Supplementary Materials: Learning from Multiple Annotator Noisy Labels via Sample-wise Label Fusion

Zhengqi Gao[1], Fan-Keng Sun[1], Mingran Yang[1],
Sucheng Ren[2], Zikai Xiong[1], Marc Engeler[3], Antonio Burazer[3], Linda Wildling[3], Luca Daniel[1], and Duane S. Boning[1]

[1] Massachusetts Institute of Technology, Cambridge MA 02139, USA
[2] South China University of Technology, Guangzhou, China
[3] Takeda Pharmaceuticals Co., Ltd., Zurich, Switzerland

A Ablation Studies on MNIST

Hammer-Spammer Synthesis. Here we conduct experiments on MNIST under the hammer-spammer synthesis method used in MBEM and TraceReg. Specifically, we consider the class-wise hammer-spammer synthesis. A hammer refers to always correct and a spammer refers to always wrong. In their original implementations, an annotator is a hammer with probability p and a spammer with probability $1-p$ for any class $k \in \{0, 1, \cdots, K-1\}$ (where $K = 10$ in MNIST dataset). During our implementation, we find that this synthesis has too much variance. Namely, one annotator provides wrong labels for all samples while other annotators provide correct labels for some classes and achieve moderate accuracies. Thus, we slightly revise the hammer-spammer implementation. For one annotator, we randomly select N_{correct} classes and specify that this annotator is a hammer on those classes, and a spammer on the remaining ones. In this example, we make up five annotators following our class-wise hammer-spammer synthesis. Other settings (e.g., learning rate, hyper-parameter) are identical to the experiment in the main text.

Table 1. Test accuracies of different methods (%) are evaluated under $N_{\text{correct}} = 3$. The test accuracy of training with golden labels (i.e., upper bound) is 99.20%.

Max AnTs'	Mean AnTs'	Min AnTs'	TraceReg	Mjv	MBEM	WDN	Ours
40.68%	39.56%	38.55%	84.54%	86.27%	**88.31%**	88.26%	88.26%

Test accuracies of different methods are reported in Table 1. The first, second, and third columns report the max, mean, and min test accuracy that the model trained solely with one annotator's labels could achieve. All values in the table are reported after averaging five independent experiments. We observe that in this case, MBEM, WDN, and our method achieve similar accuracies and outperform others. Since in this setting, the annotator synthesis method is not sample-wise[4], it makes sense that our method is not outstanding. This alternatively indicates that our method might be better suitable to the case when the annotator labels are data-dependent. When labels are data-independent, methods such as MBEM might be sufficient. Nevertheless, it appears to us that in real applications, sample-wise annotator labels seem more reasonable. It would be great if a public dataset is available in the future so that all methods could be tested on the same page.

Remarks on our Euclidean synthesis. *At present, no publicly available dataset provides annotator error data, and we have to synthesize annotators' labels in some way.* In previous literature, they also use synthetic methods (e.g., the hammer-spammer synthesis mentioned above) to generate annotator labels on ImageNet, CIFAR-10. We believe a synthesis method based on image similarity metric is more realistic, and Euclidean distance is widely used in literature and code packages. Another major reason to use Euclidean distance in synthesis is that it can measure similarity not only in images, but also in audio, text, etc. Our Euclidean distance is calculated on raw inputs. Using it on latent features would require training a good network and performing inference on all data in preprocessing, which is time consuming. A dataset containing multiple annotators' labels for one data sample would greatly help the community, as we could then more easily inspect the performances of different methods with apple-to-apple comparisons.

Remarks on the range of ϵ in our synthesis. Going back to our annotator label synthesis methods, when Euclidian distance between an image and annotator's 'weakness' image is smaller than ϵ, random labels are returned. The range of ϵ (e.g., $[30, 35]$ in Table 1 of the main text) was selected as follows: (i) min ϵ corresponds to when annotators provide wrong labels, but our method achieves similar accuracy as training with true labels; (ii) max ϵ corresponds to training where one annotator's labels gives disastrous accuracy (e.g., around 15%). Case (i) shows how bad the annotators' labels can be such that our proposed method will start to drop from the golden accuracy. Case (ii) demonstrates how good our method is when at least one annotators' labels are almost completely unreliable. Our choices of min and max ϵ lie at two extreme ends and the chosen range is wide enough to verify the method at all cases. Similar criterion is applied to CIFAR-100 and ImageNet-100 experiments.

B Ablation Studies on CIFAR-100

Here we report results of ablation studies on (i) only using confusion matrices $\{\mathbf{P}_n^{(r)}\}_{r=1}^R$, (ii) only using weight vectors $\mathbf{w}_n$, (iii) hyper-parameter λ, and (iv) number of basis matrices M in Table 2 and Fig. 1.

[4] Precisely, not as sample-wise as the synthesis method in our main text.

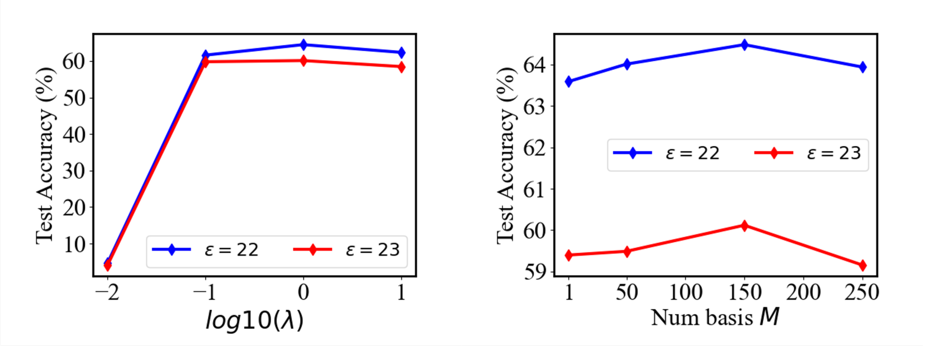

Fig. 1. Left: ablation study on λ. The best test accuracy is achieved at $\lambda = 1.0$ for both $\epsilon = 22$ and $\epsilon = 23$. This is also the value we use in the main text. Right: ablation study on M. The best test accuracy is achieved at $M = 150$.

Table 2. Accuracies (%) on CIFAR-100 under different ϵ.

ϵ	Ours (w/ only $\mathbf{w}_n$)	Ours (w/ only $\mathbf{P}_n^{(r)}$)	Ours (w/ both)
22	63.59	60.25	**64.48**
23	59.40	57.07	**60.12**

As we mentioned earlier, there is no publicly available dataset provides annotator error data, and we have to synthesize annotators' labels in some way. Most of our experiments are carried out based on our Euclidean synthesis technique. Here we perform an extra experiment using neural networks as annotators in Table 3. This, to the best of our efforts, is the closest to a real-world scenario. Specifically, we inspect the reported test accuracies of the trained models listed at https://github.com/chenyaofo/pytorch-cifar-models. Next, for our experimental purpose, we deliberately choose three models (MobileNet, Vgg11, ShuffleNet) which perform badly, download and regard them as the three annotators. Results are reported in the following Table 3.

Table 3. Accuracies (%) on CIFAR-100. We take pretrained MobileNet, Vgg11, ShuffleNet as the three annotators. Their test accuracies are 65.28%, 66.90%, 60.17%, respectively. We note that since our data normalization might be different from the one used to originally train these three models, we witness a difference between the accuracies reported here and on https://github.com/chenyaofo/pytorch-cifar-models. The results of this experiment show that our label fusion approach again outperforms other annotator error methods, with this alternative setup to obtain example annotator errors.

MBEM	WDN	TraceReg	Mjv	Ours	w/ true
70.48	72.08	71.47	70.60	**73.14**	73.24