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

Abstract

Label noise, commonly found in real-world datasets, has a detrimental impact on a model's generalization. To effectively detect incorrectly labeled instances, previous works have mostly relied on distinguishable training signals, such as training loss, as indicators to differentiate between clean and noisy labels. However, they have limitations in that the training signals incompletely reveal the model's behavior and are not effectively generalized to various noise types, resulting in limited detection accuracy. In this paper, we propose DynaCor framework that distinguishes incorrectly labeled instances from correctly labeled ones based on the dynamics of the training signals. To cope with the absence of supervision for clean and noisy labels, DynaCor first introduces a label corruption strategy that augments the original dataset with intentionally corrupted labels, enabling indirect simulation of the model's behavior on noisy labels. Then, DynaCor learns to identify clean and noisy instances by inducing two clearly distinguishable clusters from the latent representations of training dynamics. Our comprehensive experiments show that DynaCor outperforms the state-of-the-art competitors and shows strong robustness to various noise types and noise rates.

1. Introduction

The remarkable success of deep neural networks (DNNs) is largely attributed to massive and accurately labeled datasets. However, creating such datasets is not only expensive but also time-consuming. As a cost-effective alternative, various methods have been employed for label collection, such as crowdsourcing [11] and extracting image labels from accompanying text on the web [27, 54]. Unfortunately, these approaches have led to the emergence of noise in real-world datasets, with reported noise rates ranging from 8.0% to 38.5% [25, 27, 54], which severely degrades the model's performance [1, 58].

*Corresponding authors

To cope with the detrimental effect of such noisy labels, a variety of approaches have been proposed, including noise robust learning that minimizes the impact of inaccurate information from noisy labels during the training process [7, 29, 49, 54] and data re-annotation through algorithmic methods [16, 39, 61]. Among them, the task of noisy label detection, which our work mainly focuses on, aims to identify incorrectly labeled instances in a training dataset [7, 22, 34]. This task has gained much attention in that it can be further utilized for improving the quality of the original dataset via cleansing or rectifying such instances.

Motivated by the *memorization effect*, which refers to the phenomenon where DNNs initially grasp simple and generalized patterns in correctly labeled data and then gradually overfit to incorrectly labeled data [1], most existing studies have utilized distinguishable training signals as indicators of label quality to differentiate between clean and noisy labels. To elaborate, these training signals are derived from the model's behavior on individual instances during the training [42, 44], involving factors such as training loss or confidence scores. Note that it is impractical to acquire annotations explicitly indicating whether each instance is correctly labeled or not. Hence, numerous studies have crafted various heuristic training signals [12, 19, 22], designed based on human prior knowledge of the model's distinctive behaviors when faced with clean and noisy labels.

Despite their effectiveness, the training signal-based detection methods still exhibit several limitations: (1) They only focus on a scalar signal at a single epoch (or a representative one across the entire training trajectory), which leads to limited detection accuracy (See Appendix B.2). Since the model's distinct behaviors on clean and noisy labels draw different temporal trajectories of training signals, a single scalar is insufficient to distinguish them by capturing temporal patterns within training dynamics. (2) Existing detection approaches based on heuristics are not effectively generalized to various types of label noise. Noisy labels can originate from diverse sources, including human annotator errors [33, 50], systematic biases [46], and unreliable annotations from web crawling [54], resulting in different noise types and rates for each dataset; this eventually requires considerable efforts to tune hyperparameters for training recipes of DNNs [26, 29, 45].

To tackle these challenges, our goal is to propose a fully data-driven approach that directly learns to distinguish the training dynamics of noisy labels from those of clean labels using a given dataset without solely relying on heuristics. The primary technical challenge in this data-driven approach arises from the absence of supervision for clean and noisy labels. As a solution, we introduce a *label corruption* strategy–image augmentation attaching intentionally corrupted labels via random label replacement. Since the augmented instances are highly likely to have incorrect labels, we can utilize them to capture the training dynamics of noisy labels. In other words, this allows us to simulate the model's behavior on noisy labels by leveraging the augmented instances with corrupted labels.

In this work, we present a novel framework, named DynaCor, that learns discriminative **Dyna**mics with label **Cor**ruption for noisy label detection. To be specific, DynaCor identifies clean and noisy labels via clustering of latent representations of training dynamics. To this end, it first generates training dynamics of original instances and corrupted instances. Then, it computes the dynamics representations that encode discriminative patterns within the training trajectories by using a parametric dynamics encoder. The dynamics encoder is optimized to induce two clearly distinguishable clusters (i.e., each for clean and noisy instances) based on two different types of losses for (1) high cluster cohesion and (2) cluster alignment between original and corrupted instances. Furthermore, DynaCor adopts a simple validation metric for the dynamics encoder based on the clustering quality so as to indirectly estimate its detection performance where ground-truth annotations of clean and noisy labels are not available for validation as well.

The contribution of this work is threefold as follows:

- We introduce a label corruption strategy that augments the original data with corrupted labels, which are highly likely to be noisy, enabling indirect simulation of the model's behavior on noisy labels during the training.
- We present a data-driven DynaCor framework to distinguish incorrectly labeled instances from correctly labeled ones via clustering of the training dynamics.
- Our extensive experiments on real-world datasets demonstrate that DynaCor achieves the highest accuracy in detecting incorrectly labeled instances and remarkable robustness to various noise types and noise rates.

2. Related Work

We provide a brief overview of the two primary research directions for addressing incorrectly labeled instances in a noisy dataset: (1) *Noisy label detection* focuses on identifying instances that are incorrectly labeled within a dataset, aiming to enhance data quality. (2) *Noise robust learning* is centered on developing learning algorithms and models that are resilient to the impact of noisy labels, ensuring robust performance even in the presence of labeling errors.

Noisy label detection. The main challenge in detecting noisy labels lies in defining a surrogate metric for label quality, essentially indicating how likely an instance is correctly labeled. The widely adopted option is the training loss, assessing the disparity between the model prediction and given labels [15, 19, 20], with higher loss often indicating incorrect labels. Various proxy measures, including gradient-based values [47, 60] and prediction-based metrics [31, 34, 39, 41] have been developed to differentiate between clean and noisy labels, utilizing methods like Gaussian mixture models [4, 22, 26, 64] or manually designed thresholds [15, 31, 34, 57, 63]. However, these approaches may overlook the potential benefits of adopting a data-driven (or learning-centric) detection model [7], which can be easily generalized to various noise types and levels. As a training-free alternative, a recent study [63] introduces a non-parametric KNN-based approach based on the assumption that instances situated closely in the input feature spaces derived from a pre-trained model are more likely to share the same clean label. However, its efficacy in detection heavily depends on the quality of the pre-trained model and may not be universally applicable across domains with specific fine-grained visual features.

Noise robust learning. Extensive research have focused on creating noise robust methods: loss functions [47, 60], regularization [6, 8, 29], model architectures [2, 5, 9, 13, 21, 54, 56], and training strategies [23, 30, 52, 59]. Recent studies have endeavored to integrate the process of detecting noisy labels and appropriately addressing them into the training pipeline in various ways: re-weighting losses [20, 36, 38] or re-annotation [16, 39, 61]. Besides, several studies [4, 26, 45, 51] treat detected noisy labels as unlabeled and make use of established semi-supervised techniques [3, 16, 59, 61]. Current robust learning typically relies on clean data, i.e., test data, for validation, while noisy detection methods can function without it, making direct comparisons difficult [63]. In this sense, we will discuss how these noise robust learning approaches can be effectively combined with noisy detection methods (Sec. 5.5).

3. Problem Formulation

For multi-class classification, let $\mathcal{X}$ be an input feature space and $\mathcal{Y} = \{1, 2, .., C\}$ be a label space. Consider a dataset $D = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$, where each sample is independently drawn from an unknown joint distribution over $\mathcal{X} \times \mathcal{Y}$. In real-world scenarios, we can only access a noisily labeled training set $\widetilde{D} = \{(\mathbf{x}_n, \tilde{y}_n)\}_{n=1}^N$, where $\tilde{y}$ denotes a noisy annotation, and there may exist $n \in \{1, ..., N\}$ such

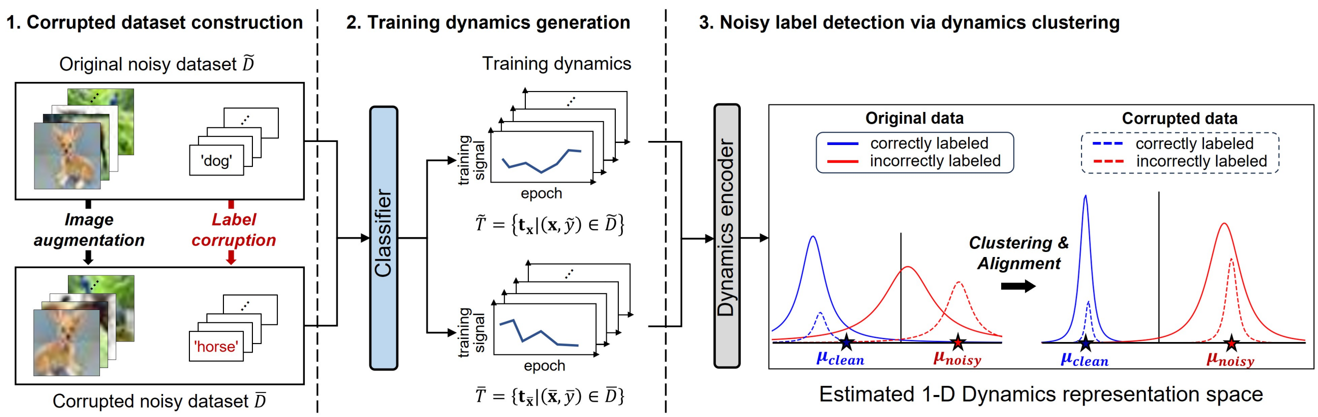


Figure 1. The proposed DynaCor framework consists of three steps: (1) Corrupted dataset construction generates the augmented images with corrupted labels, likely resulting in noisy labels, in order to provide guidance for discrimination between clean and noisy labels. (2) Training dynamics generation collects the trajectory of training signals for both the original and corrupted datasets by training a classifier. (3) Noisy label detection is performed by discovering two distinguishable clusters of dynamics representations, and for this, the dynamics encoder is optimized to enhance both cluster cohesion and alignment between the original and the corrupted datasets.

that $y_n \neq \tilde{y}_n$. In this work, we focus on the task of *noisy label detection*, which aims to identify the incorrectly labeled instances, i.e., $\{(\mathbf{x}_n, \tilde{y}_n) \in \widetilde{D} \mid y_n \neq \tilde{y}_n\}$. As an evaluation metric, we use F1 score [28], treating the incorrectly labeled instances as positive and the remainings as negative.

4. Methodology

4.1. Overview

DynaCor (**Dyna**mics learning with label **Cor**ruption for noisy label detection) framework learns discriminative patterns inherent in training dynamics, thereby distinguishing incorrectly labeled instances from clean ones. As illustrated in Figure 1, DynaCor consists of three major steps.

- Corrupted dataset construction (Sec. 4.2): To address the challenge arising from the lack of supervision for incorrectly labeled instances, we introduce a corrupted dataset that intentionally corrupts labels, providing guidance to identify incorrectly labeled instances.
- Training dynamics generation (Sec. 4.3): We generate training dynamics, which denote a model's behavior on individual instances during training, by training a classifier using both the original and the corrupted dataset.
- Noisy label detection via dynamics clustering (Sec. 4.4): We seek to discover underlying patterns in the training dynamics by learning representations that reflect the intrinsic similarities among data points, leveraging the characteristics of the corrupted dataset. For this, we encode the training dynamics via a dynamics encoder that learns discriminative representation using clustering and alignment losses. Then we find clusters using a robust validation metric designed for dynamics-based clustering.

4.2. Corrupted dataset construction

Given the original dataset $\widetilde{D}$, we construct a corrupted dataset $\bar{D}$ by intentionally corrupting labels for a randomly sampled subset of $\widetilde{D}$ with a corruption rate $\gamma \in (0, 1]$. Specifically, to obtain a corrupted instance $(\bar{\mathbf{x}}, \bar{y})$ from an original data instance $(\mathbf{x}, \tilde{y})$, we transform an input image using weak augmentation such as horizontal flip or center crop, i.e., $\bar{\mathbf{x}} = \mathrm{Aug}(\mathbf{x})$. Then, we randomly flip the class label to one of the other classes, i.e., $\bar{y} \in \{1, ..., C\} \setminus \{\tilde{y}\}$. The corrupted dataset, guaranteed to exhibit symmetric noise at a higher rate than the original, provides additional signals for discerning incorrectly labeled instances in the clustering process, as detailed in the following analysis.

Analysis: the noise rate of the corrupted dataset. We analyze the lower bound on the noise rate of the corrupted dataset $\bar{D}$. Let $\eta \in [0, 1]$ denote the noise rate of the original dataset $\widetilde{D}$.[1] Following the previous literature [14, 15, 40], we presume the *diagonally dominant condition*, i.e., $\Pr(\tilde{y} = i | y = i) > \Pr(\tilde{y} = j | y = i), \forall i \neq j$, which indicates that correct labels should not be overwhelmed by the false ones. With this condition of $\eta < 1 - \frac{1}{C}$, we have the following proposition.

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 $1 - \frac{1}{C}$.*

The proof is presented in Appendix C, from which we can derive $\eta < \eta_\gamma$.

[1] $\eta = \frac{1}{|\widetilde{D}|} |\{(\mathbf{x}, \tilde{y}) \in \widetilde{D} \mid \tilde{y} \neq y,\ (\mathbf{x}, y) \in D\}|$

4.3. Training dynamics generation

4.3.1 Training dynamics

The training dynamics indicates a model's behavior on individual instances during the training, quantitatively describing the training process [42, 44]. Concretely, the training dynamics is defined as the trajectory of training signals derived from a model's output across the training epochs. In the literature, various types of training signals [1, 42, 62] have been employed for analyzing the model's behavior.

Given a classifier f, let $f(\mathbf{x}) \in \mathbb{R}^C$ denote the output logits of an instance $\mathbf{x}$ for C classes. Let t be a transformation function that maps C logits to a scalar training signal. In this paper, we use *quantized logit difference* as the training signal.[2] It quantizes the difference between a logit [34] of a given label and the largest logit among the remaining classes, i.e., $t(f(\mathbf{x}), \tilde{y}) = \text{sign}(f_{\tilde{y}}(\mathbf{x}) - \max_{c \neq \tilde{y}} f_c(\mathbf{x}))$, where $f_c(\mathbf{x})$ denotes the logit for class c, and $\text{sign}(\mathbf{x}) = 1$ or -1 if $\mathbf{x} >= 0$ or < 0, respectively. The training dynamics for an instance $\mathbf{x}$ is defined as

$$\mathbf{t}_{\mathbf{x}} = [t^{(1)}(f(\mathbf{x}), \tilde{y}), .., t^{(E)}(f(\mathbf{x}), \tilde{y})], \tag{1}$$

where $t^{(e)}(f(\mathbf{x}), \tilde{y})$ denotes the training signal computed at epoch e, and E is the maximum number of training epochs. For the sake of convenience, we denote $\mathbf{t}_{\mathbf{x}}$ and $t^{(e)}_{\mathbf{x}}$ as an abbreviation for $\mathbf{t}(\mathbf{x}, \tilde{y}; f)$ and $t^{(e)}(f(\mathbf{x}), \tilde{y})$, respectively.

4.3.2 Dynamics generation for noisy label detection

We generate training dynamics for both the original and the corrupted datasets. Specifically, we train a classifier by minimizing the classification loss on $\widetilde{D}$ and $\bar{D}$:

$$\frac{1}{|\widetilde{D}|} \sum_{(\mathbf{x}, \tilde{y}) \in \widetilde{D}} \ell_{ce}(f(\mathbf{x}), \tilde{y}) + \frac{1}{|\bar{D}|} \sum_{(\bar{\mathbf{x}}, \bar{y}) \in \bar{D}} \ell_{ce}\left(f(\bar{\mathbf{x}}), \bar{y}\right), \tag{2}$$

where ℓ_{ce} is the softmax cross-entropy loss. For each instance $\mathbf{x}$, we obtain a training dynamics $\mathbf{t}_{\mathbf{x}} \in \mathbb{R}^E$ as specified in Eq. (1) by tracking $t^{(e)}_{\mathbf{x}}$ over the course of training epochs E. Training dynamics of the original and the corrupted datasets are denoted by $\widetilde{T} := \{\mathbf{t}_{\mathbf{x}} | (\mathbf{x}, \tilde{y}) \in \widetilde{D}\}$ and $\bar{T} := \{\mathbf{t}_{\bar{\mathbf{x}}} | (\bar{\mathbf{x}}, \bar{y}) \in \bar{D}\}$, respectively.

4.4. Noisy label detection via dynamics clustering

We use a clustering approach to identify incorrectly labeled instances within the original dataset. Using a dynamics encoder, we encode the generated dynamics and progressively find clusters of correctly and incorrectly labeled instances in the representation space. The dynamics clustering iterates two key processes: (1) identifications of incorrectly labeled instances (Sec. 4.4.1), and (2) learning distinct representations for each cluster (Sec. 4.4.2). The clustering quality is assessed by a newly introduced validation metric by leveraging the corrupted dataset without a clean validation dataset (Sec. 4.4.3).

4.4.1 Identification of incorrectly labeled instances

Cluster initialization. Given a training dynamics $\mathbf{t}_{\mathbf{x}}$, a dynamics encoder generates its representation, i.e., $\mathbf{z}_{\mathbf{x}} = \text{Enc}(\mathbf{t}_{\mathbf{x}}) \in \mathbb{R}^{d_{\mathbf{z}}}$. Let $\widetilde{Z}$ and $\bar{Z}$ denote the set of dynamics representations of the original and the corrupted datasets, respectively. We first introduce trainable parameters for centroids of noisy and clean clusters, i.e., $\boldsymbol{\mu}_{noisy}$, $\boldsymbol{\mu}_{clean} \in \mathbb{R}^{d_{\mathbf{z}}}$. We initialize $\boldsymbol{\mu}_{noisy}$ as the average representation of the corrupted instances $\bar{Z}$, while $\boldsymbol{\mu}_{clean}$ is initialized as the average representation of the original instances $\widetilde{Z}$. Note that this initialization is conducted only once at the beginning of the dynamics clustering step.

Noisy label identification. We determine whether each instance $\mathbf{x}$ has been incorrectly labeled based on its assignment probability to the noisy cluster. The assignment probability is computed based on the similarity between $\mathbf{z}_{\mathbf{x}}$ and the noisy cluster's centroid $\boldsymbol{\mu}_{noisy}$. We employ a kernel function based on the Student's t-distribution [43] with one degree of freedom as follows:

$$\begin{aligned} q_{noisy}(\mathbf{z}_{\mathbf{x}}) &= \frac{(1 + d(\mathbf{z}_{\mathbf{x}}, \boldsymbol{\mu}_{noisy}))^{-1}}{(1 + d(\mathbf{z}_{\mathbf{x}}, \boldsymbol{\mu}_{noisy}))^{-1} + (1 + d(\mathbf{z}_{\mathbf{x}}, \boldsymbol{\mu}_{clean}))^{-1}}, \\ q_{clean}(\mathbf{z}_{\mathbf{x}}) &= 1 - q_{noisy}(\mathbf{z}_{\mathbf{x}}), \end{aligned} \tag{3}$$

where $d(\mathbf{a}, \mathbf{b}) = 1 - \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{||\mathbf{a}||_2 \cdot ||\mathbf{b}||_2}$. Based on the assignment probability, we regard an instance as incorrectly labeled when its probability to the noisy cluster is predominant.

$$v(\mathbf{z}_{\mathbf{x}}) := \mathbb{1}[q_{noisy}(\mathbf{z}_{\mathbf{x}}) > q_{clean}(\mathbf{z}_{\mathbf{x}})], \tag{4}$$

$v(\mathbf{z}_{\mathbf{x}}) = 1$ indicates that $\mathbf{x}$ is predicted to have a noisy label.

4.4.2 Learning discriminative patterns in dynamics

We introduce the strategy of inducing two distinguishable clusters (each for correctly and incorrectly labeled instances) in the dynamics representation space. We propose two types of losses for (1) high cluster cohesion and (2) cluster alignment between original and corrupted instances.

Clustering loss. We introduce a clustering loss to make the clusters more distinguishable. We enhance cluster cohesion by adjusting each instance's representation to be closer to a centroid through a self-enhancing target distribution.

[2] We provide a detailed analysis of various training signals for identifying incorrectly labeled instances in Appendix B.3

The target distribution is constructed by amplifying the predicted assignment probability [55] as follows:

$$p_{noisy}(\mathbf{z_x}) = \frac{q^2_{noisy}(\mathbf{z_x})/s_{noisy}}{q^2_{noisy}(\mathbf{z_x})/s_{noisy} + q^2_{clean}(\mathbf{z_x})/s_{clean}},$$
$$p_{clean}(\mathbf{z_x}) = 1 - p_{noisy}(\mathbf{z_x}), \tag{5}$$

where $s_{noisy} = \sum_{\mathbf{z}\in\widetilde{Z}\cup\bar{Z}} q_{noisy}(\mathbf{z})$ and $s_{clean} = \sum_{\mathbf{z}\in\widetilde{Z}\cup\bar{Z}} q_{clean}(\mathbf{z})$. Then, we minimize the KL divergence between the cluster assignment distribution $\mathbf{q}(\mathbf{z_x}) = [q_{noisy}(\mathbf{z_x}),\, q_{clean}(\mathbf{z_x})]$ and the target distribution $\mathbf{p}(\mathbf{z_x}) = [p_{noisy}(\mathbf{z_x}),\, p_{clean}(\mathbf{z_x})]$ as follows:

$$\mathcal{L}_{cluster} = \sum_{\mathbf{z_x}\in\widetilde{Z}\cup\bar{Z}} \mathrm{KL}(\mathbf{p}(\mathbf{z_x})||\mathbf{q}(\mathbf{z_x})). \tag{6}$$

Alignment loss. We introduce an alignment loss that aligns the representation from each cluster's original and corrupted datasets. We hypothesize[3] that symmetric noise is relatively easy to identify among various noise types with diverse difficulty levels. Consequently, incorrectly labeled instances in the corrupted dataset exhibit more distinctive dynamics patterns than those in the original data, i.e., a red dashed line is farther away from blue lines than a red line in the 3rd step of Fig.1 (left). From this perspective, the mismatched noise types between the original and the corrupted datasets positively impact the clustering process by adopting alignment loss, which forces a red line to be aligned with a red dashed line in the 3rd step of Fig.1 (right).

Instances in the original dataset predicted as noisy and clean are denoted by $\widetilde{Z}_{noisy} = \{\mathbf{z_x} \in \widetilde{Z}|v(\mathbf{z_x}) = 1\}$ and $\widetilde{Z}_{clean} = \{\mathbf{z_x} \in \widetilde{Z}|v(\mathbf{z_x}) = 0\}$, respectively. Analogously, for the corrupted dataset, we obtain $\bar{Z}_{noisy} = \{\mathbf{z_x} \in \bar{Z}|v(\mathbf{z_x}) = 1\}$ and $\bar{Z}_{clean} = \{\mathbf{z_x} \in \bar{Z}|v(\mathbf{z_x}) = 0\}$. Then, we employ the alignment loss to reduce the discrepancy between the representations of the original dataset and the corrupted dataset as follows:

$$\mathcal{L}^n_{align} = d\Big(\frac{1}{|\widetilde{Z}_{noisy}|}\sum_{\mathbf{z_x}\in\widetilde{Z}_{noisy}}\mathbf{z_x},\, \frac{1}{|\bar{Z}_{noisy}|}\sum_{\mathbf{z_x}\in\bar{Z}_{noisy}}\mathbf{z_x}\Big),$$
$$\mathcal{L}^c_{align} = d\Big(\frac{1}{|\widetilde{Z}_{clean}|}\sum_{\mathbf{z_x}\in\widetilde{Z}_{clean}}\mathbf{z_x},\, \frac{1}{|\bar{Z}_{clean}|}\sum_{\mathbf{z_x}\in\bar{Z}_{clean}}\mathbf{z_x}\Big),$$
$$\mathcal{L}_{align} = \frac{1}{2}(\mathcal{L}^n_{align} + \mathcal{L}^c_{align}). \tag{7}$$

Optimization. To sum up, the dynamics encoder is optimized by minimizing the following loss:

$$\mathcal{L} = \mathcal{L}_{cluster} + \alpha\mathcal{L}_{align}, \tag{8}$$

where α is a hyperparameter that controls the impact of the alignment loss.

[3]It is theoretically proved in [32]

4.4.3 Validation metric

One practical challenge in training the dynamics encoder is determining an appropriate stopping point in the absence of ground-truth annotations of clean and noisy labels for validation. As a solution, we introduce a new validation metric for the dynamics encoder to estimate its detection performance indirectly. For noisy label detection, we aim to maximize (a) the assignment of incorrectly labeled instances to the noisy cluster while minimizing (b) the assignment of correctly labeled instances to the noisy cluster. Intuitively, in an ideally clustered space, the difference between (a) and (b) needs to be maximized.

Since we cannot access the ground-truth annotations to compute (a) and (b), we use the most representative instances as a workaround. Considering the corrupted dataset has a higher noise rate than the original dataset, we emulate (a) using instances predicted as noisy among the corrupted dataset, i.e., $\bar{Z}_{noisy}$. Similarly, (b) is emulated using instances predicted as clean among the original dataset with a lower noise rate, i.e., $\widetilde{Z}_{clean}$. Our validation metric is defined as the difference between two emulated values as

$$\Big(\sum_{\mathbf{z_x}\in\bar{Z}_{noisy}}\frac{q_{noisy}(\mathbf{z_x})}{|\bar{Z}_{noisy}|} - \sum_{\mathbf{z_x}\in\widetilde{Z}_{clean}}\frac{q_{noisy}(\mathbf{z_x})}{|\widetilde{Z}_{clean}|}\Big)^2. \tag{9}$$

The larger value indicates the better clustering quality for noisy label detection. Compared to the conventional metrics for assessing cluster separation [10, 37], this metric is tailored for our DynaCor framework and provides a more effective measure of noisy label detection efficacy.

5. Experiments

5.1. Experiment setup

Datasets. We evaluate the performance of DynaCor on benchmark datasets with different types of label noise, originating from diverse sources: (1) synthetic noise on CIFAR-10 and CIFAR-100 [24], (2) real-world human noise on CIFAR-10N and CIFAR-100N [50], and (3) systematic noise[4] on Clothing1M [54]. In the case of synthetic noise, following the previous experimental setup [63], we artificially introduce the noise by using different strategies with specific noise rates η as outlined below.

- **Symmetric Noise** (Sym., $\eta = 0.6$) randomly replaces the label with one of the other classes.
- **Asymmetric Noise** (Asym., $\eta = 0.3$) performs pairwise label flipping, where transition can only occur from a given class i to the next class $(i \bmod C) + 1$.
- **Instance-dependent Noise** (Inst., $\eta = 0.4$) changes labels based on the transition probability calculated using instance's corresponding features [53].

[4]In case of Clothing1M, systematic noise is induced by automatic annotation from the keywords present in the surrounding text of each image.

Dataset	CIFAR-10					CIFAR-100				
Noise type Noise rate (η)	Sym. 0.6	Asym. 0.3	Inst. 0.4	Agg. 0.09	Worst 0.4	Sym. 0.6	Asym. 0.3	Inst. 0.4	Human 0.4	Avg.
Avg.Encoder	**98.0** $\pm$ 0.03	89.7 $\pm$ 0.14	22.4 $\pm$ 33.5	67.3 $\pm$ 0.42	**92.8** $\pm$ 0.11	**96.7** $\pm$ 0.07	74.9 $\pm$ 0.17	76.8 $\pm$ 0.51	79.5 $\pm$ 0.31	77.6
AUM	95.7 $\pm$ 0.07	86.5 $\pm$ 0.18	81.9 $\pm$ 0.72	74.0 $\pm$ 0.16	88.7 $\pm$ 0.19	96.4 $\pm$ 0.10	74.7 $\pm$ 0.21	81.2 $\pm$ 0.25	74.6 $\pm$ 1.25	83.7
CL	96.6 $\pm$ 0.04	**94.0** $\pm$ 0.10	82.0 $\pm$ 0.21	68.6 $\pm$ 0.33	88.3 $\pm$ 0.11	88.0 $\pm$ 0.08	68.6 $\pm$ 0.16	75.9 $\pm$ 0.12	71.9 $\pm$ 0.10	81.5
CORES	97.7 $\pm$ 0.03	5.00 $\pm$ 0.33	19.2 $\pm$ 0.10	**80.5** $\pm$ 0.09	77.5 $\pm$ 0.09	83.9 $\pm$ 0.20	21.9 $\pm$ 0.32	36.7 $\pm$ 0.41	36.0 $\pm$ 0.12	50.9
SIMIFEAT-V	95.1 $\pm$ 0.06	89.4 $\pm$ 0.08	88.1 $\pm$ 0.11	79.6 $\pm$ 0.13	91.6 $\pm$ 0.06	86.0 $\pm$ 0.09	73.8 $\pm$ 0.07	80.5 $\pm$ 0.09	77.1 $\pm$ 0.12	84.6
SIMIFEAT-R	96.1 $\pm$ 1.41	88.9 $\pm$ 0.14	91.2 $\pm$ 0.07	79.6 $\pm$ 0.40	91.7 $\pm$ 0.35	90.3 $\pm$ 0.07	68.0 $\pm$ 0.10	77.3 $\pm$ 0.09	79.3 $\pm$ 0.11	84.7
DynaCor	**98.0** $\pm$ 0.04	**94.0** $\pm$ 0.15	**92.3** $\pm$ 0.38	79.6 $\pm$ 0.37	92.3 $\pm$ 0.19	94.3 $\pm$ 0.34	**76.3** $\pm$ 0.23	**81.7** $\pm$ 0.21	**80.4** $\pm$ 0.17	**87.7**

Table 1. Average F1 score (%) along with standard deviation across ten independent runs of DynaCor and baseline methods on CIFAR-10 and CIFAR-100. All methods except SIMIFEAT utilize the identical fixed image encoder from CLIP [35] and train only a subsequent MLP, while SIMIFEAT uses pre-trained CLIP as a feature extractor. The rightmost column averages the F1 scores across nine different settings. "Agg.", "Worst", and "Human" correspond to the real-world human label noises [50]. The best results are in **bold**.

Dataset	CIFAR-10					CIFAR-100				
Noise type	Sym.	Asym.	Inst.	Agg.	Worst	Sym.	Asym.	Inst.	Human	Avg.
Avg.Encoder	94.1 $\pm$ 0.14	85.4 $\pm$ 0.19	88.5 $\pm$ 0.20	63.6 $\pm$ 0.72	87.6 $\pm$ 0.18	**92.5** $\pm$ 0.34	75.2 $\pm$ 0.36	76.0 $\pm$ 0.49	**78.8** $\pm$ 0.18	82.4
AUM	75.4 $\pm$ 0.22	46.4 $\pm$ 0.30	57.7 $\pm$ 0.03	16.7 $\pm$ 0.01	57.8 $\pm$ 0.04	75.8 $\pm$ 0.21	46.7 $\pm$ 0.32	57.8 $\pm$ 0.10	58.0 $\pm$ 0.21	54.7
CL	88.7 $\pm$ 0.56	91.9 $\pm$ 0.12	82.5 $\pm$ 0.37	57.0 $\pm$ 0.31	80.0 $\pm$ 0.32	77.9 $\pm$ 0.39	62.4 $\pm$ 0.24	67.3 $\pm$ 0.28	65.2 $\pm$ 0.19	74.8
CORES	92.9 $\pm$ 0.17	26.7 $\pm$ 0.44	49.2 $\pm$ 1.15	63.6 $\pm$ 0.58	74.7 $\pm$ 0.36	66.3 $\pm$ 0.35	33.8 $\pm$ 0.46	39.2 $\pm$ 0.45	31.9 $\pm$ 0.48	53.2
SIMIFEAT-V	**94.6** $\pm$ 0.06	84.7 $\pm$ 0.17	83.7 $\pm$ 0.08	69.4 $\pm$ 0.17	88.3 $\pm$ 0.08	88.0 $\pm$ 0.09	70.3 $\pm$ 0.14	77.8 $\pm$ 0.10	76.2 $\pm$ 0.14	81.4
SIMIFEAT-R	92.9 $\pm$ 1.84	84.0 $\pm$ 0.13	86.9 $\pm$ 0.08	68.8 $\pm$ 0.32	**88.5** $\pm$ 0.36	89.7 $\pm$ 0.07	66.2 $\pm$ 0.11	75.5 $\pm$ 0.08	77.8 $\pm$ 0.13	81.2
DynaCor	93.6 $\pm$ 0.18	**94.2** $\pm$ 0.45	**91.5** $\pm$ 0.31	**72.6** $\pm$ 2.46	87.8 $\pm$ 0.37	91.3 $\pm$ 0.46	**79.2** $\pm$ 0.59	**79.5** $\pm$ 1.14	77.3 $\pm$ 0.54	**85.2**

Table 2. Average F1 score (%) under identical settings to those in Table 1 except for the backbone model. All methods except SIMIFEAT utilize a randomly initialized Renset34 [17], while SIMIFEAT uses a pre-trained ResNet34 on ImageNet [11] as a feature extractor.

In the case of human noise, we choose two noise subtypes for CIFAR-10N (denoted by Agg. and Worst) and a single noise subtype for CIFAR-100N (denoted by Human). More details of the datasets are presented in Appendix A.1.

Baselines. We compare DynaCor with various noisy label detection methods. All the methods except SIMIFEAT use training signals to identify incorrectly labeled instances.

- **Avg.Encoder** is a naive baseline that discriminates between clean and noisy labels by using a one-dimensional Gaussian mixture model [64] on the averaged training signals (i.e., logit difference) over the epochs.
- **AUM** [34] uses summation of training signals (i.e., logit difference) over the epochs and identifies correctly/incorrectly labeled instances based on a threshold.
- **CL** [31] uses a predicted probability of the given label (i.e., confidence) and filter out the instances with low confidence based on class-conditional thresholds.
- **CORES** [7] leverages a training loss for noisy label detection, progressively filtering out incorrectly labeled instances using its proposed sample sieve.
- **SIMIFEAT** [63] is a training-free approach that effectively detects noisy labels by utilizing K-nearest neighbors in the feature space of a pre-trained model.

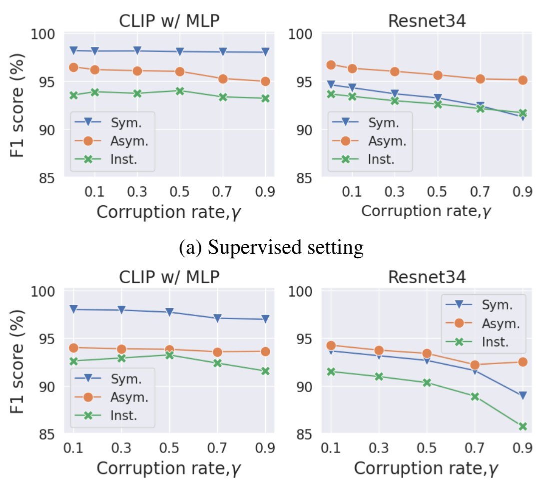
Implementation details. For our label corruption process, we use the corruption rate $\gamma = 0.1$ as the default. To generate the training dynamics, we employ DNN classifiers: ResNet34 [17] and the pre-trained ViT-B/32-CLIP [35] with a multi-layer perceptron (MLP) of two hidden layers. To encode the training dynamics, we use a three-layered 1D-CNN architecture [48] as the dynamics encoder. The hyperparameter α is selected as either 0.05 or 0.5. For more details about implementation, please refer to Appendix A.2.

5.2. Noisy label detection performance

We first evaluate DynaCor and the baseline methods for noisy label detection. Table 1 and Table 2 present their detection F1 scores for two classifiers, CLIP w/ MLP and ResNet34, across various noise types and rates. Notably, DynaCor achieves the best performance on average, i.e., +3.0% in Table 1 and +2.8% in Table 2, demonstrating its robustness to various types of noisy conditions. On the other hand, the baseline methods relying on training signals (i.e., Avg.Encoder, AUM, CL, and CORES) show considerable variations in performance across different noise types. For example, in the case of CIFAR-10, Avg.Encoder and CORES perform well for symmetric noises, whereas they struggle with identifying asymmetric or instance noises. It is worth noting that asymmetric and instance noise are more complex than symmetric noise in that they can have a more detrimental impact on model performance [32]. These results strongly support the superiority of our DynaCor framework in handling a wide range of label noise variations.

Validation metric	CIFAR-10 Inst.	CIFAR-10 Agg.	CIFAR-100 Inst.	CIFAR-100 Human
Max epoch	86.7 $\pm$ 6.75	77.8 $\pm$ 3.35	61.0 $\pm$ 10.3	64.3 $\pm$ 4.40
DBI	86.3 $\pm$ 8.75	76.7 $\pm$ 3.91	60.0 $\pm$ 10.2	64.8 $\pm$ 9.70
Ours	92.3 $\pm$ 0.38	79.6 $\pm$ 0.37	81.7 $\pm$ 0.21	80.4 $\pm$ 0.17
Opt epoch	92.6 $\pm$ 0.40	80.40 $\pm$ 0.44	81.8 $\pm$ 0.08	80.5 $\pm$ 0.18

Table 3. F1 score (%) of our dynamics encoder over various validation metrics on CIFAR-10 and CIFAR-100 using CLIP w/ MLP as a classifier.



(a) Supervised setting

(b) Unsupervised setting: DynaCor.

Figure 2. F1 score (%) changes with respect to corruption rate (γ) on CIFAR10 in supervised and unsupervised settings using CLIP w/ MLP (Left) and ResNet34 (Right) as classifiers.

5.3. Effectiveness of validation metric

To demonstrate the effectiveness of the proposed validation metric (Sec.4.4.3), we compare the detection performance of our dynamics encoder by employing our proposed metric and alternative criteria as stopping conditions during the training. **Max epoch** signifies the training over the maximum number of epochs. **Davies-Bouldin Index (DBI)** [10] assesses the quality of clustering results by calculating the ratio of intra-cluster distances to inter-cluster separations. A lower DBI value implies more compact and well-separated clusters, i.e., better clustering quality. In addition, **Opt epoch** selects the optimal training epoch that achieves the best detection results, providing the upper bound of detection performance.

In Table 3, our performance is close to the optimal case across various noise types and datasets, whereas Max epoch and DBI fail to stop the training process at a proper epoch on CIFAR-100. In conclusion, using the proper validation metric is critical for achieving competitive detection performance, particularly in the scenario where ground-truth annotations are not available for validation.

$\mathcal{L}_{cluster}$	$\mathcal{L}_{align}$	Asym.	Inst.	Agg.
		93.8 $\pm$ 0.17	91.8 $\pm$ 0.39	78.8 $\pm$ 0.37
✓		93.2 $\pm$ 0.11	**92.7** $\pm$ 0.36	76.8 $\pm$ 0.83
✓	✓	**94.0** $\pm$ 0.15	92.3 $\pm$ 0.38	**79.6** $\pm$ 0.37

Table 4. F1 score (%) of DynaCor that ablates the clustering and alignment loss on CIFAR10 using CLIP w/ MLP as a classifier. The first row reports the detection performance with a randomly initialized dynamics encoder.

5.4. Quantitative analyses

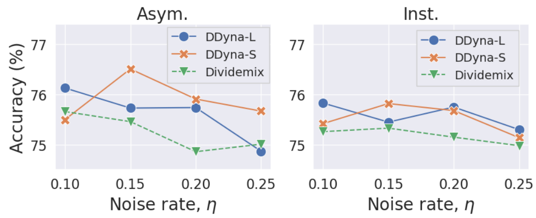
The effect of corruption rate. We analyze the effect of increasing the corruption rate, which in turn amplifies the overall noise level.[5] For thorough analyses, we conduct a controlled experiment within a supervised framework using classification,[6] assuming the availability of ground-truth annotations that indicate each instance as being correctly or incorrectly labeled. We then compare these results, generally regarded as the performance upper bound for unsupervised methods, with those obtained by an unsupervised approach. We focus on assessing the ability of our proposed unsupervised learning model, i.e., DynaCor, to discriminate training dynamics and how this discrimination is affected by increasing the overall noise level through corruption.

As shown in Figure 2, the detection F1 scores achieved by DynaCor (Figure 2b) approaches those of supervised learning (Figure 2a), demonstrating the effectiveness of training dynamics. This proximity is especially notable when utilizing a powerful image encoder, i.e., CLIP, which makes the training dynamics less susceptible to changes in the corruption rate. In contrast, the training dynamics from ResNet34 are more affected by increased corruption rate. Surprisingly, in the case of "Inst." type label noise, the training dynamics from the CLIP w/ MLP classifier become even more distinguishable as the corruption rate increases to 0.5. It shows that a higher noise rate in the training dataset can enhance the discernibility of the training dynamics. We hypothesize that the symmetric noise introduced through our label corruption process may reduce the overall difficulty of the detection task. This is consistent with the assertion in Sec. 4.4.2 that the symmetric noise is relatively straightforward to identify and, in turn, contributes to improving the performance of noisy label detection.

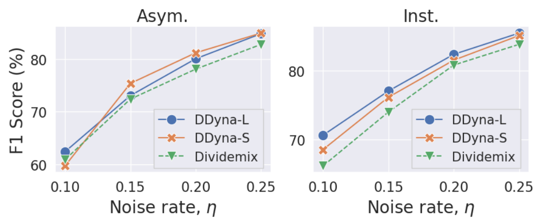
The effect of two losses. We examine the effect of the clustering and alignment losses within our DynaCor framework. In Table 4, both losses enhance detection performance. We also observe that the alignment loss effectively addresses the high imbalance between clean and noisy instances, particularly in scenarios with a low noise rate (e.g.,

[5] The overall noise rate is formulated as $\eta_{over} = \frac{\eta + \gamma \cdot \eta_\gamma}{1+\gamma}$.

[6] See Appendix B.1 for the details.



(a) Classification accuracy (%) of robust learning



(b) Noisy label detection F1 score (%)

Figure 3. Compatibility analysis of Dividemix with DynaCor on CIFAR100 over "Asym." and "Inst." with respect to noise rate

"Agg." on CIFAR-10). Given that DynaCor intentionally increases the noise rate by augmenting instances with corrupted labels, its benefits become more pronounced when dealing with datasets featuring a small original noise rate. In such cases, the alignment loss is crucial in stabilizing the clustering process by aligning the distinct distributions of original and corrupted instances.

5.5. Compatibility analyses with robust learning

We investigate the compatibility and synergistic effects of integrating our framework with various robust learning techniques: a semi-supervised approach (Dividemix [26]), loss functions (GCE [61] and SCE [47]), and a regularization method (ELR [29]). Detailed analyses of incorporating the loss functions and regularization technique on the Clothing1M dataset are provided in Appendix D.

For the semi-supervised approach, we select Dividemix [26] that iteratively detects incorrectly labeled instances and treats them as *unlabeled* instances. We construct integrated models of Dividemix and DynaCor through two distinct approaches: (1) **DDyna-L** is leveraging Dividemix to obtain the training dynamics of both original and corrupted datasets within our framework, and (2) **DDyna-S** is substituting the original detection method in Dividemix, i.e., GMM, with DynaCor. For the base architecture, we employ an 18-layer PreAct ResNet [18], adhering to its default optimization settings and hyperparameters, as specified in the original paper [26].

Classification accuracy. We explore the impact of our framework on the classifier's accuracy, specifically introducing a corrupted dataset (DDyna-L) and supplanting the existing noise detection method (DDyna-S). Figure 3a demonstrates that both enhance classification performance. In essence, results obtained with DDyna-L demonstrate that instances with symmetric label noise introduced through our corruption process prove beneficial for noise robust learning, especially in scenarios featuring a low noise rate in the original dataset, pointed out as a challenging setting for Dividemix [50].

Detection F1 score. To report the noisy label detection performance within robust learning framework, i.e., Dividemix and DDyna-S, we measure F1 score at every epoch and report the value when test classification accuracy is at its highest. Note that they leverage a clean test dataset to identify the optimal detection point; on the contrary, the noisy detection method (DDyna-L) operates without access to clean data, instead employing the procedure for model validation on the noisy dataset itself (Sec. 4.4.3), presenting a more challenging task. Figure 3b indicates that DDyna-S and DDyna-L further improves the detection F1 score of Dividemix, indicating the great compatibility of DynaCor with existing semi-supervised noise robust learning. In scenarios involving "Inst." label noise, DDyna-L exhibits compelling synergistic effects across a wide range of noise rates.

6. Conclusion

This paper proposes a new DynaCor framework that distinguishes incorrectly labeled instances from correctly labeled ones via clustering of their training dynamics. DynaCor first introduces a label corruption strategy that augments the original dataset with intentionally corrupted labels, enabling indirect simulation of the model's behavior on noisy labels. Subsequently, DynaCor learns to induce two clearly distinguishable clusters for clean and noisy instances by enhancing the cluster cohesion and alignment between the original and corrupted dataset. Furthermore, DynaCor adopts a simple yet effective validation metric to indirectly estimate its detection performance in the absence of annotations of clean and noisy labels. Our comprehensive experiments on real-world datasets demonstrate the detection efficacy of DynaCor, its remarkable robustness to various noise types and noise rates, and great compatibility with existing approaches to noise robust learning.

7. Acknowledgements

This work was supported by the IITP grant funded by the MSIT (No.2018-0-00584, 2019-0-01906, 2020-0-01361), the NRF grant funded by the MSIT (No.2020R1A2B5B03097210, RS-2023-00217286), and the Digital Innovation Hub project supervised by the Daegu Digital Innovation Promotion Agency (DIP) grant funded by the Korea government (MSIT and Daegu Metropolitan City) in 2024 (No. DBSD1-07).

References

[1] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR, 2017. 1, 4

[2] Alan Joseph Bekker and Jacob Goldberger. Training deep neural-networks based on unreliable labels. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2682–2686. IEEE, 2016. 2

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019. 2

[4] 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

[5] 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. 2

[6] De Cheng, Yixiong Ning, Nannan Wang, Xinbo Gao, Heng Yang, Yuxuan Du, Bo Han, and Tongliang Liu. Class-dependent label-noise learning with cycle-consistency regularization. *Advances in Neural Information Processing Systems*, 35:11104–11116, 2022. 2

[7] Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. *arXiv preprint arXiv:2010.02347*, 2020. 1, 2, 6

[8] Hao Cheng, Zhaowei Zhu, Xing Sun, and Yang Liu. Mitigating memorization of noisy labels via regularization between representations. *arXiv preprint arXiv:2110.09022*, 2021. 2

[9] Lele Cheng, Xiangzeng Zhou, Liming Zhao, Dangwei Li, Hong Shang, Yun Zheng, Pan Pan, and Yinghui Xu. Weakly supervised learning with side information for noisy labeled images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 306–321. Springer, 2020. 2

[10] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979. 5, 7

[11] 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, 6

[12] Mahsa Forouzesh and Patrick Thiran. Differences between hard and noisy-labeled samples: An empirical study. *arXiv preprint arXiv:2307.10718*, 2023. 1

[13] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *International conference on learning representations*, 2016. 2

[14] Xian-Jin Gui, Wei Wang, and Zhang-Hao Tian. Towards understanding deep learning from noisy labels with small-loss criterion. *arXiv preprint arXiv:2106.09291*, 2021. 3

[15] 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. *Advances in neural information processing systems*, 31, 2018. 2, 3

[16] Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5138–5147, 2019. 1, 2

[17] 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. 6

[18] 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. 8

[19] Jinchi Huang, Lie Qu, Rongfei Jia, and Binqiang Zhao. O2u-net: A simple noisy label detection approach for deep neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3326–3334, 2019. 1, 2

[20] 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

[21] Ishan Jindal, Matthew Nokleby, and Xuewen Chen. Learning deep networks from noisy labels with dropout regularization. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 967–972. IEEE, 2016. 2

[22] 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

[23] Taehyeon Kim, Jaehoon Oh, NakYil Kim, Sangwook Cho, and Se-Young Yun. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. *arXiv preprint arXiv:2105.08919*, 2021. 2

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[25] 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. 1

[26] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020. 2, 8

[27] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017. 1

[28] Zachary C Lipton, Charles Elkan, and Balakrishnan Naryanaswamy. Optimal thresholding of classifiers to maximize f1 measure. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II 14*, pages 225–239. Springer, 2014. 3

[29] 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. 1, 2, 8

[30] Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, pages 6448–6458. PMLR, 2020. 2

[31] Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021. 2, 6

[32] Diane Oyen, Michal Kucer, Nicolas Hengartner, and Har Simrat Singh. Robustness to label noise depends on the shape of the noise distribution. *Advances in Neural Information Processing Systems*, 35:35645–35656, 2022. 5, 6

[33] Joshua C Peterson, Ruairidh M Battleday, Thomas L Griffiths, and Olga Russakovsky. Human uncertainty makes classification more robust. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9617–9626, 2019. 1

[34] 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. 1, 2, 4, 6

[35] 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. 6

[36] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343. PMLR, 2018. 2

[37] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987. 5

[38] 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. *Advances in neural information processing systems*, 32, 2019. 2

[39] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *International Conference on Machine Learning*, pages 5907–5915. PMLR, 2019. 1, 2

[40] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*, 2014. 3

[41] Zeren Sun, Xian-Sheng Hua, Yazhou Yao, Xiu-Shen Wei, Guosheng Hu, and Jian Zhang. Crssc: salvage reusable samples from noisy data for robust learning. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 92–101, 2020. 2

[42] Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, 2020. 1, 4

[43] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9 (11), 2008. 4

[44] Haonan Wang, Wei Huang, Ziwei Wu, Hanghang Tong, Andrew J Margenot, and Jingrui He. Deep active learning by leveraging training dynamics. *Advances in Neural Information Processing Systems*, 35:25171–25184, 2022. 1, 4

[45] Haobo Wang, Ruixuan Xiao, Yiwen Dong, Lei Feng, and Junbo Zhao. Promix: combating label noise via maximizing clean sample utility. *arXiv preprint arXiv:2207.10276*, 2022. 2

[46] Jingkang Wang, Hongyi Guo, Zhaowei Zhu, and Yang Liu. Policy learning using weak supervision. *Advances in Neural Information Processing Systems*, 34:19960–19973, 2021. 1

[47] 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. 2, 8

[48] Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE, 2017. 6

[49] Hongxin Wei, Lue Tao, Renchunzi Xie, and Bo An. Open-set label noise can improve robustness against inherent label noise. *Advances in Neural Information Processing Systems*, 34:7978–7992, 2021. 1

[50] 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, 5, 6, 8

[51] Qi Wei, Haoliang Sun, Xiankai Lu, and Yilong Yin. Self-filtering: A noise-aware sample selection for label noise with confidence penalization. In *European Conference on Computer Vision*, pages 516–532. Springer, 2022. 2

[52] Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *International conference on learning representations*, 2020. 2

[53] 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. 5

[54] 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, 2, 5

[55] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016. 5

[56] Jiangchao Yao, Jiajie Wang, Ivor W Tsang, Ya Zhang, Jun Sun, Chengqi Zhang, and Rui Zhang. Deep learning from noisy image labels with quality embedding. *IEEE Transactions on Image Processing*, 28(4):1909–1922, 2018. 2

[57] 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. 2

[58] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021. 1

[59] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2

[60] 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. 2

[61] Zizhao Zhang, Han Zhang, Sercan O Arik, Honglak Lee, and Tomas Pfister. Distilling effective supervision from severe label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9294–9303, 2020. 1, 2, 8

[62] Tianyi Zhou, Shengjie Wang, and Jeffrey Bilmes. Curriculum learning by dynamic instance hardness. *Advances in Neural Information Processing Systems*, 33:8602–8613, 2020. 4

[63] Zhaowei Zhu, Zihao Dong, and Yang Liu. Detecting corrupted labels without training a model to predict. In *International conference on machine learning*, pages 27412–27427. PMLR, 2022. 2, 5, 6

[64] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *2011 international conference on computer vision*, pages 479–486. IEEE, 2011. 2, 6