Identifying Hard Noise in Long-Tailed Sample Distribution

Xuanyu Yi[1], Kaihua Tang[1], Xian-Sheng Hua[2], Joo-Hwee Lim[3], Hanwang Zhang[1]
[1]Nanyang Technological University, [2]Damo Academy, Alibaba Group, [3]Institute for Infocomm Research, Singapore
xuanyu001@e.ntu.edu.sg, kaihua.tang@ntu.edu.sg, xshua@outlook.com
joohwee@i2r.a-star.edu.sg, hanwangzhang@ntu.edu.sg

Abstract

Conventional de-noising methods rely on the assumption that all samples are independent and identically distributed, so the resultant classifier, though disturbed by noise, can still easily identify the noises as the outliers of training distribution. However, the assumption is unrealistic in large-scale data that is inevitably long-tailed. Such imbalanced training data makes a classifier less discriminative for the tail classes, whose previously "easy" noises are now turned into "hard" ones—they are almost as outliers as the clean tail samples. We introduce this new challenge as Noisy Long-Tailed Classification (NLT). Not surprisingly, we find that most de-noising methods fail to identify the hard noises, resulting in significant performance drop on the three proposed NLT benchmarks: ImageNet-NLT, Animal10-NLT, and Food101-NLT. To this end, we design an iterative noisy learning framework called Hard-to-Easy (H2E). Our bootstrapping philosophy is to first learn a classifier as noise identifier *invariant* to the class and context distributional changes, reducing "hard" noises to "easy" ones, whose removal further improves the invariance. Experimental results show that our H2E outperforms state-of-the-art de-noising methods and their ablations on long-tailed settings while maintaining a stable performance on the conventional balanced settings. Datasets and codes are available at https://github.com/yxymessi/H2E-Framework.

1 Introduction

Any visual model should learn to co-exist with noise because any real-world dataset is imperfect [1]. During data collection, noise such as sensory failure (*e.g*, low-quality or corrupted images) and human error (*e.g*, mislabeling or ambiguous annotation) may hurt model training. In general, noise can be understood as a small population of training samples whose image contents differ from the ground-truth classes [2]. Therefore, if the data is independent and identically distributed (IID) regardless of class [3–5], noise samples can be identified as the outliers of the classifier confidence [6–8]. Specifically, we first learn the classifier on noisy data, then identify the noises as outliers, and finally remove them for a cleaner data that improves the classifier—a virtuous cycle [9, 10]. In particular, we term the noise that can be identified as outliers as "easy" noise.

On a dataset with the balanced number of diverse training samples per class—the conventional settings as in most de-noise literature [9, 11, 12]—the IID assumption is easy to be satisfied. The key reason is that such dataset can guarantee a robust classifier that only focuses on the context-invariant class feature (or causal feature) [13–16]. Therefore, after learning to exclude all the varying contexts (non-causal features), the class features of clean and noisy samples are indeed different. For example, even if the noise is as tricky as a "leopard" sample mislabeled as "cat" that is visually similar to "leopard", after removing the context, "cat" feature is still different from "leopard" feature, who is an "easy" outlier of "cat". So, the premise of the IID assumption is the disentanglement of the class and context features.

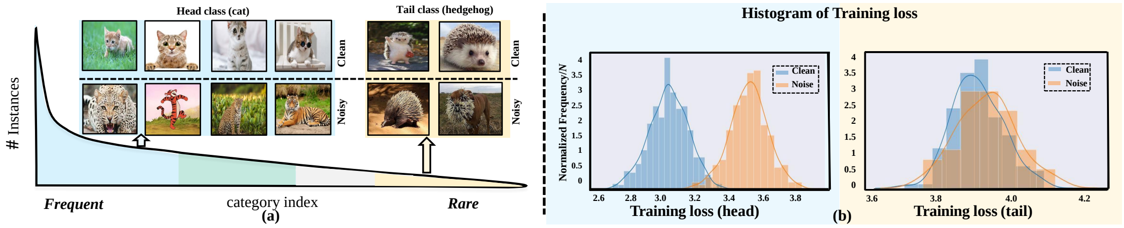


Figure 1: (a) Large-scale datasets are both long-tailed and noisy. For instance, a head category "cat" may contain noisy samples such as "leopard" and "cartoon tiger" while noise like "porcupine" and "spiny horse" in tail category "hedgehog" (b) The identification of noise based on classifier confidence (or training loss) is no longer applicable in tail classes for most de-noise algorithms

However, should a dataset be at scale, long-tailed distribution will be inevitable [17, 18], and thus disentangling class and context becomes challenging. The reasons are due to the following two folds that make the classifier dependent on **class prior** and **context distribution**. *First*, as head has more samples than tail, the classifier will be biased to head [19]. *Second*, head samples have more diverse contexts than tail, *i.e*, contexts are not shared by all the classes and some contexts are unique to certain tail classes due to sample scarcity. So, the resultant classifier fails to learn context-invariant class features, but entangling context with class [20]. As shown in Fig. 1(a), the "spine" context is highly correlated to "hedgehog" and thus "spine" is a confusing context to mis-recognize the noise "porcupine" and "spiny horse" as "hedgehog". Thus the long-tailed distribution will turn "easy" noise into "hard", especially for the tail classes. Fig. 1(b) illustrate such an example: the noises in tail class are almost as outliers as the tail samples. We leave a more detailed analysis in Section 3.

In this paper, we present a new challenge for noisy learning at scale, called Noisy Long-Tailed classification (NLT), which unifies the long-tailed distribution with realistic noisy data, completing the pioneering work with only synthetic noise on imbalanced data [21–23]. For rigorous and reproducible evaluations in the community, we introduce three benchmarks: ImageNet-NLT, Animal10-NLT, and Food101-NLT, with various noise and imbalance ratio for comprehensive diagnosis (Section 5). Not surprisingly, most of the existing de-noise methods degrade significantly on the benchmarks, especially for those who heavily rely on outlier detection [24, 10, 9, 25].

One may wonder if we could first learn a balanced classifier on the noisy data by using long-tailed classification methods [28, 19, 29], and then apply the conventional outlier detection for noise identification. The answer is "No" because those methods can only mitigate the class bias but not the context bias in long-tailed data. Fig. 2 demonstrates that with the increase of noise ratio, the performance of a SOTA long-tailed method [27] decreases significantly.

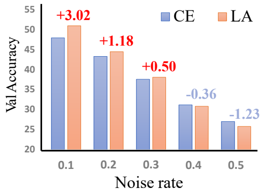


Figure 2: The comparison of CE (cross-entropy) [26] and Logit-Adjustment [27] in CIFAR-100 with different noise ratios.

To this end, we propose an iterative Hard-to-Easy (H2E) framework for NLT. It has two stages: 1) A noise identifier that is invariant to the class and context distributional change caused by long-tailed distribution (Section 4.1). Such invariance can reduce "hard" noises to "easy" ones. Specifically, we sample three data distribution: long-tailed, balanced, and reversed long-tailed, as three context environments, and then apply Invariant Risk Minimization (IRM) [13] to learn a long-tailed classifier as the noise identifier invariant to these environments. Note that this stage is iterative as "clearner" data improves training better backbones. 2) Thanks to the noise identifier, we can eventually learn a robust classifier.

Our contributions are summarized as follows:

- We present the task: Noisy Long-Tailed classification (NLT) with non-synthetic real-world noises. NLT is challenging because it turns "easy" noises into "hard" ones that cannot be identified by prior work.

- We propose a strong NLT baseline: Hard-to-Easy (H2E) noisy learning framework. The success of H2E is based on learning a noise identifier invariant to the class and context changes introduced by long-tailed data.
- We introduce three NLT benchmarks: ImageNet-NLT, Animal10-NLT, and Food101-NLT. Extensive experimental results on them show the limitations of existing de-noise method and the potential of learning invariance for noisy learning.

2 Related Work

Long-Tailed Classification. Most existing long-tailed methods can be categorized into three types: 1) class-wise re-balancing using re-sampling strategies [28, 29], re-weighted losses [30–32], and post-hoc adjustments [27, 19], 2) data augmentation [33, 34], and 3) model ensembling [35, 36]. Since the latter two aim to boost the overall performance by directly increasing the model capability, which is generally suitable for all classification tasks, rather than focusing on tackling the imbalance between head and tail, we mainly focus on the class-wise re-balancing methods in this paper. Besides, the performance of conventional long-tailed algorithms may significantly degrade in the noisy environment, as they assume the training samples to be annotated correctly, which is impractical in real-world images at scale.

Noisy Learning. The previous learning with noise algorithms can be summarized into 1) the noisy sample selection [24, 6, 9] and 2) the regularization [37–39]. Since the latter methods are generally applicable for all classification tasks, we mainly investigate the former in this paper, as they are more related to the proposed H2E framework. Most of the noisy sample selection methods filter out noisy samples by adopting the *small-loss trick*, which treats samples with small training losses as correct-annotated. In particular, Co-teaching [6] trains two networks simultaneously where each network selects small-loss samples in a mini-batch to train the other. Beta Mixture Model (BMM) [40] separates the clean and noise samples during training based on the loss value of each sample. Similarly, Divide-Mix [9] fits a Gaussian mixture model on per-sample loss distribution to divide the training samples into clean set and noisy set. However, in the existence of the class imbalance, most of these methods may not work well since the training loss converges easier in major classes than minor classes, resulting in the risk of discarding most samples in the minor classes.

3 Noisy Long-Tailed Classification

The classification of an image x as class c can be defined as predicting $p(y = c|x)$ based on a dataset of image and ground-truth label pairs $\{(x, y)\}$ [41], where the noise is caused by the wrong label assignment $\widetilde{y} \rightarrow x$ ($\widetilde{y} \neq y$). By Bayes theorem [42], we can decompose the predictive model as $p(y = c|x) = \frac{p(x|y=c) \cdot p(y=c)}{p(x)}$, where $p(y = c)$ is the class distribution, $p(x)$ is the marginal distribution of images. In the independent and identical distribution (IID) assumption of uniform $p(x)$ and $p(y = c)$, it is relatively easy to obtain an *ideal* noise identifier: the classifier $p(y = c|x)$ *per se*, which will be explained later.

Unfortunately, the IID assumption is not practical in general as large-scale dataset is usually imbalanced in not only class distribution, but also context distribution. We assume that any image x is generated by a set of hidden semantics $z = \{z_1, z_2, z_3, ...\}$, which includes two disjoint subsets: class-specific attributes z_c (*e.g*, the cat-like shape in the "cat" category) and context-specific environmental attributes z_e (*e.g*, the fur color). So, we can further decompose the predictive model $p(y = c|x = (z_c, z_e))$ as follows:

$$p(y = c|z_c, z_e) = \frac{p(z_c|y = c)}{p(z_c, z_e)} \cdot \overbrace{p(z_e|y = c, z_c)}^{context\ bias} \cdot \overbrace{p(y = c)}^{class\ bias}. \tag{1}$$

From Eq. (1), the noise identifier $p(y = c|z_c, z_e)$ is affected by the variations of 1) class bias $p(y = c)$: the distribution shift caused by class imbalance, and 2) context bias $p(z_e|y = c, z_c)$: spurious correlation [1] between context attributes and class. Such negative effect motivates us to

[1] For a thought example based on Eq. (1), if a class-specific attribute "body" and a context-specific attribute "spine" have strong co-occurrence under the "hedgehog" class, the wrong annotation "hedgehog" of a "porcupine" image with "spine" could be imperceptible for the identifier due to the high spurious correspondence $p(z_e = \text{“}spine\text{”}|y = \text{“}hedgehog\text{”}, z_c = \text{“}body\text{”})$.

```
Algorithm 1 H2E Framework
Input: NLT-Dataset $\{(x, y)\}$, # Iteration $T$, Confidence Threshold $\tau$.
Stage0 (Input: $\{\{(x, y)\}, \tau\} \rightarrow$ Output: $\{\Phi_0(\cdot), f_0(\cdot)\}$
    Initialize backbone $\Phi_0(\cdot)$, linear classifier $f_0(\cdot)$ by Part A in Appendix.

for $t = 1, 2, \ldots T$ do
    Stage1 (Input: $\{\{(x, y)\}, \Phi_{t-1}(\cdot), f_{t-1}(\cdot), g_{t-1}(\cdot)\}) \rightarrow$ Output: $\{\Phi_t(\cdot), f_t(\cdot), g_t(\cdot)\}$
     // Learn Noise Identifier.
       $\{e_1, e_2, \cdots\}$ generated multiple environments with Sec. 4.1.
       $g_t(\cdot) \leftarrow g_{t-1}(\cdot)$ by learning parameters $w$ through IRM with Eq.(2).
     // Easy Noise Removal.
     $\{(\tilde{x}, \tilde{y})\} \leftarrow \{(x, y)\}$ by commensurate Mixup with Eq.(3).
     $\Phi_t(\cdot) \leftarrow \Phi_{t-1}(\cdot)$, $f_t(\cdot) \leftarrow f_{t-1}(\cdot)$ by fine-tuning on $\{(\tilde{x}, \tilde{y})\}$.
end for

Stage2 (Input: $\{\{(x, y)\}, \Phi_T(\cdot), f_T(\cdot), g_T(\cdot)\ ) \rightarrow$ Output: updated $f_T(\cdot)$
    // Robust classifier tackling class imbalance.
    Update $f_T(\cdot)$ by reweighted Balance-softmax from Eq.(4).
Output: The final robust model $f_T(\Phi_T(\cdot))$ .
```

introduce the concept of "hard" and "easy" noise, which has not been addressed in the de-noise literature yet.

Noise is defined as training samples with a mismatch between the ground-truth label y and class-specific (causal) features z_c.

"Easy" Noise could be easily detected by the ideal identifier $p(y = c|z_c, z_e)$, regardless of the influence by $p(z_e|y = c, z_c) \cdot p(y = c)$. That is to say z_e is independent of y, *i.e*, $p(z_e|y = c, z_c)$ approaching $p(z_e|z_c)$ and z_e can be eliminated by $p(z_e|z_c)/p(z_e, z_c) = 1/p(z_c)$. Meanwhile, $p(y = c)$ is uniformly distributed under IID assumption in the conventional de-noise setting [24, 6, 9]. Since the above $1/p(z_c)$ and $p(y = c)$ could be both considered as constant, noise can be easily identified because $p(y = c|z_c, z_e)$ is directly calculated through the observation of $p(z_c|y = c)$.

"Hard" Noise is elusive as $p(y = c|z_c, z_e)$ is affected by the negative impact of $p(z_e|y = c, z_c) \cdot p(y = c)$, leading to erroneous abnormal identification.

The proposed **N**oisy **L**ong-**T**ailed classification aims to learn from the training data that possesses two joint phenomena: 1) the class distribution $p(y = c)$ is long-tailed; 2) part of the training samples (noise) are wrongly annotated. Some previous "easy" noises are thus turned into "hard" ones, resulting in that most of the conventional noise removal algorithms [1, 40, 7] are no longer reliable in NLT since the outlier samples can be either caused by the noisy labels with lower $p(z_c|y = c)$ or rare contexts and classes with lower $p(z_e|y = c, z_c) \cdot p(y = c)$. Therefore, we propose the following Hard-to-Easy framework, aiming to learn a fair noise identifier invariant to the change of $p(z_e|y = c, z_c) \cdot p(y = c)$, so the "hard" noises can thus be converted into "easy" ones.

4 Hard-to-Easy (H2E) Framework

As shown in Algorithm 1, our H2E framework is composed of two stages with an initial warm-up stage, where **Stage 1** (Section 4.1) obtains a fairer identifier by turning "hard" noise into "easy" through invariant muti-environment learning, thus obtaining a "cleaner" representation by removing the identified "easy" noise. An iterative virtuous circle is conducted to progressively identify "harder" noises and learn better representations. Eventually, in **Stage 2** (Section 4.2), a long-tailed loss, *e.g*, a balanced loss [30], is attached to the clean backbone from Stage 1 to learn a robust classifier.

4.1 Stage 1: Hard-to-Easy Noise Converter

Input : An initialized model containing backbone $\Phi(\cdot)$ and projection layer $f(\cdot)$, the training dataset $\{(x, y)\}$.

Output : A fair noise identifier $g(\cdot)$ invariant to environments, a fine-tuned cleaner backbone $\Phi(\cdot)$ and projection layer $f(\cdot)$.

As we discussed in Section 3, the imbalanced $p(z_e|y = c, z_c) \cdot p(y = c)$ turns "easy" noise into "hard", since the noise identifier $p(z_c|y = c)$ cannot be disentangled from the context and class bias. To better adapt to the long-tailed classification, the proposed noise identifier combines the previous LWS [29] and Logit Adjustment [27] classifiers as $g(\cdot) = f(\Phi(\cdot)) - w \cdot \log \pi$, where $\Phi(\cdot)$ is the frozen backbone extracting the image feature; $f(\cdot)$ projects feature vectors to the logit space; w is learnable parameters; π is the class distribution $p(y)$. However, the above $g(\cdot)$ can only remove the class bias $p(y = c)$ but not the context bias $p(z_e|y = c, z_c)$.

Intuitively, the crux for mitgating context bias $p(z_e|y = c, z_c)$ is to directly eliminate the impact of certain context z_e distribution, making $g(\cdot)$ an invariant identifier by capturing the class-specific attributes z_c. Inspired by Invariant Risk Minimization (IRM) [13], we construct a set of environments $\mathcal{E} = \{e_1, e_2, ...\}$, ensuring the diverse $p(z_e|y = c, z_c)$ in different environments. Then, IRM essentially regularizes $g(\cdot)$ to be equally optimal across environments with different context-distribution, thus removing the influence of context bias. The objective function of the proposed noise identifier invariant across $\mathcal{E}$ is thus defined as follows:

$$\begin{aligned} &\min_g \sum_{e \in \mathcal{E}} R^e(x, y; f(\Phi(\cdot)), g) \\ &\text{subject to } g \in \arg\min_g R^e(x, y; f(\Phi(\cdot)), g) \text{ for all } e \in \mathcal{E}, \end{aligned} \tag{2}$$

where $R^e(x, y; f(\Phi(\cdot)), g)$ is the risk under environment e; $g \in \arg\min_g R^e(x, y; M, g)$ for all $e \in \mathcal{E}$ means that the invariant identifier g should minimize the risk under all environments simultaneously. The implementation of IRM loss is in Appendix. Detailed process of Hard-to-Easy transformation is as below :

Environment Construction. A set of diverse environments $\{e_1, e_2, ...\}$ are constructed which ensure the variance of $p(z_e|y = c, z_c)$. The criterion of ideal environment construction is the orthogonality of context distribution; however, considering the computation consumption, we only construct three learning environments with classical sampling strategies and provide further ablations of the settings of environment construction in Sec. 5.4. As illustrated in Fig. 3, each learning environments adopts a different class-wise sampling strategy: 1) the instance-balanced sampler maintains the raw distribution of dataset, 2) the class-balanced sampler ensures the equal probability of being selected for each class, and 3) the class-reversed sampler aims to over-correct the imbalanced $p(y)$ by deliberately picking samples of class $y = c$ with the probability negatively correlated with class size. Then, in order to generate diverse distributions of $p(z_e|y = c, z_c)$ to avoid the over-sampling that generates a lot of duplicate samples (especially in tail categories), we adopt different data augmentation methods for each environment: e_3 with class-reversed sampler is equipped with the **"Strong"** augmentation [33] as it has the most number of duplicate samples, e_2 with class-balanced sampler uses **"Simple"** Random Flip and Resized Crop, as it has less duplicates, and e_1 is without augmentation (**"OFF"**) as it has no duplicate samples.

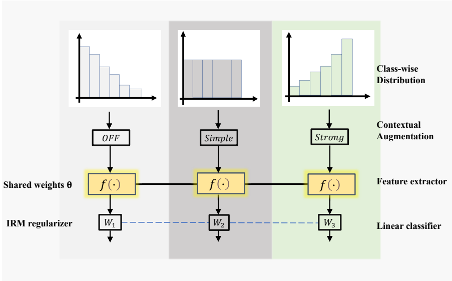


Figure 3: Multi-environment with diverse class and context distributions are built, then an IRM optimization [13] is applied to obtain an invariant identifier across environments

Easy Noise Removal. After obtaining the robust noise identifier $g(\cdot)$, in order to learn a better backbone with less contamination from the noise and prevent those clean images from being mistakenly penalized, we adopt a soft noisy removal strategy that uses Mixup [43] to dynamically augment samples according to the confidence generated by the noise identifier $g(\cdot)$. Specifically, we fine-tune $\Phi(\cdot)$ and $f(\cdot)$ with the generation of training pair $(\tilde{x}_{ij}, \tilde{y}_{ij})$ through conducting linear mixture of each two images as follows:

$$\begin{aligned} \tilde{x}_{ij} &= \delta_{ij} x_i + (1 - \delta_{ij}) x_j, \\ \tilde{y}_{ij} &= \delta_{ij} y_i + (1 - \delta_{ij}) y_j, \end{aligned} \tag{3}$$

where x_i and x_j are two images with labels y_i and y_j, respectively; δ_{ij} is the de-noise weight in proportion to the confidences $g(x_i)/g(x_j)$. Intuitively, the sample with a higher probability of being noise will have smaller weight in the mixed image $\tilde{x}_{ij}$. Such commensurate Mixup strategy alleviates the noise memorization effect[44] and prevent the overfitting of the already-detected easy noises. Moreover, the long-tailed effect can also be better eliminated by the Mixup, compared with other noise identification methods [24, 6, 9].

Iterative Refinement As both the noise identifier and the easy noise removal can benefit from the improvement of each other, we introduce an iterative framework to progressively identify "harder" noises and learn better representations, refers to $(\Phi_{t-1}, f_{t-1}, g_{t-1}) \rightarrow (\Phi_t, f_t, g_t)$ in Algorithm 1. It's worth noting that the iterative framework needs an initial step to learn a relatively pure feature representation by filtering the "simplest" noises, *i.e*, those samples that both class-specific contents $p(z_c|y = c)$ and context-specific environments $p(z_e|y = c, z_c)$ are obvious outliers of the corresponding class $y = c$. Due to the model memorization effect[44], those "simplest" noises can be identified by a commonly adopted warm-up stage when the noisy data haven't affected the learning of generalized patterns yet. The detailed implementation of this step is in Appendix.

4.2 Stage 2: Robust Classifier

Input : The last-iteration model $f(\Phi(\cdot))$ in Stage 1, noise identifier $g(\cdot)$, and training dataset $\{(x, y)\}$.

Output : The final robust model $f(\Phi(\cdot))$.

For the sake of simplicity of symbol, we omit all the subscript in this section. After obtaining purified representation from the iterative H2E, we assume the network has already modeled the underline $p(x|y = c)$. Therefore, we only need to tackle the class bias $p(y = c)$ in the cleaner data by any existing long-tailed classification algorithms. Without loss of generality, we resort to the balanced softmax loss [30], which can be defined as: $\mathcal{L}(x, y) = -y \log CE\left(f(\Phi(x)) + \log \pi\right)$, where CE denotes the cross-entropy loss; π is the distribution of $p(y)$ in the training data; $f(\cdot)$ is the learnable linear classifier initialized from the last-iteration in Stage 1. Noted that the noise identifier $g(\cdot)$ is not directly selected as the initialized classifier since there is a trade-off between the robustness of noise identification and performance of classification.

Besides, to eliminate all the noises detected by $g(\cdot)$, the final robust classifier is thus optimized by re-weighting all samples from the training data according to $\theta(x, y)$ as follows:

$$\mathcal{L}_{overall} = \frac{1}{N} \sum_{(x,y)} \theta(x, y) \cdot \mathcal{L}(x, y), \tag{4}$$

where N refers to the batch size, $\theta(x, y)$ is the weight parameter generated by $g_I(\cdot)$; $\theta(x, y) = p(y|x)$ when $p(y|x)$ is larger than any other $p(y'|x)$ and $\theta(x, y) = \eta$, a hyper-parameter threshold otherwise in order to make noisy samples contribute less to the loss.

5 Experiments

5.1 Benchmarks

We constructed three benchmarks for Noisy Long-Tailed (NLT) classification using both synthetic and realistic noise with class imbalance to imitate the real-world dataset at scale. As conventions [45], we call them **blue** (synthetic) and **red** (realistic), respectively. Our benchmarks: ImageNet-NLT, Animal10-NLT and Food101-NLT are built on top of three standard image classification dataset : Red Mini-ImageNet [45], Animal-10N [46] and Food-101N [47].

Dataset Construction. During the dataset construction, we adopted the standard rule of first building a balanced but noisy dataset and then transforming them into the long-tailed distribution to simulate the real distribution of noisy labels. To be specific, as for ImageNet-NLT, we augmented the vanilla Mini-ImageNet by adding correct-annotated samples from ImageNet with the same taxonomy. Then we followed the construction of Red Mini-ImageNet to replace ρ proportion of the original training images with noisy images from the web where ρ denotes the noise rate that is uniform across classes. Blue noises in ImageNet-NLT were generated by randomly sampling ρ training

Table 1: Overview of three NLT benchmarks with controlled noise level and imbalance ratio

Dataset	#Class	Train Size	Val Size	Noise Levels(%)	Imbalance Ratio
Red ImageNet-NLT	100	31,817	5,000	10,20,30	0,20
Blue ImageNet-NLT	100	31,817	5,000	10,20,30	0,20
Food101-NLT	101	63,460	25,000	≃8.0	20,50,100,200
Animal10-NLT	10	17,023	5,000	≃18.4	20,50,100,200

images from each class and substituting their labels uniformly drawn from other classes. The above process is not necessary for the construction of Animal10-NLT and Food101-NLT since their original datasets have already contained various real-world noises. After obtaining the balanced but noisy datasets, we simulated the long-tailed distribution in the real-world following the same setting as LDAM [48] to clip the size of each class: the long-tailed imbalance follows an exponential decay in the number of training samples across different classes. The imbalance ratio η denotes the ratio between the size of the maximum and minimum class.

Before sampling the long-tailed subsets of the original datasets, the balanced Red Mini-ImageNet contains 60,000 images from the original Mini-ImageNet [49] and 54,400 images with incorrect labels collected from the web. Animal-10N is a real-world noisy dataset of human-annotated online images of ten bewildering animals, with 50,000 training and 5,000 testing images in an estimated 8 % noise rate. Food-101N is a webly noisy food dataset containing 310,000 images from Google, Yelp, Bing and other search engines using the Food-101 [50] taxonomy.

Dataset Overview. Generally, ImageNet-NLT was further split into Red ImageNet-NLT (with realistic noise) and Blue ImageNet-NLT (with synthetic noise), both of which contain 31,817 training and 5,000 testing images of size 84×84. The imbalance ratio η is fixed at 20 with various noise ratio(%) $\rho \in \{10, 20, 30\}$ in the training sets while the testing set have a balanced number of images and correct annotations from the ILSVRC12 validation set. Animal10-NLT has $\{17,023, 13,996, 12,406\}$ training images with different imbalance ratio $\eta \in \{20, 50, 100\}$ and 5,000 balanced, clean testing images of size 64×64 with estimated $\rho = 0.08$. Food101-NLT has $\{63,460, 50,308, 43,303\}$ training images with different imbalance ratio $\eta \in \{20, 50, 100\}$ and 5,000 validation, 25,000 testing images of size 256×256 with estimated $\rho = 0.20$.

5.2 Implementation Details

We compared the proposed H2E with previous state-of-the-art methods in both fields of learning with noise and long-tailed classification. Moreover, since noisy long-tailed classification is rarely explored and the number of algorithms designed to fit our setting is small, we further proposed several joint algorithms that combine both long-tailed algorithms and de-noise methods for ablation.

LT Baselines: 1) LWS [29] decouples the learning procedure into representation learning and classifier fine-tuning, that re-scales the magnitude of classifier after obtaining the model capable of recognizing all classes; 2) The post-hoc logit adjustment (LA) [27] is another widely-used algorithm to compensate the long-tailed distribution by adding a class-dependent offset to each logit; 3) BBN [28] uses a framework of Bilateral-Branch network with a cumulative learning strategy; 5) LDAM [48] is a label-distribution-aware margin loss designed to re-balance the distribution.

De-noise Baselines : 1) Co-teaching+ [10] trains two networks then predict first, and selects small-loss data to teach its peer by keeping the data with prediction disagreement only; 2) Nested Co-teaching (N-Coteaching) [12] conducts adaptive data compression to train two separate networks and is further fine-tuned with Co-teaching (iii). 3) Co-Learning [51] further predigests these co-training methods through a shared feature encoder; 4) MentorMix [45] minimizes the empirical risk using curriculum learning to overcome both synthetic and realistic web noises; 5) Normalized Loss (NL) [52] combines passive and active loss to prevent over-fitting to noise labels. 6) Confident Learning (CL) [1] is a muti-round learning method which refines the selected set of clean samples by repeating the training round. (7) Two well-known SOTA denoise algorithms JoCoR [53] and DivideMix [9] are also included.

Joint Baselines: 1) HAR [22] is the first algorithm to tackle the long-tailed distribution with label noises (synthetic ones), that applies a Lipschitz regularizer with varying regularization to deal with noisy and rare examples in a unified way; 2) Co-teaching-WBL (Co-WBL) conducts a temperature weight to offset the tail classes in the procedure of Co-teaching [6] and fine-tunes with the balanced

Table 2: The evaluation (Top-1 Accuracy%) on ImageNet-NLT: we reported both blue (synthetic) and red (realistic) noises with three different noise rates: 10%, 20%, and 30%. Experiments demonstrate the effectiveness of the proposed H2E on all settings. The reported H2E-iter has the same number of total epochs with others

Category	Methods	10% ρ		20% ρ		30% ρ	
		red	blue	red	blue	red	blue
Baseline	CE	54.36	45.80	50.20	40.66	46.90	34.80
Denoise	Co-teaching+ [10]	45.58	53.16	44.14	49.43	43.16	37.47
	CL [1]	52.44	48.26	51.42	44.23	48.62	38.21
	MentorMix [45]	59.26	54.60	55.18	50.20	54.68	45.84
	NL [52]	56.36	52.48	53.84	44.80	51.28	39.14
	Co-learning [51]	50.19	49.72	48.77	42.65	44.37	37.20
LT	LWS [29]	57.05	52.36	53.62	44.78	49.15	36.54
	LA [27]	58.92	51.34	54.50	45.24	51.94	37.86
	BBN [28]	57.83	52.24	54.88	45.76	51.58	41.35
	LDAM [48]	59.24	53.02	55.98	46.60	54.38	42.76
Joint	HAR [22]	57.14	53.24	54.04	47.14	52.13	43.92
	NL+LA	59.80	51.88	57.21	46.52	53.56	37.40
	Co-WBL	61.44	54.98	57.62	52.40	54.08	45.81
	LDAM+NL	60.06	52.90	56.24	48.14	54.03	43.62
	MentorMix-RS	62.20	55.44	56.14	52.85	55.91	48.27
Ours	H2E	64.86	58.12	60.92	55.84	58.38	51.52
	H2E-iter	**65.29**	**59.42**	**62.12**	**56.31**	**60.66**	**52.57**

softmax loss [30]; 3) We also intuitively add Re-sampling strategy into the MentorMix [45], denoted as MentorMix-RS, to re-balance before curriculum learning; 4) A distribution-robust loss function [48] and a noise-robust loss function [52] is also combined, denoted as LDAM+NL.

Experimental Details. ResNet-18 [54] backbone was adopted for all methods in ImageNet-NLT and Animal10-NLT, and ResNet-50 [54] for Food101-NLT. They were all trained *from scratch* by SGD with weight decay of 1×10^{-4} and momentum of 0.9. All models were implemented in PyTorch and on NVIDIA Tesla A100 GPUs for 200 epochs with batch size of 512, except for Co-teaching+ [10] and Co-teaching-WBL with the batch size of 256. The initial learning rate was set to 0.2 and the default learning rate decay strategy is Cosine Annealing scheduler except for [28], [48], [45], which we followed the original setting to apply the multi-step scheduler, and we also maintained the warm-up stage and their backbone variations based on the corresponding papers. It's worth noting that we reported the version of single iteration H2E as well for fair comparison, which is conducted straightforwardly with 200 epochs. Further experiment on the ablation of iteration is included in Appendix.

5.3 Main Results

Evaluation on ImageNet-NLT. We conducted extensive experiments on ImageNet-NLT with three different noise ratios including both synthetic noises and realistic web noises, denoted as blue noises and red noises, respectively, following the setting of Jiang *et al* [45]. Fig. 4 presents that the iterative noise detection can better transfer hard noises into easy ones thus improving the model robustness step by step. We compared our method with several popular LT, de-noise baselines and a few joint baselines were also proposed in our experiments to intuitively combine LT algorithms with de-noise methods. As presented in Table 1, the proposed H2E consistently outperforms the baseline methods across different noise rates and noise types (red and blue). In particular, compared with MentorMix [45], which achieves the best performance among selected de-noise methods, H2E improves the test accuracy by 6.1% on average.

Besides, we can see from Table 2 that vanilla long-tailed methods outperform de-noise baselines in most lower noisy situations, while their performance gap is narrowed in a higher noise level. Intriguingly, some de-noise methods such as CL [1] and Co-teaching+ [10] are built upon the strong assumption of class balance and highly rely on the *small-loss trick*, so their performance degrades dreadfully in NLT, even worse than Cross-Entropy in many cases. As for the combined methods, we intuitively followed the essence of de-noise and long-tailed algorithms, and proposed MentorMix-

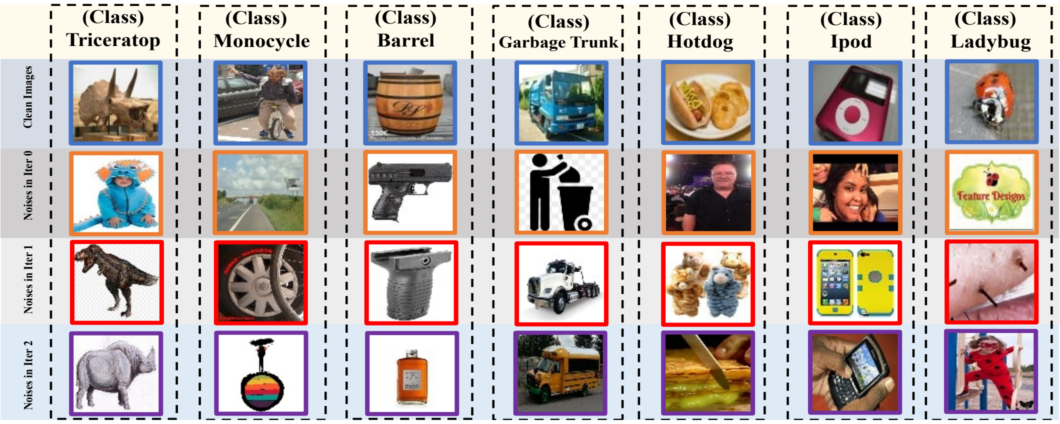


Figure 4: The example of iterative hard-to-easy transformation on Red ImageNet-NLT, presenting H2E gradually detects harder noises and improve overall robustness

RS and Co-teaching-WBL that outperform their counterparts and each individual component in most cases. However, for other strategies such as NL+LA, the improvement is limited and unstable with the contradiction of their re-balance strategies.

Not surprisingly, when we compared results between the red and blue noise settings under the same ratios, all the methods perform much better in red than blue noise except for Co-teaching+ [10], which applies strong intervention on blue noises. This finding is consistent with Jiang *et al* [45]'s conclusion and extends it into a more realistic situation. We believe the underlying reason behind is that blue noises corrupted by label flipping hurts the representation of the DNN more seriously than those open-set [55] and label-dependent red noises, which share more context-specific and class-related attributes. Further analysis of the combination of realistic noise and synthetic noise is given in Appendix.

Evaluation on Animal10-NLT and Food101-NLT. We further investigated the performance of H2E and other methods in Animal10-NLT and Food101-NLT with various imbalance ratios $\eta \in \{10, 20, 50\}$. As shown in Table 3, our method retains the most robust performance and outperforms other approaches in most cases as the imbalance ratio increase while most de-noise methods [12, 52] suffer from class imbalance and even perform worse than the Cross-Entropy. Long-tailed methods [28, 48] perform much better than de-noise methods in Animal10-NLT, attributes to the relatively low noise rate (estimated as 8%) and their specific design on network structures, e.g. cosine classifier in LDAM [48] and extra blocks in BBN [28]. Note that H2E is still comparable with state-of-art de-noise algorithms in a strictly balance training set, with 85.1% test accuracy in Animal-10N [46] and 73.4% test accuracy in Food-101N [47] from scratch.

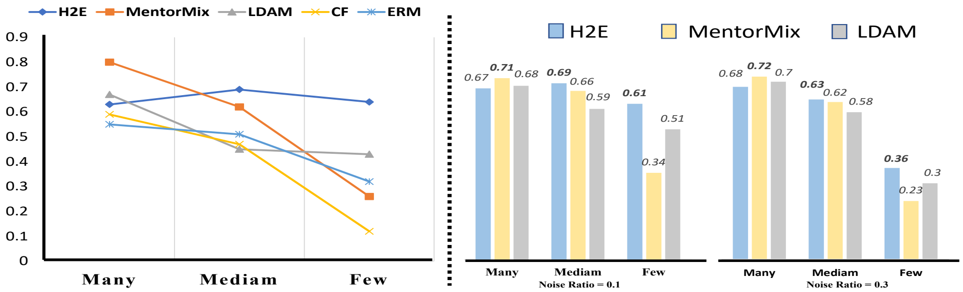
5.4 Ablation Studies and Further Analysis

Q1: ***Why H2E outperforms other methods in NLT?*** To better diagnose the improvement of H2E, we followed [56] and further recorded test accuracy and the precision of noise identification on three splits of classes: Many-shot(the top 25%), Medium-shot(the middle 50%) and Few-shot(the last 25%).

A1: Specifically in Fig. 5(b), the proposed H2E surpasses MentorMix[45] and LDAM[48] in few-shot by 20 % and 8% on average, which concretely demonstrates the robustness of H2E under imbalance distribution. It is clear from Fig. 5(a) that considering the precision of noise detection, H2E outperforms all of the selected methods in tail classes, which highlights its power to identify hard noises. From these two aspects, we could give a conclusion: the higher performance of H2E indeed attributes to its comparatively better hard noise identification capability and less hurt on correct-annotated but rare samples, especially on tail classes.

Table 3: Evaluations (Top-1 Accuracy%) on Food101-NLT and Animal10-NLT

Dataset	Food101-NLT			Animal10-NLT		
Methods/ η	20	50	100	20	50	100
CE	57.21	49.94	44.71	66.10	59.94	53.02
NL [52]	60.13	53.42	46.29	48.20	33.46	22.08
N-Coteaching [12]	52.44	40.21	29.78	57.54	41.40	39.04
DivideMix [9]	69.46	57.15	42.80	72.43	65.77	47.60
Co-learning [51]	53.76	45.92	35.10	61.70	52.76	43.23
JoCoR [53]	49.07	32.98	33.49	51.29	44.02	37.19
LDAM [48]	61.35	59.29	48.61	75.40	72.82	**68.21**
LA [27]	62.81	55.42	52.30	69.08	67.78	61.89
BBN [28]	63.44	57.89	53.16	72.14	70.26	60.08
LWS [29]	61.29	54.42	51.10	71.16	69.35	62.40
CL +LA	50.16	42.18	39.13	54.14	46.23	41.92
HAR [22]	59.95	52.45	46.12	71.92	68.43	62.19
Co-teaching-WBL	58.04	52.12	53.97	72.43	71.06	66.60
H2E	**70.35**	**63.69**	**58.66**	**77.04**	**74.94**	66.58



(a) ***Precision*** **of noise identification on** ***Training*** **Set** (b) ***Top-1 Accuracy%*** **of performance on** ***Testing*** **Set**

Figure 5: (a) Evaluation(Precision) of noise identification capability on Blue ImageNet-NLT. The proposed H2E indeed significantly improves the Few-shot(tail) categories by better identifying hard noises. (b)Evaluations (Top-1 Accuracy%) on Red ImageNet-NLT. We compare test accuracy in Many, Medium and Few shots among different methods

Q2: ***What impact performance of H2E considering environment construction?*** We conducted two ablation experiments on ImageNet-NLT: one is to analyze the number of environments and the other is to unify the augmentation strategies in each environment to so-called "OFF" augmentation.

A2: From Table 4, we found out that the overall improvement is converged to the number of environments when $e > 2$; Moreover, H2E will averagely degrade by *1.14*% without handling the duplication in tail classes , *i.e*, not augmentations, but it still largely outperforms other baselines in most settings comparing with the result in Table. 2, which shows the power of the proposed H2E.

Q3: ***How effective is each individual component in H2E?*** In Table 4, we replaced and modified each individual stage in H2E with other feasible methods to examine the effectiveness of the each stage.

A3: Considering a muti-stage framework, substituting any part of H2E caused the performance dropping to some extent. In detail, if we replace one component with other baselines, the Top-1 Accuracy will averagely degrade by *2.81*%.

Q4: ***What's the justification of augmentation strategies in environment construction?***

A4: (1) All methods in Section 5 contains the so-called strong augmentation in the stage of data prepossessing for fair comparison, **so we don't take any unfair advantage**. (2) Different augmentations are introduced only to construct environments with context-wise distribution shift. It's directly derived from our formulation, so IRM can focus on class-specific attributes, making it easier to converge and avoid both class and context bias.

Table 4: Ablation studies of **env**

Settings/ ρ		10%		20%		30%	
#env	Aug	red	blue	red	blue	red	blue
2		61.40	54.68	57.78	55.18	55.78	48.22
2	✓	62.15	55.70	59.66	55.38	56.02	49.10
3		62.79	57.40	60.64	55.06	57.14	49.38
3	✓	64.86	58.12	60.92	55.84	58.38	51.52
4		62.49	55.78	60.18	55.40	56.22	49.30
4	✓	65.38	56.52	60.42	55.96	57.56	49.78

Table 5: Effectiveness for each component in H2E framework

Component/ ρ		10%		20%		30%	
Stage1	Stage2	red	blue	red	blue	red	blue
CF	CE+RW	56.54	48.40	51.29	43.60	48.09	38.77
CF	H2E	63.90	56.10	60.08	54.76	57.80	49.30
LDAM	H2E	61.94	56.42	57.10	55.69	54.31	48.46
H2E	ERM+RW	60.20	57.02	58.54	50.94	56.27	46.24

Table 6: Evaluations (Top-1 Accuracy%) on Food101N and Animal10N under balanced class distributions.

Methods	Animal-10N	Food-101N
CE	81.28	69.42
NL [52]	83.24	69.91
N-Coteaching [12]	84.90	65.72
MentorMix [45]	84.10	73.58
HAR [22]	81.94	71.76
DivideMix [9]	85.72	75.83
Co-teaching+ [10]	83.66	72.97
H2E	85.10	73.34

6 Conclusion

We presented a novel noisy learning algorithm, Hard-to-Easy (H2E) for Noisy Long-Tailed Classification (NLT). We motivated from the observation that the tail class confidence boundary between clean and noisy samples are not clear, rendering conventional noise identification methods ineffective. Our analysis shows that it is because the class and context imbalance in long-tailed data that turn the "easy" noises into "hard" ones. The highlight of H2E is that it learns a robust noise identifier invariant to the class and context environmental changes. On three newly proposed NLT benchmarks: ImageNet-NLT, Animal10-NLT, and Food101-NLT, we demonstrated that H2E significantly outperforms existing de-noise methods, which do not take the imbalance into account. In future, we will conduct further analysis on NLT settings and more effective environment-invariant learning algorithms.

Acknowledgements: This project is partially supported by Alibaba-NTU Singapore Joint Research Institute (JRI), AI Singapore (AISG) Research Programme and the Agency for Science, Technology AND Research (A*STAR).

A Implement details

A.1 Initializing Stage

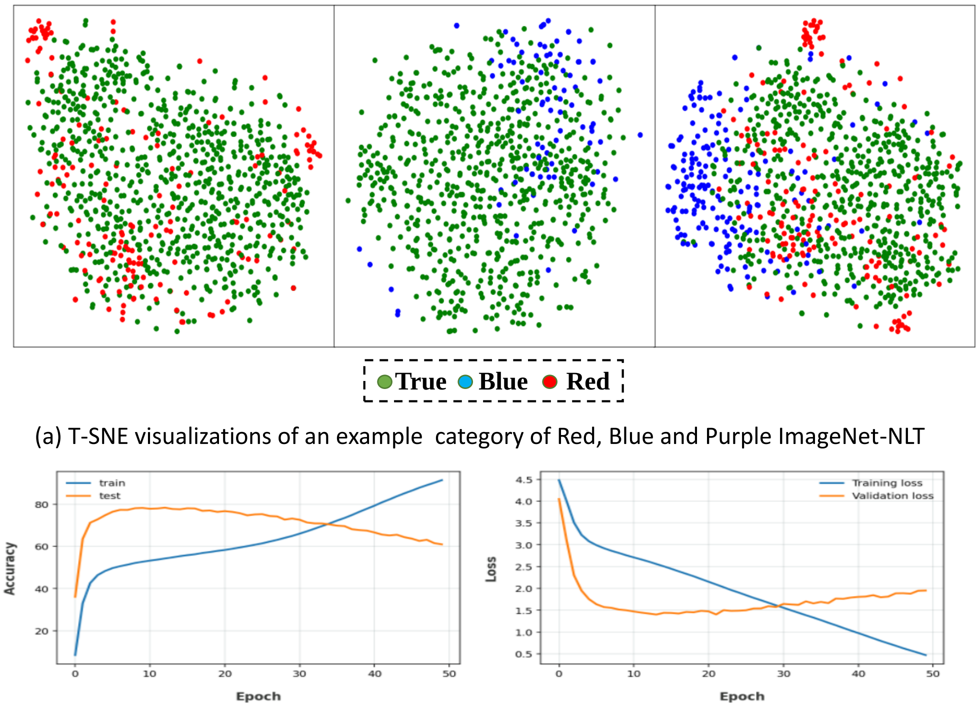
The hard noise identifier phase and easy noise removal proceed iteratively. The detailed implementation of the initializing step will be introduced here. In a general way, it is conducted before the noise removal stage which utilizes the model memorization effect [44]. Li *et al.* proved the distance

$$\|W_t - W_0\|_F \lesssim \left(\sqrt{K} + \left(K^2\epsilon_0/\|C\|^2\right) t\right) \tag{5}$$

from the initial weight W_0 to current weight W_t on a unit Euclidean ball assuming distinguishable samples,where K denotes the scales of clusters, and C is ϵ_0-neighborhood cluster centers. It demonstrates that DNNs tend to learn simple and generalized patterns in the first step,then over-fit to noisy patterns from easy to hard.

We use the preliminary network trained in the warm-up stage to extract features ν in each category and construct a cosine similarity matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$,where $M_{ij} = \frac{\nu(\mathbf{x}_i)^T \nu(\mathbf{x}_j)}{\|\nu(\mathbf{x}_i)\|_2 \|\nu(\mathbf{x}_j)\|_2}$ measures the similarity between images.

We define the density $\rho_i = \frac{1}{\|D_g\|} \sum_{j=1}^{\|D_g\|} \mathrm{M_{ij}}$ for each image in category D_g. Since the image with less p has more similar images around them, we could detect and give initial weight parameters



(a) T-SNE visualizations of an example category of Red, Blue and Purple ImageNet-NLT

(b) The training and validation loss , accuracy of an ERM-based pretrained ResNet-18 in Blue ImageNet-NLT

Figure 6: (a) The T-SNE [57] visualization of a certain category in Red, Blue and Purple ImageNet-NLT indicates the distinct patterns between synthetic and realistic noises : realistic noises share more cluster effect and severe confusion with true samples. Figure 3. (b) shows the corruption brought by blue noise degrades the performance of DNN by full over-fitting on mislabelled samples.

W_D in instance level based on the sequence with the above density. We control this procedure and normalize the weight parameter in both head and tail classes,t hus wouldn't meet self-confirmation bias caused by the imbalanced distribution.

B Extra experiment

B.1 Additional Results on balanced noisy Dataset

Although our proposed H2E is particularly designed under both longtailed and noisy datasets, it could still work well on balanced and noisy datasets. In the extra experiment, we just construct one environment and use the balanced softmax loss [30] to substitute the IRM loss. The implement details are the same as before. Table 6 gives competitive results compared to various state-of-art denoise algorithms. It demonstrates that without the strong assumption of small loss trick and frequent reweighting (For instance, Co-teaching [6] samples its small-loss instances as the useful knowledge and teaches it to its peer network for future training.), H2E framework could still show strong robustness when learning with noise on balanced datasets.

B.2 Additional Results on higher imbalance ratio

We further apply our method to the setups with higher imbalance ratio. For instance, in Animal10-NLT with imbalance ratio 200, H2E outperforms CE, MentorMix and BBN by *13.42%* , *7.12%* and *2.35%* respectively.

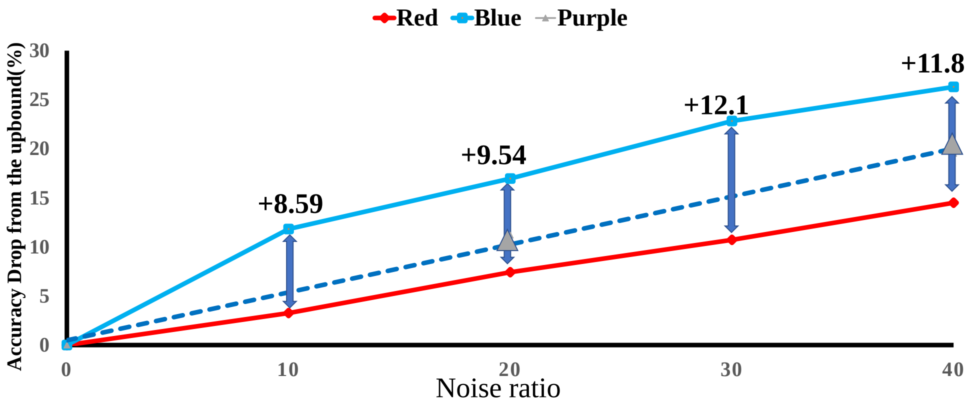


Figure 7: Performance drop of a ERM-based DNN from the up-bound accuracy in the clean setting with the increase of noise ratios.

B.3 Additional Results on Purple ImageNet-NLT

In the main manuscript, we focused on adding one type of noise (synthetic or realistic) and presented its performance for comparison. It is interesting to discuss the results under the longtailed dataset with compound noise, thus we constructed Purple ImageNet-NLT and compared H2E with previous state-of-the-art methods under this new setting. From Table 7, our method consistently retains the most robust performance and out-performs other approaches in most cases. This further supports that our proposed framework can adapt to complex noise conditions.

B.4 Ablation of Iterative Pattern

Sample reweighting is necessary and inevitable in noise sample selection: Some methods [6, 12, 10] conduct the reweighting schedule in frequently. For example, Co-teaching [6] teaches its peer network with small-loss samples in each mini-batch for future training. Others [9, 45] apply this procedure as segmentation of clean and noisy after several steps. As an illustration, DivideMix [9] first train several epochs with confident penalty and then conduct Gaussian mixture model on the loss from the former step to distinguish clean and noisy samples. We believe that the reason why the latter outperforms the former in our settings is largely because frequent sample reweighting in the early stage will extensively degrade the model representation capability in tail classes. The frequency of reweighting procedure should be considered and if we fixed the total number of epochs, there is a trade-off between the average training epochs in each iteration and the total number of loops. We analyzed the effect of the quantity of iterations n in Fig. 8, which states clearly that the performance of H2E is relatively stable in certain range $(n = 1, 2, 3, 4)$. However, when the number of epochs per iteration becomes much smaller, the overall performance degrades step by step, which is mainly caused by fact that the volatile oscillations of the model with few training epochs cannot support the hard noise identification stage to be better constrained and play a role. The detailed threshold of the quantity of iterations n changes depending on different settings.

C Discussion

Jiang *et al* [45] found that DNNS generalize better on red noise and reported the comparison of performance drop from the peak accuracy at different noise levels in blue and red noise settings. They hypothesized DNNS are more robust to web labels since they are more relevant ,in our words, sharing more context-specific attributes, to the clean training samples. We further proved this hypothesis with the following observation and analysis.

DNNS perform much better under red noise. Jiang *et al* [45] noticed the generalization performance of DNNS drops sharply with the ratio of noisy samples increases on blue noise while has relatively smaller difference on red noise. We first confirm Jiang *et al*'s conclusion in Fig. 7, where with the same noise ratio, the performance drop of DNNS training from scratch is considerably

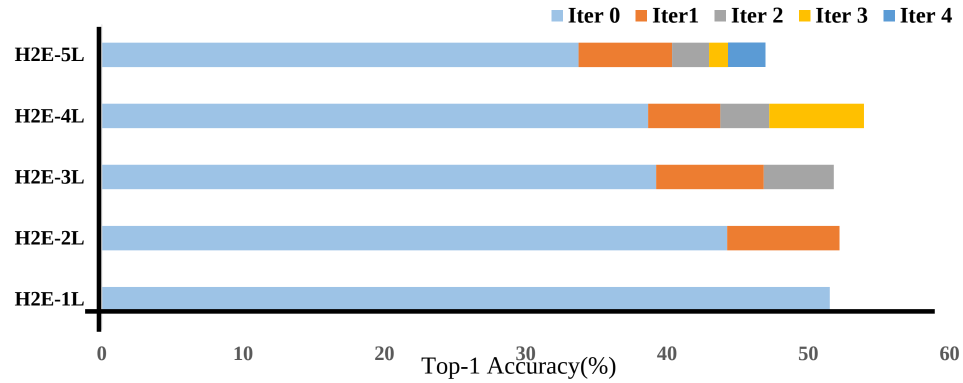


Figure 8: The ablation of iterative patterns. We reported the proposed H2E with different iteration numbers $n = 1, 2, 3, 4, 5$, where each iteration has $Total - Epoch/n$ epochs. The improvement of each iteration is presented with different colors.

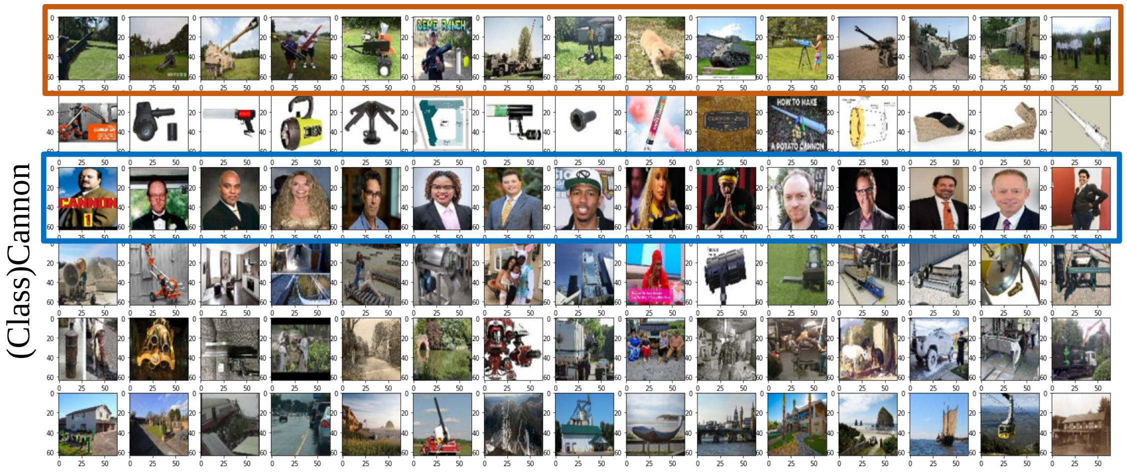


Figure 9: Visualization of red noise clustering results in class 'Cannon'.

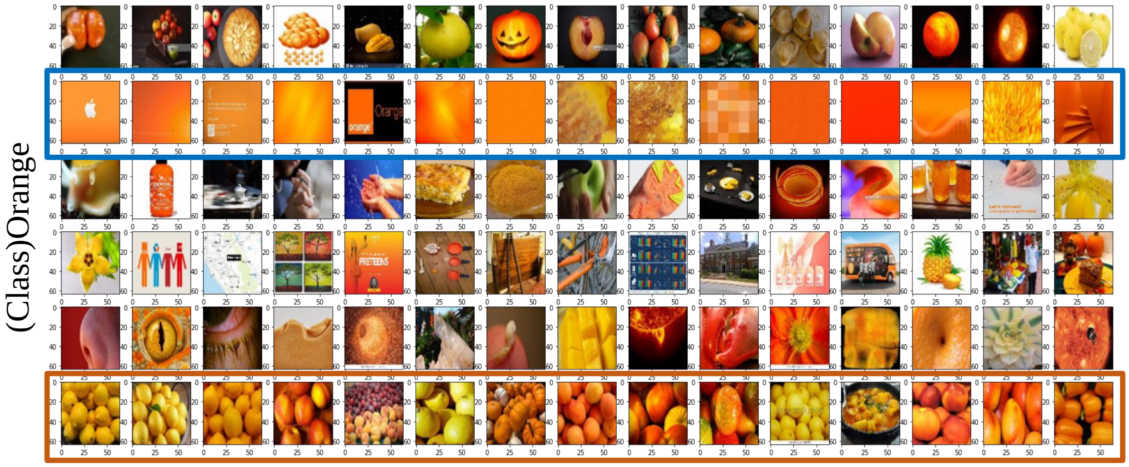


Figure 10: Visualization of red noise clustering results in class 'Orange'.

Table 7: The evaluation (Top-1 Accuracy%) on Purple ImageNet-NLT: we reported purple noises with two different noise rates: 20%, and 40%, where red and blue noise has the same proportion. Experiments demonstrate the effectiveness of the proposed H2E on all settings. The reported H2E-iter has the same number of total epochs with others.

Category	Methods	20% noise rate	40% noise rate
Baseline	CE	46.42	38.08
Denoise Baseline	Co-teaching+ [10]	41.65	38.84
	CL	48.49	40.14
	MentorMix [45]	52.94	43.57
	NL [52]	50.18	41.22
Longtail Baseline	LWS [29]	48.04	40.98
	LA [27]	52.30	42.38
	BBN [28]	48.36	41.42
	LDAM [48]	50.42	38.92
Combined Baseline	HAR [22]	49.77	38.63
	NL+LA	51.33	42.47
	Co-teaching-WBL	54.76	43.61
	LDAM+NL	53.21	41.09
	MentorMix-RS	54.82	45.14
Our methods	H2E	**59.94**	**50.64**
	H2E-iter	**61.78**	**52.22**

smaller in Red ImageNet-NLT than Blue ImageNet-NLT. However, it needs to be noted that the difference of Top-1 accuracy in this two settings is stable with the increase of noise ratio, which is inconsistent with Jiang *et al*'s finding in the balanced Red and Blue Mini-ImageNet. We consider this disagreement is mainly attribute to the fact that comparing with its counterpart in a balanced setting, red noise could heavily corrupt the tail classes by degrading the diversity of correct-annotated samples.

Red noise presents more cluster effect. Jiang *et al* [45] collected red noise retrieved by Google image search from text-to-image and image-to-image search, which resulted in the fact that part of the red noise contained semantic confusion. For instance in Fig. 10 and Fig. 9, for class *Orange*: Fruit of various citrus species in the family Rutaceae, plenty of pictures with orange color are selected from web ; for class *Cannon*: a large-caliber gun classified as a type of artillery, several artillery commanders and pop singers are picked by Google. In a word, semantic confusion generate red noise and meanwhile cause relatively more cluster effect, which may corrupt some noise identification strategies based on Self-supervised Learning [23] and Clustering [21].

Red noise is harder to identify but degrades less. We conducted a pretrained ResNet-50 as the feature extractor and gave the T-SNE [57] visualization of feature representation space in class *prayer rug* with red, blue and purple noise. Fig. 3 (a) shows that red noises appear in pairs and confuses heavily with true samples, which makes part of them harder to identify while blue noises present scattered distribution and have clearer boundary with true samples. Here comes the question : *Since the detection of red noises is much difficult than the blue ones, why most of the denoise methods could perform better in red noise settings?* From observation and analysis, two reasons are given: (1) In Fig. 6 (b), we found that DNN easily fits blue noises(random labels) and causes performance degradation even in the last epochs with relatively robust representation capability, while generalize much better with red noises and maintain stable performance in the last stage with the increase of epochs. (2) Blue noises are generated by random label flipping, which indicates that they have corresponding true labels and classes included in the training set, while most of Red noises from web aren't affiliated with any class in the training set. The different attribute of closed-set and open-set led to their different corruption on DNNS to certain extent.

References

[1] Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021. 1, 4, 7, 8

[2] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013. 1

[3] David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*, 2017. 1

[4] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. *arXiv preprint arXiv:1608.08967*, 2016.

[5] PS Sastry and Naresh Manwani. Robust learning of classifiers in the presence of label noise. In *Pattern Recognition and Big Data*, pages 167–197. World Scientific, 2017. 1

[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. *arXiv preprint arXiv:1804.06872*, 2018. 1, 3, 4, 6, 7, 12, 13

[7] Xinlei Chen and Abhinav Gupta. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1431–1439, 2015. 4

[8] Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, pages 1062–1070. PMLR, 2019. 1

[9] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020. 1, 2, 3, 4, 6, 7, 10, 11, 13

[10] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, pages 7164–7173. PMLR, 2019. 1, 2, 7, 8, 9, 11, 13, 15

[11] Baharan Mirzasoleiman, Kaidi Cao, and Jure Leskovec. Coresets for robust training of neural networks against noisy labels. *arXiv preprint arXiv:2011.07451*, 2020. 1

[12] Yingyi Chen, Xi Shen, Shell Xu Hu, and Johan AK Suykens. Boosting co-teaching with compression regularization for label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2688–2692, 2021. 1, 7, 9, 10, 11, 13

[13] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 1, 2, 5

[14] Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. *arXiv preprint arXiv:2010.05761*, 2020.

[15] Jiaxin Qi, Kaihua Tang, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Class is invariant to context and vice versa: On learning invariance for out-of-distribution generalization. In *ECCV*, 2022.

[16] Tan Wang, Qianru Sun, Sugiri Pranata, Karlekar Jayashree, and Hanwang Zhang. Equivariance and invariance inductive bias for learning from insufficient data. In *European Conference on Computer Vision (ECCV)*, 2022. 1

[17] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *arXiv preprint arXiv:2110.04596*, 2021. 2

[18] Jungsoo Lee, Eungyeup Kim, Juyoung Lee, Jihyeon Lee, and Jaegul Choo. Learning debiased representation via disentangled feature augmentation. *arXiv preprint arXiv:2107.01372*, 2021. 2

[19] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *arXiv preprint arXiv:2009.12991*, 2020. 2, 3

[20] Tan Wang, Zhongqi Yue, Jianqiang Huang, Qianru Sun, and Hanwang Zhang. Self-supervised learning disentangled group representation as feature. *arXiv preprint arXiv:2110.15255*, 2021. 2

[21] Tong Wei, Jiang-Xin Shi, Wei-Wei Tu, and Yu-Feng Li. Robust long-tailed learning under label noise. *arXiv preprint arXiv:2108.11569*, 2021. 2, 15

[22] Kaidi Cao, Yining Chen, Junwei Lu, Nikos Arechiga, Adrien Gaidon, and Tengyu Ma. Heteroskedastic and imbalanced deep learning with adaptive regularization. *arXiv preprint arXiv:2006.15766*, 2020. 7, 8, 10, 11, 15

[23] Shyamgopal Karthik, Jérome Revaud, and Boris Chidlovskii. Learning from long-tailed data with noisy labels. *arXiv preprint arXiv:2108.11096*, 2021. 2, 15

[24] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2304–2313. PMLR, 2018. 2, 3, 4, 6

[25] Pengxiang Wu, Songzhu Zheng, Mayank Goswami, Dimitris Metaxas, and Chao Chen. A topological filter for learning with label noise. *arXiv preprint arXiv:2012.04835*, 2020. 2

[26] John Shore and Rodney Johnson. Properties of cross-entropy minimization. *IEEE Transactions on Information Theory*, 27(4):472–482, 1981. 2

[27] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020. 2, 3, 5, 7, 8, 10, 15

[28] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9719–9728, 2020. 2, 3, 7, 8, 9, 10, 15

[29] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019. 2, 3, 5, 7, 8, 10, 15

[30] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. *arXiv preprint arXiv:2007.10740*, 2020. 3, 4, 6, 8, 12

[31] Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7610–7619, 2020.

[32] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *arXiv preprint arXiv:1902.07379*, 2019. 3

[33] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 3, 5

[34] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *CVPR*, 2020. 3

[35] Yifan Zhang, Bryan Hooi, Lanqing Hong, and Jiashi Feng. Test-agnostic long-tailed recognition by test-time aggregating diverse experts with self-supervision. *arXiv preprint arXiv:2107.09249*, 2021. 3

[36] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella X Yu. Long-tailed recognition by routing diverse distribution-aware experts. *ICLR*, 2020. 3

[37] Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *International Conference on Machine Learning*, pages 6226–6236. PMLR, 2020. 3

[38] Himanshu Kumar, Naresh Manwani, and PS Sastry. Robust learning of multi-label classifiers under label noise. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, pages 90–97. 2020.

[39] Ehsan Amid, Manfred K Warmuth, Rohan Anil, and Tomer Koren. Robust bi-tempered logistic loss based on bregman divergences. *arXiv preprint arXiv:1906.03361*, 2019. 3

[40] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning*, pages 312–321. PMLR, 2019. 3, 4

[41] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 2014. 3

[42] José M Bernardo and Adrian FM Smith. *Bayesian theory*, volume 405. John Wiley & Sons, 2009. 3

[43] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 5

[44] Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International conference on artificial intelligence and statistics*, pages 4313–4324. PMLR, 2020. 6, 11

[45] Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. Beyond synthetic noise: Deep learning on controlled noisy labels. In *International Conference on Machine Learning*, pages 4804–4815. PMLR, 2020. 6, 7, 8, 9, 11, 13, 15

[46] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. SELFIE: Refurbishing unclean samples for robust deep learning. In *ICML*, 2019. 6, 9

[47] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5447–5456, 2018. 6, 9

[48] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *NeurIPS*, 2019. 7, 8, 9, 10, 15

[49] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2019. 7

[50] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014. 7

[51] Cheng Tan, Jun Xia, Lirong Wu, and Stan Z Li. Co-learning: Learning from noisy labels with self-supervision. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1405–1413, 2021. 7, 8, 10

[52] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *International Conference on Machine Learning*, pages 6543–6553. PMLR, 2020. 7, 8, 9, 10, 11, 15

[53] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13726–13735, 2020. 7, 10

[54] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 8

[55] Hongxin Wei, Lue Tao, Renchunzi Xie, and Bo An. Open-set label noise can improve robustness against inherent label noise. *arXiv preprint arXiv:2106.10891*, 2021. 9

[56] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019. 9

[57] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 12, 15