. 4 illustrates impacts of different ingredients in PNP. Clean, ID, and OOD denote the adoption of Eq. (5), Eq. (6), and Eq. (8), respectively. AUX indicates that the constraint loss Eq. (11) is utilized. CONS suggests the employment of in-distribution consistency regularization. LSR is adopted in default. The best result of "Standard" is 29.11%, and that of "Standard + LSR" is 33.10%. From this table, we can observe that each ingredient exhibits a non-trivial significance in our approach. Firstly, by using NPN to identify clean / ID noisy / OOD noisy samples, the performance is promoted by a large margin compared to "Standard + LSR". Secondly, the employment of the auxiliary constraint empowers the model to achieve more performance boost. Lastly, through imposing consistency regularization on in-distribution data, PNP-hard and PNP-soft are further advanced in robustness.

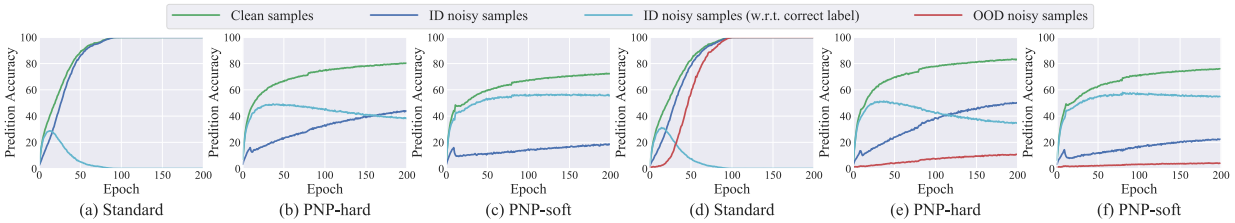

Figure 3. The prediction accuracy (%) of different types of training samples. (a)-(c) present results on CIFAR100N (*Asym*-40%). (d)-(f) provide comparison on CIFAR80N (*Asym*-40%)

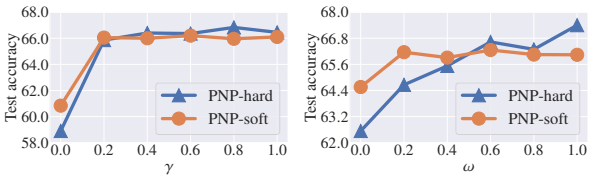

Figure 4. The sensitivity of hyper-parameter γ and ω. (Experiments are conducted on CIFAR80N with *Sym*-20% label noise.)

4.4.2 Prediction Accuracy over Different Samples

It has been established that the over-parameterized deep networks possess extraordinary learning capability and can memorize all training data. Therefore, the fundamental principle in learning from noisy labels is to hinder deep networks from overfitting noisy samples. Fig. 3 exhibits the prediction accuracy of different types of training samples (*i.e.*, clean, ID noisy, and OOD noisy samples). Besides, we additionally analyze the prediction accuracy of ID noisy samples w.r.t. their true labels. Fig. 3 (a) and (d) show cases of "Standard", in which networks eventually overfit noisy samples. The cyan curves (*i.e.*, ID noisy samples w.r.t. their true labels) justify the claim that deep networks tend to learn clean and simple patterns before overfitting noisy samples. From Fig. 3 (b)(c)(e)(f), we can observe that clean samples are fitted progressively during training, but the overfitting on noisy samples is significantly suppressed by adopting PNP-hard / PNP-soft. Despite lack of correct supervision, knowledge from ID noisy samples is still learned by our approach. By comparing results of PNP-hard and PNP-soft, we can find the latter has a stronger capability to hinder the network from overfitting noisy samples (especially for in-distribution noisy ones), certifying its superior robustness.

4.4.3 Sensitivity of Hyper-parameters

For studying sensitivity of hyper-parameters, we primarily investigate two parameters (*i.e.*, γ and ω) in the value range of $\{0.2, 0.4, 0.6, 0.8, 1.0\}$. Fig. 4 presents the results on the CIFAR80N (*Sym*-20%) dataset. For better understanding these two hyper-parameters, we additionally provide experimental results of $\gamma = 0$ and $\omega = 0$. The left sub-figure reveals that our method is considerably robust against the value of γ. The sharp performance increase from $\gamma = 0$ to $\gamma > 0$ demonstrates the importance of employing the auxiliary constraint loss (*i.e.*, Eq. (11)). The right sub-figure exhibits the sensitivity of ω. From this sub-figure, we can observe that while PNP-soft is fairly robust against ω, PNP-hard can benefit from the value increase of this hyper-parameter. Our hypothesis is that the superior noise-robustness of PNP-soft weakens the impact of ω. Since PNP-hard is less robust to label noise, a stronger consistency regularization may better boost the model performance. This sub-figure also reveals a notable performance gap between $\omega = 0$ and $\omega > 0$, manifesting the necessity of adopting in-distribution consistency regularization.

5. Conclusion

In this paper, we focused on the challenge of learning from real-world (open-set) noisy labels. To mitigate their negative impact, we proposed a simple yet effective approach named PNP to model label noise in an end-to-end probabilistic manner. PNP followed the sample selection paradigm but bypassed the requirement for selection thresholds, which were hard-to-tune and dataset-dependent. Specifically, PNP trained two networks in parallel, enabling simultaneous predictions of the category label (*i.e.*, LPN) and the noise type (*i.e.*, NPN). Moreover, a regression task was proposed to optimize the NPN and a consistency regularization was adopted to empower the discrimination ability. Finally, we evaluated two selection paradigms of PNP (*i.e.*, PNP-hard and PNP-soft). A series of experimental results on synthetic and real-world datasets justified the effectiveness and superiority of our proposed approach.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 62102182, 61976116, 61905114, 62072245, and 61932020), Natural Science Foundation of Jiangsu Province (No. BK20210327 and BK20211520), Fundamental Research Funds for the Central Universities (No. 30920021135), and National Key R&D Program of China (No. 2018AAA0102001 and 2021YFF0602101).

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *ICML*, pages 312–321, 2019. 2

[2] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *NeurIPS*, pages 1002–1012, 2017. 2

[3] Tao Chen, Guo-Sen Xie, Yazhou Yao, Qiong Wang, Fumin Shen, Zhenmin Tang, and Jian Zhang. Semantically meaningful class prototype learning for one-shot image segmentation. *IEEE TMM*, 24:968–980, 2022. 1

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 1

[5] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *ICLR*, 2017. 2

[6] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, pages 8527–8537, 2018. 2, 3, 6, 7

[7] Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *ICCV*, pages 5138–5147, 2019. 6, 7

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5

[9] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *NeurIPS*, pages 10456–10465, 2018. 2

[10] Lang Huang, Chao Zhang, and Hongyang Zhang. Self-adaptive training: beyond empirical risk minimization. In *NeurIPS*, volume 33, 2020. 6, 7

[11] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, pages 2309—2318, 2017. 2

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[13] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshops*, pages 554–561, 2013. 1

[14] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 1(4):7, 2009. 5

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012. 1

[16] David Krueger, Nicolas Ballas, Stanislaw Jastrzebski, Devansh Arpit, Maxinder S Kanwal, Tegan Maharaj, Emmanuel Bengio, Asja Fischer, Aaron Courville, Simon Lacoste-Julien, et al. A closer look at memorization in deep networks. In *ICML*, 2017. 1, 2, 3

[17] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *CVPR*, pages 5447–5456, 2018. 5, 6, 7

[18] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *ICLR*, 2020. 6, 7

[19] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Learning to learn from noisy labeled data. In *CVPR*, pages 5051–5059, 2019. 1

[20] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991. 4

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 1

[22] Huafeng Liu, Chuanyi Zhang, Yazhou Yao, Xiu-Shen Wei, Fumin Shen, Zhenmin Tang, and Jian Zhang. Exploiting web images for fine-grained visual recognition by eliminating open-set noise and utilizing hard examples. *IEEE TMM*, 24:546–557, 2022. 1

[23] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *ICML*, pages 6543–6553, 2020. 3

[24] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 1

[25] Eran Malach and Shai Shalev-Shwartz. Decoupling "when to update" from "how to update". In *NeurIPS*, pages 960–970, 2017. 2, 6, 7

[26] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, pages 1944–1952, 2017. 2

[27] Xiaojiang Peng, Kai Wang, Zhaoyang Zeng, Qing Li, Jianfei Yang, and Yu Qiao. Suppressing mislabeled data via grouping and self-attention. In *ECCV*, pages 786–802, 2020. 6, 7

[28] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, pages 7263–7271, 2017. 1

[29] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. In *ICLR*, 2015. 2, 3

[30] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, pages 4331–4340, 2018. 2

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015. 1

[32] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *ICML*, pages 5907–5915, 2019. 6, 7

[33] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. In *ICLR*, 2015. 2

[34] Zeren Sun, Xian-Sheng Hua, Yazhou Yao, Xiu-Shen Wei, Guosheng Hu, and Jian Zhang. Crssc: salvage reusable samples from noisy data for robust learning. In *ACM MM*, pages 92–101, 2020. 1, 2, 3, 5, 6, 7

[35] Zeren Sun, Huafeng Liu, Qiong Wang, Tianfei Zhou, Qi Wu, and Zhenmin Tang. Co-ldl: A co-training-based label distribution learning method for tackling label noise. *IEEE TMM*, 24:1093–1104, 2022. 1

[36] Zeren Sun, Yazhou Yao, Xiu-Shen Wei, Yongshun Zhang, Fumin Shen, Jianxin Wu, Jian Zhang, and Heng-Tao Shen. Webly supervised fine-grained recognition: Benchmark datasets and an approach. In *ICCV*, pages 10602–10611, 2021. 5, 6, 7

[37] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, pages 1139–1147, 2013. 5

[38] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 1, 5

[39] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, pages 5552–5560, 2018. 1, 2

[40] Sunil Thulasidasan, Tanmoy Bhattacharya, Jeff Bilmes, Gopinath Chennupati, and Jamal Mohd-Yusof. Combating label noise in deep learning using abstention. In *ICML*, pages 6234–6243, 2019. 3

[41] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. *CNS-TR-2011-001*, 2011. 1

[42] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *CVPR*, pages 13726–13735, 2020. 2, 3, 6, 7

[43] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, pages 2691–2699, 2015. 1

[44] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *NeurIPS*, volume 33, pages 6256–6268, 2020. 4

[45] Jufeng Yang, Xiaoxiao Sun, Yu-Kun Lai, Liang Zheng, and Ming-Ming Cheng. Recognition from web data: A progressive filtering approach. *IEEE TIP*, 27(11):5303–5315, 2018. 1

[46] Yazhou Yao, Tao Chen, Guo-Sen Xie, Chuanyi Zhang, Fumin Shen, Qi Wu, Zhenmin Tang, and Jian Zhang. Non-salient region object mining for weakly supervised semantic segmentation. In *CVPR*, pages 2623–2632, 2021. 1

[47] Yazhou Yao, Xiansheng Hua, Guanyu Gao, Zeren Sun, Zhibin Li, and Jian Zhang. Bridging the web data and fine-grained visual recognition via alleviating label noise and domain mismatch. In *ACM MM*, pages 1735–1744, 2020. 1

[48] Yazhou Yao, Zeren Sun, Chuanyi Zhang, Fumin Shen, Qi Wu, Jian Zhang, and Zhenmin Tang. Jo-src: A contrastive approach for combating noisy labels. In *CVPR*, pages 5192–5201, June 2021. 1, 2, 3, 4, 5, 6, 7

[49] Yazhou Yao, Jian Zhang, Fumin Shen, Xiansheng Hua, Jingsong Xu, and Zhenmin Tang. Exploiting web images for dataset construction: A domain robust approach. *IEEE TMM*, 19(8):1771–1784, 2017. 1

[50] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *CVPR*, pages 7017–7025, 2019. 2, 6, 7

[51] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *ICML*, pages 7164–7173, 2019. 2, 6, 7

[52] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017. 1, 2

[53] Chuanyi Zhang, Yazhou Yao, Huafeng Liu, Guo-Sen Xie, Xiangbo Shu, Tianfei Zhou, Zheng Zhang, Fumin Shen, and Zhenmin Tang. Web-supervised network with softly update-drop training for fine-grained visual classification. In *AAAI*, pages 12781–12788, 2020. 1

[54] Chuanyi Zhang, Yazhou Yao, Xing Xu, Jie Shao, Jingkuan Song, Zechao Li, and Zhenmin Tang. Extracting useful knowledge from noisy web images via data purification for fine-grained recognition. In *ACM MM*, pages 4063–4072, 2021. 1

[55] Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. Learning with feature-dependent label noise: A progressive approach. In *ICLR*, 2021. 6, 7

[56] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, pages 8778–8788, 2018. 2, 3