Identifying Hard Noise in Long-Tailed Sample Distribution:Appendix

Xuanyu Yi[1], Kaihua Tang[1], Xian-Sheng Hua[2], Joo-Hwee Lim[3], Hanwang Zhang[1]

[1] Nanyang Technological University, Singapore
[2] Damo Academy, Alibaba Group, Hangzhou, China
[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

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 [10]. Li *et al.* proved the distance

$$\|W_t - W_0\|_F \lesssim \left(\sqrt{K} + \left(K^2\epsilon_0/\|C\|^2\right)t\right) \tag{1}$$

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 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.

A.2 Experimental Implementation

ResNet-18 [5] backbone was adopted for all methods in ImageNet-NLT and Animal10-NLT, and ResNet-50 [5] 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+ [16] 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 [17], [2], [6], 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.

B 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 $\rho = 0.20$.

C Extra experiment

C.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 [14] to substitute the IRM loss. The implement details are the same as before. Table 1 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 [4] 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.

C.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.

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

Methods	Animal-10N	Food-101N
CE	81.28	69.42
NL [11]	83.24	69.91
N-Coteaching [3]	84.90	65.72
MentorMix [6]	84.10	73.58
HAR [1]	81.94	71.76
DivideMix [9]	85.72	75.83
Co-teaching+ [16]	83.66	72.97
H2E	85.10	73.34

C.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 2, 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.

Table 2. 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+ [16]	41.65	38.84
	CL	48.49	40.14
	MentorMix [6]	52.94	43.57
	NL [11]	50.18	41.22
Longtail Baseline	LWS [7]	48.04	40.98
	LA [13]	52.30	42.38
	BBN [17]	48.36	41.42
	LDAM [2]	50.42	38.92
Combined Baseline	HAR [1]	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**

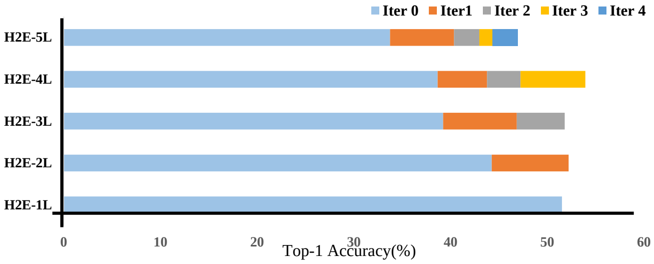


Fig. 1. The ablation of iterative patterns. We reported the proposed H2E with different iteration numbers $n = 1, 2, 3, 4, 5$, where each iteration has $Total - Epoch/n$ epochs. The improvement of each iteration is presented with different colors.

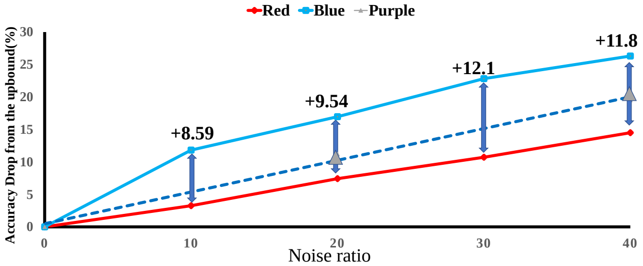


Fig. 2. Performance drop of a ERM-based DNN from the up-bound accuracy in the clean setting with the increase of noise ratios.

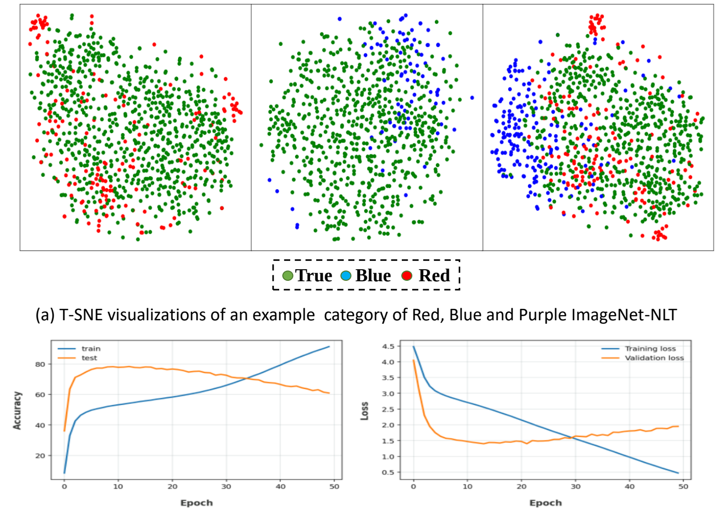
C.4 Ablation of Iterative Pattern

Sample reweighting is necessary and inevitable in noise sample selection: Some methods [4,3,16] conduct the reweighting schedule in frequently. For example, Co-teaching [4] teaches its peer network with small-loss samples in each mini-batch for future training. Others [9,6] 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. 1, 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.

D Discussion

Jiang *et al* [6] 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.



(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

Fig. 3. (a) The T-SNE [12] 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.

DNNS perform much better under red noise. Jiang *et al* [6] 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. 2, where with the same noise ratio, the performance drop of DNNS training from scratch is considerably 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* [6] 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. 5 and Fig. 4, 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 [8] and Clustering [15].

Red noise is harder to identify but degrades less. We conducted a pretrained ResNet-50 as the feature extractor and gave the T-SNE [12] 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. 3 (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.

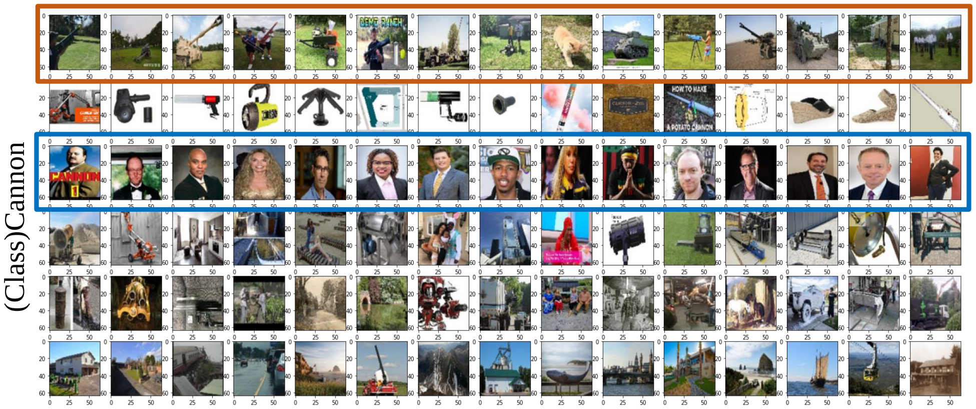


Fig. 4. Visualization of red noise clustering results in class 'Cannon'.

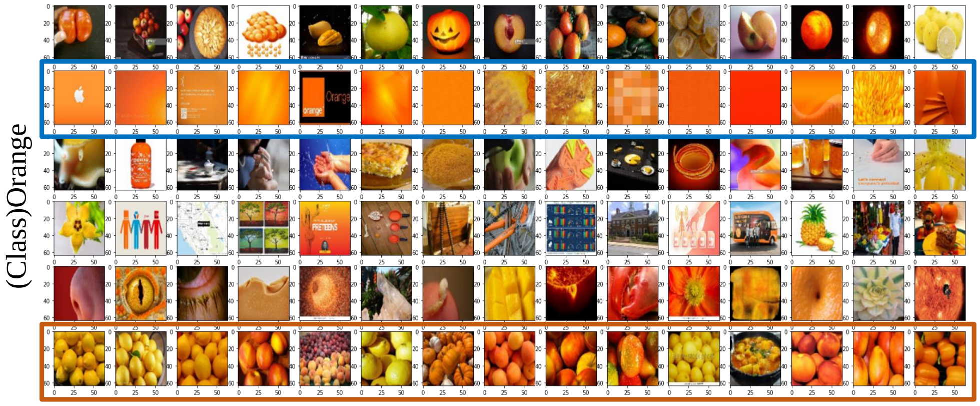


Fig. 5. Visualization of red noise clustering results in class 'Orange'.

References

1. Cao, K., Chen, Y., Lu, J., Arechiga, N., Gaidon, A., Ma, T.: Heteroskedastic and imbalanced deep learning with adaptive regularization. arXiv preprint arXiv:2006.15766 (2020) 3
2. Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. NeurIPS (2019) 2, 3
3. Chen, Y., Shen, X., Hu, S.X., Suykens, J.A.: Boosting co-teaching with compression regularization for label noise. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2688–2692 (2021) 3, 4
4. Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., Sugiyama, M.: Co-teaching: Robust training of deep neural networks with extremely noisy labels. arXiv preprint arXiv:1804.06872 (2018) 2, 4
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 1
6. Jiang, L., Huang, D., Liu, M., Yang, W.: Beyond synthetic noise: Deep learning on controlled noisy labels. In: International Conference on Machine Learning. pp. 4804–4815. PMLR (2020) 2, 3, 4, 5, 6
7. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. arXiv preprint arXiv:1910.09217 (2019) 3
8. Karthik, S., Revaud, J., Chidlovskii, B.: Learning from long-tailed data with noisy labels. arXiv preprint arXiv:2108.11096 (2021) 6
9. Li, J., Socher, R., Hoi, S.C.: Dividemix: Learning with noisy labels as semi-supervised learning. arXiv preprint arXiv:2002.07394 (2020) 3, 4
10. Li, M., Soltanolkotabi, M., Oymak, S.: Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In: International conference on artificial intelligence and statistics. pp. 4313–4324. PMLR (2020) 1
11. Ma, X., Huang, H., Wang, Y., Romano, S., Erfani, S., Bailey, J.: Normalized loss functions for deep learning with noisy labels. In: International Conference on Machine Learning. pp. 6543–6553. PMLR (2020) 3
12. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008) 5, 6
13. Menon, A.K., Jayasumana, S., Rawat, A.S., Jain, H., Veit, A., Kumar, S.: Long-tail learning via logit adjustment. arXiv preprint arXiv:2007.07314 (2020) 3
14. Ren, J., Yu, C., Sheng, S., Ma, X., Zhao, H., Yi, S., Li, H.: Balanced meta-softmax for long-tailed visual recognition. arXiv preprint arXiv:2007.10740 (2020) 2
15. Wei, T., Shi, J.X., Tu, W.W., Li, Y.F.: Robust long-tailed learning under label noise. arXiv preprint arXiv:2108.11569 (2021) 6
16. Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., Sugiyama, M.: How does disagreement help generalization against label corruption? In: International Conference on Machine Learning. pp. 7164–7173. PMLR (2019) 2, 3, 4
17. Zhou, B., Cui, Q., Wei, X.S., Chen, Z.M.: 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. pp. 9719–9728 (2020) 2, 3