Supplementary Material: "Learning Discriminative Dynamics with Label Corruption for Noisy Label Detection"

Suyeon Kim[1], Dongha Lee[2*], SeongKu Kang[3], Sukang Chae[1], Sanghwan Jang[1], Hwanjo Yu[1*]
[1] POSTECH, [2] Yonsei University, [3] University of Illinois at Urbana Champaign
{kimsu, chaesgng2, s.jang, hwanjoyu}@postech.ac.kr, donalee@yonsei.ac.kr, seongku@illinois.edu

A. Experiment Setup

A.1. Datasets

Synthetic noise: instance-dependent label noise. We detail the process of generating instance-dependent label noise [16], which is the synthetic type label noise utilized in our experiments. The key idea is that the probability of an instance being incorrectly labeled to other classes is calculated based on both the input feature and its label, using randomly generated feature projection matrices with respect to each class. The procedure is provided in Algorithm 1.

Algorithm 1 Instance-Dependent Label Noise Synthesis

```
Input: Clean dataset $D = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$, $\mathbf{x}_n \in \mathbb{R}^{d_\mathbf{x}}$, Noise rate $\eta$, Number of classes $C$
Output: Noisily labeled dataset $\tilde{D} = \{(\mathbf{x}_n, \tilde{y}_n)\}_{n=1}^N$
Sample $C$ feature projection matrices $\{\mathbf{W}_1, ..., \mathbf{W}_C\}$ from a standard normal distribution $\mathcal{N}(0,1)$, with each $\mathbf{W}_c \in \mathbb{R}^{d_\mathbf{x} \times C}$.
for $n = 1, \dots, N$ do
    Sample $q \in \mathbb{R}$ from a truncated normal distribution $\mathcal{N}(\eta, 0.1^2)$ within the interval [0,1].
    Compute probability vector by $p = \mathbf{x}_n \mathbf{W}_{y_n} \in \mathbb{R}^C$.
    Set the probability of the true class to be negative infinity $p_{y_n} = -\infty$.
    Adjust $p = q \times \mathrm{Softmax}(p)$ and set $p_{y_n} = 1 - q$.
    Sample corrupted label $\tilde{y}_n$ from $C$ classes according to the modified probability distribution $p$.
end for
```

Clothing1M [17]. To assess DynaCor's performance with systematic type label noise, we use a real-world dataset Clothing1M, which consists of clothing images across 14 classes[1] collected from online shopping websites. It comprises one million images with inherent noisy labels induced by automated annotations derived from keywords in the text surrounding each image. It also provides 50K, 14K, and 10K instances verified as clean for training, validation, and testing purposes. Adhering to the previous experimental setup [6], for training, we utilize randomly sampled 120K instances from the 1M noisy dataset while ensuring each class is balanced. To evaluate classification performance, we use the 10K clean test set.

A.2. Reproducibility

For reproducibility, we provide detailed hyperparameters for (1) classifiers used to generate training dynamics or to learn robust models and (2) dynamics encoder to learn discriminative representations of the training dynamics.

Classifier. Table 5 shows details of the datasets, models, and training parameters used to generate training dynamics or to learn robust models in each section of this paper. Optimizer and momentum are fixed as SGD and 0.9, respectively. In the case of CLIP with MLP, we obtain input features using a fixed image encoder from CLIP and train only MLP, which consists of two fully connected layers of 512 units with ReLUs [8]. Resnet50 is pre-trained on ImageNet [2] and is fine-tuned on Clothing1M. We follow the experimental setups described in the reference papers.

Dataset	CIFAR-10/CIFAR-100			Clothing1M
Section	5.2 to 5.4		5.5	Appendix D
Model	CLIP [12] w/ MLP	Resnet34 [3, 15]	PreAct- Resnet18 [4, 9]	Resnet50 [3, 6]
Learning rate	0.1	0.1	0.02	0.002
Weight decay	5×10^{-4}	5×10^{-4}	5×10^{-4}	0.001
LR scheduler	Cosine	Multi-step	Multi-step	Multi-step
Batch size	128	128	128	64
Epochs	30	100	300	10
α	0.5	0.05	0.05	0.5

Table 5. Detailed hyperparameters used in the experiments for the classifiers.

Dynamics encoder. For the dynamics encoder in DynaCor, we use a 1D Convolutional Neural Network (1D-

[*] Corresponding authors

[1] T-shirt, Shirt, Knitwear, Chiffon, Sweater, Hoodie, Windbreaker, Jacket, Down Coat, Suit, Shawl, Dress, Vest, and Underwear

CNN). It consists of three convolutional layers, each incorporating rectified linear units (ReLUs) [8], followed by a linear layer with 512 output units. For optimization, we use Adam [7] with a learning rate 1×10^{-5} and a weight decay 5×10^{-4} without implementing a learning rate scheduler. The model is trained for 10 epochs with a batch size of 1024.

B. Analyses of Training Dynamics

To assess the distinguishability of the inherent patterns manifested in the training dynamics, we conduct a controlled experiment using classification within a supervised learning framework. This is predicated on the assumption that ground-truth annotations are available, explicitly specifying each instance as being correctly or incorrectly labeled.

We first provide preliminaries for analyses (Sec. B.1). Then, we demonstrate the efficacy of capturing temporal patterns in training dynamics versus summarizing these dynamics into a single scalar value (Sec. B.2) on various training signals. Lastly, we evaluate which training signals exhibit more distinctive patterns (Sec. B.3).

B.1. Preliminaries

Training signals. Table 6 summarizes various training signals introduced in the literature. Given an instance $(\mathbf{x}, y)$ and a classifier f, let $f(\mathbf{x}) \in \mathbb{R}^C$ and $f_y(\mathbf{x})$ denote the output logits of an instance $\mathbf{x}$ for C classes and its value for class y, respectively. $\ell(\cdot,\cdot)$ is a loss function, and $p_y(\mathbf{x}) = \frac{\exp f_y(\mathbf{x})}{\sum_{c=1}^{C} \exp f_c(\mathbf{x})}$ is a predicted probability of class y. $\mathbf{v_x}$ indicates penultimate layer representation vectors of an instance $\mathbf{x}$, and $\mathbf{u}_y$ is a representative vector for class y, derived through performing eigen decomposition on the gram matrix of data representations. $\langle\cdot,\cdot\rangle$ denotes inner product.

Training signal	Formula, $t_{\mathbf{x}}$
Loss [5]	$\ell(f(\mathbf{x}), y)$
Probability [1]	$p_y(\mathbf{x})$
Probability difference [13]	$\max_c p_c(\mathbf{x}) - p_y(\mathbf{x})$
Logit difference [11]	$f_y(\mathbf{x}) - \max_{c \neq y} f_c(\mathbf{x})$
Alignment of pre-logits [6]	$\langle \mathbf{u}_y,\ \mathbf{v_x} \rangle^2$

Table 6. Various types of training signals.

Supervised experimental setting. As illustrated in Figure 4, we generate training dynamics by employing a classifier that predicts the class probabilities for each input instance across the set of classes. Subsequently, we construct a new dataset comprising these extracted training dynamics and the corresponding ground-truth labels that are assumed to exist. This new dataset is then utilized to train a 1D convolutional neural network (1D-CNN) classifier (henceforth referred to as a *binary classifier*) that distinguishes between correctly and incorrectly labeled instances based on the patterns in their training dynamics. We train the *binary classifier* (whose encoder is the same as our dynamics encoder) for 20 epochs using the Adadelta [18] optimizer with an initial learning rate of 1 and a StepLR scheduler that reduces it by 1% for every epoch. The batch size is set to 128. During training, we monitor the model's performance on a validation set and report the F1 score for detecting incorrectly labeled instances on the test set, corresponding to the point where the validation F1 score achieves its maximum value.

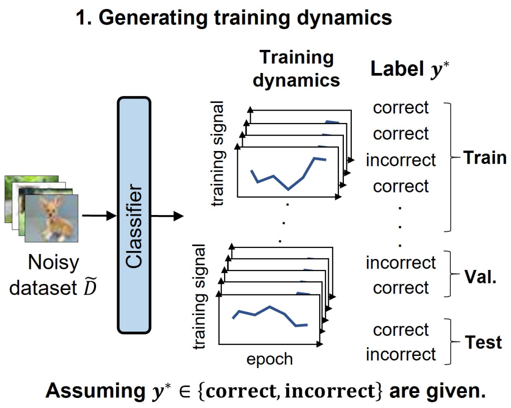

Figure 4. Dataset construction for supervised learning.

B.2. Temporal patterns in training dynamics

To assess the effectiveness of capturing temporal patterns within training dynamics compared to summarizing them into a single scalar value [1, 11], we conduct experiments using them as input to the *binary classifier* in the supervised setting. For the training dynamics, we use

$$\mathbf{t_x} = [t_{\mathbf{x}}^{(1)}, .., t_{\mathbf{x}}^{(E)}], \tag{1}$$

where $t_{\mathbf{x}}^{(e)}$ is a training signal at epoch e for an instance $\mathbf{x}$, and E is the maximum number of training epochs. For the summarized one, we use a statistical method [1, 11] that average the series of temporal signals into a single scalar value $s_{\mathbf{x}}$ to encapsulate the essential features.

$$s_{\mathbf{x}} = \frac{1}{E} \sum_{e=1}^{E} t_{\mathbf{x}}^{(e)}, \tag{2}$$

To evaluate the relative efficacy of these approaches, we use two distinct types of training signals: probability and logit difference in Table 6. For the *binary classifier* of the summarized one, we adopt a multi-layer perceptron (MLP) of two hidden layers. To ensure the model's sufficient capacity to learn patterns in the data, we increase the model parameters until performance does not improve further.

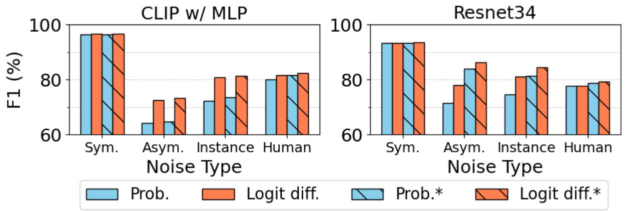

Figure 5. Comparison of detection F1 score (%) achieved by the *binary classifiers* trained using the training dynamics (comb-pattern bar and star marker in legend) versus those trained with the summarized one for various noise types on CIFAR-100. Prob. and Logit diff. indicate the types of training signals in Table 6. Noise rates of Sym., Asym., and Instance are 0.6, 0.4, and 0.3, respectively. The human-induced noise has noise rates of 0.4. CLIP w/ MLP (Left) and Resnet34 (Right) are used for training dynamics generation.

Figure 5 shows that the models trained with the training dynamics consistently outperform those with the summarized training dynamics. The results demonstrate that temporal patterns within training dynamics help distinguish between correctly and incorrectly labeled instances.

B.3. Comparison of various training signals

We compare the detection F1 score of the *binary classifier* trained with the training dynamics derived from various training signals in the supervised setting.

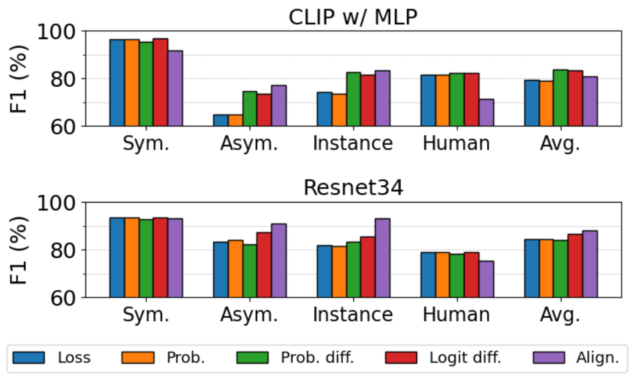

Figure 6. Comparison of detection F1 score (%) of the raw training dynamics from various training signals on CIFAR-100. Noise rates of Sym., Asym., and Instance are 0.6, 0.4, and 0.3, respectively. The human-induced noise type has noise rates of 0.4. The Avg. indicates an averaged F1 score (%) over all noise types. CLIP w/ MLP (Upper) and Resnet34 (Lower) are used for training dynamics generation.

Figure 6 shows that, on average, more processed training signals, such as probability differences and alignment of pre-logits, exhibit superior performance compared to simpler ones. In this study, we select logit difference as the base proxy measure due to its consistent performance across various experimental settings. Moreover, we observe that detection performance for different types of noises is highly correlated with model architecture. We leave the study of the influence of model architectures in future work.

C. Proof of the Lower Bound of η_γ

Proposition 1 (Lower bound of η_γ) *Let η_γ denote the noise rate of the corrupted dataset. Given the diagonally dominant condition, i,e., $\eta < 1 - \frac{1}{C}$, for any $\gamma \in (0, 1]$, η_γ has a lower bound of $1 - \frac{1}{C}$.*

Proof. The proportion of the correctly labeled instances in the corrupted dataset can be derived by multiplying the noise rate η of the original dataset by the probability that a noisy label is subsequently restored to its clean label due to the corrupting process, i.e., $\eta(\frac{1}{C-1})$. This derivation holds because the corruption process randomly flips class labels to one of the other classes uniformly. Consequently, the noise rate η_γ of the corrupted dataset is calculated as

$$\eta_\gamma = 1 - \eta \left(\frac{1}{C-1} \right). \tag{3}$$

Then, by the diagonally dominant condition, i.e., $\eta < 1 - \frac{1}{C}$, Eq. (3) implies

$$1 - \frac{1}{C} < \eta_\gamma \tag{4}$$

With this, we can derive that the corrupted dataset has a higher noise rate than the original dataset, i.e., $\eta < \eta_\gamma$. Besides, we present the formulation of the overall noise rate of the original and corrupted datasets as

$$\eta_{over} = \frac{\eta + \gamma \cdot \eta_\gamma}{1 + \gamma}. \tag{5}$$

D. Compatibility analysis with robust learning on Clothing 1M dataset

We also investigate the compatibility of DynaCor with various loss functions (GCE [19], and SCE [14]) and regularization technique (ELR [10]), specifically designed for noise robust learning. To this end, we measure the test accuracy of such noise robust classifiers trained using the original Clothing1M dataset and the cleansed dataset (i.e., the one with only correctly labeled instances identified by DynaCor), respectively.

In Table 7, we can observe consistent improvement in classification performance by cleansing the original dataset based on the detection results from DynaCor, even in case the classifier is trained with a noise-robust loss function or regularization technique.

Loss type	GCE [19]	SCE [14]	ELR [10]
Original	71.82	71.75	72.57
Cleansed	**72.23**	**72.37**	**73.06**

Table 7. Classification accuracy (%) on Clothing1M, trained with noise robust loss functions (GCE, SCE) and regularization technique (ELR) by using the original and cleansed sets, respectively.

References

[1] Wenkai Chen, Chuang Zhu, and Mengting Li. Sample prior guided robust model learning to suppress noisy labels. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 3–19. Springer, 2023. 2

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[3] 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. 1

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016. 1

[5] 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

[6] Taehyeon Kim, Jongwoo Ko, JinHwan Choi, Se-Young Yun, et al. Fine samples for learning with noisy labels. *Advances in Neural Information Processing Systems*, 34:24137–24149, 2021. 1, 2

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1, 2

[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

[10] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33:20331–20342, 2020. 3, 4

[11] Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems*, 33:17044–17056, 2020. 2

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1

[13] Reihaneh Torkzadehmahani, Reza Nasirigerdeh, Daniel Rueckert, and Georgios Kaissis. Label noise-robust learning using a confidence-based sieving strategy. *arXiv preprint arXiv:2210.05330*, 2022. 2

[14] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 322–330, 2019. 3, 4

[15] Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. *arXiv preprint arXiv:2110.12088*, 2021. 1

[16] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems*, 33:7597–7610, 2020. 1

[17] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699, 2015. 1

[18] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. 2

[19] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018. 3, 4