Robust Learning by Self-Transition for Handling Noisy Labels

Hwanjun Song[1], Minseok Kim[2], Dongmin Park[2], Yooju Shin[2], Jae-Gil Lee[2,*]

[1]NAVER AI Lab [2]Korea Advanced Institute of Science and Technology

hwanjun.song@navercorp.com,{minseokkim,dongminpark,yooju24,jaegil}@kaist.ac.kr

ABSTRACT

Real-world data inevitably contains noisy labels, which induce the poor generalization of deep neural networks. It is known that the network typically begins to rapidly memorize false-labeled samples after a certain point of training. Thus, to counter the label noise challenge, we propose a novel *self-transitional* learning method called **MORPH**, which automatically switches its learning phase at the transition point from *seeding* to *evolution*. In the seeding phase, the network is updated using all the samples to collect a seed of clean samples. Then, in the evolution phase, the network is updated using only the set of arguably clean samples, which precisely keeps expanding by the updated network. Thus, MORPH effectively avoids the overfitting to false-labeled samples throughout the entire training period. Extensive experiments using five real-world or synthetic benchmark datasets demonstrate substantial improvements over state-of-the-art methods in terms of robustness and efficiency.

CCS CONCEPTS

• **Computing methodologies → Neural networks**; **Supervised learning by classification**.

KEYWORDS

Noisy Label, Label Noise, Robust Learning, Deep Learning

1 INTRODUCTION

In supervised learning for data analysis tasks, deep neural networks (DNNs) have become one of the most popular methods in that traditional machine learning is successfully superseded by recent deep learning in numerous applications [13, 37]. However, their success is conditioned on the availability of massive data with carefully annotated human labels, which are expensive and time-consuming to obtain in practice. Some substitutable sources,

---

*Jae-Gil Lee is the corresponding author.

such as Amazon's Mechanical Turk and surrounding tags of collected data, have been widely used to mitigate the high labeling cost, but they often yield samples with *noisy labels* that may not be true [35]. In addition, data labels can be extremely complex even for experts [7] and adversarially manipulated by a label-flipping attack [41], thereby being vulnerable to label noise.

Modern DNNs are typically trained in an over-parameterized regime where the number of the parameters of a DNN far exceeds the size of the training data [19]. In principle, such DNNs have the capacity to overfit to any given set of labels, even though part of the lables are significantly corrupted. In the presence of noisy labels, DNNs easily overfit to the entire training data regardless of the ratio of noisy labels, eventually resulting in poor generalization on test data [44]. Thus, in this paper, we address an important issue of "learning from noisy labels."

One of the most common approaches is *sample selection*, which involves training a DNN for a possibly clean subset of noisy training data [5, 9, 11, 12, 31, 34]. Typically, in each training iteration, a certain number of *small-loss* training samples are selected as clean ones and subsequently used to robustly update the DNN. This small-loss trick is satisfactorily justified by the *memorization effect* [2] that DNNs tend to first learn simple and generalized patterns and then gradually memorize all the patterns including irregular ones such as outliers and false-labeled samples.

Although this family of methods has achieved better generalization by training with selected small-loss training samples, they commonly have the following *two* problems:

1. **Discarding Hard Sample**: True-labeled samples with large losses are simply discarded by the small-loss trick, though they have a great impact on generalization [4]. This issue can be exacerbated by real-world and asymmetric label noises, where the loss distributions of true- and false-labeled samples are overlapped closely [35].
2. **Inefficient Learning**: The small-loss trick suffers from confirmation bias [38], which is a hazard of favoring the samples selected at the beginning of training. Hence, recent approaches often leverage multiple DNNs to cooperate with one another [9, 42] or run multiple training rounds to iteratively refine their selected set of clean samples [31, 33], thus adding heavy computational overhead.

In this regard, we have thoroughly investigated the memorization of a DNN on real-world noisy training samples and, consequently, observed the existence of *two* learning periods in Figure 1(a): **(i)** the "noise-robust" period where the memorization of false-labeled samples is insignificant because the DNN prefers memorizing easy samples at an early stage and **(ii)** the "noise-prone" period where the memorization of false-labeled samples rapidly increases because the DNN eventually begins memorizing all the noisy samples at a late stage of training.
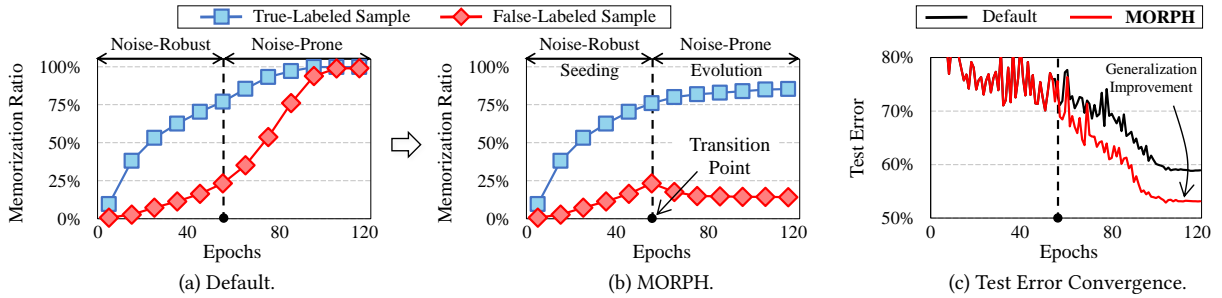
Figure 1: Key idea of MORPH: (a) and (b) show the memorization ratio when training a WideResNet-16-8 on a subset of FOOD-101N[1] with the real-world noise of 18.4%**, where the memorization ratio is the number of memorized (see Definition 3.1) true- or false-labeled samples to the total number of true- or false-labeled training samples at each epoch. "Default" is a standard training method, and "MORPH" is our proposed one; (c) contrasts the convergence of their test error.**

These findings motivate us to come up with an approach that leverages the transitional memorization nature in a single training round. In this paper, we propose **MORPH**, which is a *self-transitional* learning approach that automatically transitions its learning phase when a DNN enters the "noise-prone" period after the "noise-robust" period (i.e., the dashed line in Figure 1(b)). Thus, corresponding to these two periods, our key idea divides the training process into two learning phases, namely *seeding* and *evolution*:

1. **Seeding**: Owing to the negligible memorization of false-labeled samples, the model update is initiated using *all* the training samples in the noise-robust period. As the samples memorized at this time are mostly easy samples with true labels, they are accumulated as a *clean* seed to derive a *maximal safe set* in the next phase. Note that MORPH automatically estimates the best phase transition point without *any* supervision.

2. **Evolution**: Without memorizing false-labeled samples, the DNN evolves by being updated *only* for the maximal safe set in the noise-prone period. Then, the updated DNN recognizes more true-labeled samples previously hard to distinguish and filters out false-labeled samples incorrectly included. This alternating process repeats *per iteration* so that the maximal safe set is expanded and refined in the remaining noise-prone period.

Through self-transitional learning, MORPH avoids the confirmation bias by exploiting the noise-robust period with all the training samples, thus eliminating the need for additional DNNs and training rounds. In addition, it incrementally expands the clean seed towards the maximal safe set, which can cover even hard true-labeled samples, not just throwing them away. The alternating process in the second phase minimizes the risk of misclassifying false-labeled samples as true-labeled ones or vice versa. Hence, as shown in Figure 1(b), MORPH prevents the memorization of false-labeled samples by training with the maximal safe set during the noise-prone period, while increasing the memorization of true-labeled samples. As a result, as shown in Figure 1(c), the generalization performance of a DNN improves remarkably even in real-world noise. Our main contributions are summarized as follows:

- **No Supervision for Transition**: MORPH performs *self-transitional* learning without *any* supervision such as a true noise rate and a clean validation set, which are usually hard to acquire in real-world scenarios.

- **Noise Robustness**: Compared with state-of-the-art methods, MORPH identifies true-labeled samples from noisy data with much higher recall and precision. Thus, MORPH improved the test (or validation) error by up to $27.0pp$[2] for three datasets with two synthetic noises and by up to $8.90pp$ and $3.85pp$ for WebVision 1.0 and FOOD-101N with real-world noise.

- **Learning Efficiency**: Differently from other methods, MORPH requires neither additional DNNs nor training rounds. Thus, it was significantly faster than others by up to 3.08 times.

2 RELATED WORK

Numerous approaches have been proposed to address the challenge of learning from noisy labels. For a more thorough study on this topic, we refer the reader to surveys [7, 35].

Loss Correction: A typical method is using "loss correction," which estimates the label transition matrix and corrects the loss of the samples in a mini-batch. Bootstrap [28] updates the DNN based on its own reconstruction-based objective with the notion of perceptual consistency. F-correction [26] reweights the forward or backward loss of the training samples based on the label transition matrix estimated using a pre-trained DNN. D2L [23] employs a simple measure called local intrinsic dimensionality and then uses it to modify the forward loss in order to reduce the effects of false-labeled samples in learning. Ren et al. [29] included a small amount of clean validation data into the training data and reweighted the backward loss of the mini-batch samples such that the updated gradient minimized the loss of those validation data. However, this family of methods accumulates severe noise from the *false correction*, especially when the number of classes or the number of false-labeled samples is large [9].

Sample Selection: To be free from the false correction, many recent researches have adopted the method of "sample selection," which trains the DNN on selected samples. This method attempts to select the true-labeled samples from the noisy training data for updating the DNN. Decouple [25] maintains two DNNs simultaneously and updates the models by only using the samples with different label predictions from these two DNNs. Wang et al. [40] proposed an iterative learning framework that learns deep discriminative features from well-classified noisy samples based on the local

---

[1]We used the subset in which correct labels are identified.

[2]A pp is the abbreviation of a percentage point.

outlier factor algorithm. MentorNet [12] introduces a collaborative learning paradigm in which a pre-trained mentor DNN guides the training of a student DNN. Based on the small-loss criteria, the mentor provides the student with the samples whose labels are probably correct. Co-teaching [9] and Co-teaching+ [42] also maintain two DNNs, but each DNN selects a certain number of small-loss samples and feeds them to its peer DNN for further training. Compared with Co-teaching, Co-teaching+ further employs the disagreement strategy of Decouple. INCV [5] randomly divides the noisy training data and then employs cross-validation to classify true-labeled samples while removing large-loss samples at each training round. ITLM [31] iteratively minimizes the trimmed loss by alternating between selecting a fraction of small-loss samples at the current moment and retraining the DNN using them. SELFIE [33] trains a DNN on selectively refurbished samples together with small-loss samples. However, this family of methods suffers from the two challenges: **(i)** hard training samples with true labels are simply discarded; and **(ii)** the additional overhead for multiple DNNs or training rounds hinders them from being scalable to larger problems.

Other Directions: Beyond the supervised learning scope, Meta-Weight-Net [32] employs a meta-learning approach to reweight the loss of the training samples. Some other recent studies have attempted to combine unsupervised and semi-supervised learning. DM-DYR-SH [1] corrects the loss of training samples by modeling label noise in an unsupervised manner. DivideMix [18] treats small-loss training samples as labeled ones to adopt a semi-supervised learning technique called MixMatch [3].

Difference from Existing Work: Among these directions, our approach belongs to "sample selection" and is easily combined with meta-learning, unsupervised learning, and semi-supervised learning approaches because they are all *orthogonal* to ours. The main technical novelty of MORPH is to train the DNN with the maximal safe set, which is safely expanded from the clean seed through the notion of self-transitional learning. Furthermore, compared with prior studies on the memorization of DNNs [2, 19, 22], this is the *first* work to solve the problem of finding the best transition point.

3 PRELIMINARY

A k-class classification problem requires training data $\mathcal{D} = \{(x_i, y_i^*)\}_{i=1}^N$, where x_i is a sample and $y_i^* \in \{1, 2, \ldots, k\}$ is its *true* label. Following the label noise scenario, let's consider the noisy training data $\tilde{\mathcal{D}} = \{(x_i, \tilde{y}_i)\}_{i=1}^N$, where $\tilde{y}_i \in \{1, 2, \ldots, k\}$ is a *noisy* label which may not be true. Moreover, in conventional training, a mini-batch $\mathcal{B}_t = \{(x_i, \tilde{y}_i)\}_{i=1}^b$ comprises b samples drawn randomly from the noisy training data $\tilde{\mathcal{D}}$ at time t. Given a DNN parameterized by Θ, the DNN parameter Θ is updated according to the decent direction of the expected loss on the mini-batch via stochastic gradient descent,

$$\Theta_{t+1} = \Theta_t - \eta \nabla \Big(\frac{1}{|\mathcal{B}_t|} \sum_{(x, \tilde{y}) \in \mathcal{B}_t} f_{(x, \tilde{y})}(\Theta_t) \Big), \quad (1)$$

where η is the given learning rate and $f_{(x, \tilde{y})}(\Theta_t)$ is the loss of the sample $(x, \tilde{y})$ for the DNN parameterized by Θ_t.

As for the notion of DNN memorization, a sample x is defined to be *memorized* by a DNN if the majority of its recent predictions at time t coincide with the given label, as in Definition 3.1.

Definition 3.1. **(Memorized Sample)** Given a DNN Φ parameterized by Θ, let $\hat{y}_t = \Phi(x; \Theta_t)$ be the predicted label of a sample x at time t and $\mathcal{H}_x^t(q) = \{\hat{y}_{t_1}, \hat{y}_{t_2}, \ldots, \hat{y}_{t_q}\}$ be the history of the sample x that stores the predicted labels of the recent q epochs. Next, based on $\mathcal{H}_x^t(q)$, the probability of a label $y \in \{1, 2, \ldots k\}$ estimated as the label of the sample x is formulated by

$$p(y|x, t) = \frac{1}{|\mathcal{H}_x^t(q)|} \sum_{\hat{y} \in \mathcal{H}_x^t(q)} [\hat{y} = y],$$

$$\text{where } [S] = \begin{cases} 1, & \text{if } S \text{ is true.} \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

Subsequently, the sample x with its noisy label $\tilde{y}$ is defined to be *memorized* by the DNN with the parameter Θ_t at time t if $\text{argmax}_y p(y|x, t) = \tilde{y}$ holds. □

4 MORPH: SELF-TRANSITIONAL LEARNING

In this section, we propose MORPH which comprises the following two phases: **Phase I** for preparing the clean seed during the noise-robust period and **Phase II** for carefully evolving it towards the maximal safe set during the noise-prone period.

4.1 Phase I: Seeding

Phase I initiates to update the DNN using *all* the training samples in a conventional way of Eq. (1) during the noise-robust period, where the memorization of false-labeled samples is suppressed. Concurrently, because most of the samples memorized until the transition point are true-labeled, MORPH collects them to form a clean *seed*, which is used as an initial maximal safe set in Phase II. The major challenge here is to estimate the best transition point.

4.1.1 Best Transition Point. The DNN predominantly learns true-labeled samples until the noise-prone period begins. That is, at the best phase transition point, the DNN **(i)** not only acquires *sufficient* information from the true-labeled samples, **(ii)** but also accumulates *little* noise from the false-labeled ones. In that sense, we propose two memorization metrics, namely, *memorization recall (MR)* and *memorization precision (MP)* in Definition 4.1, which are indicators of evaluating the two properties, respectively.

Definition 4.1. **(Memorization Metrics)** Let $\mathcal{M}_t \subseteq \tilde{\mathcal{D}}$ be a set of memorized samples at time t according to Definition 3.1. Then, *memorization recall* and *precision* at time t are formulated by

$$MR(t) = \frac{|\{(x, \tilde{y}) \in \mathcal{M}_t : \tilde{y} = y^*\}|}{|\{(x, \tilde{y}) \in \tilde{\mathcal{D}} : \tilde{y} = y^*\}|},$$

$$MP(t) = \frac{|\{(x, \tilde{y}) \in \mathcal{M}_t : \tilde{y} = y^*\}|}{|\mathcal{M}_t|} \quad (3)$$

where $\mathcal{M}_t = \{(x, \tilde{y}) \in \tilde{\mathcal{D}} : \text{argmax}_y p(y|x, t) = \tilde{y}\}$. □

By Definition 4.1, the best transition point t_{tr} is naturally the moment when the recall and precision exhibit the highest value at the same time. This decision is also supported by the empirical understanding of memorization in deep learning that a better generalization of a DNN is achieved when pure memorization (i.e., high MP) and its enough amount (i.e., high MR) are satisfied simultaneously [19, 43]. However, it is not straightforward to find the best transition point without the ground-truth labels y^*.
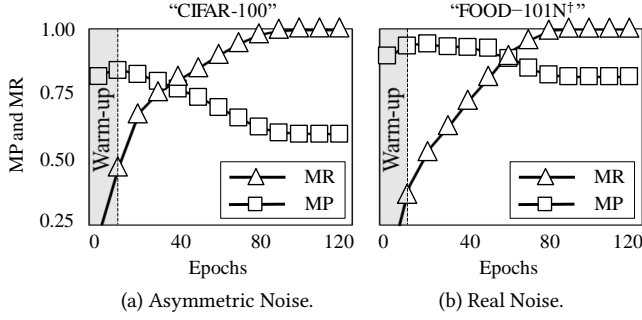
(a) Asymmetric Noise.   **(b) Real Noise.**

Figure 2: MR and MP when training WideResNet-16-8 on CIFAR-100 with the asymmetric noise of 40% **and FOOD-101N**[†] **with the real-world noise of** 18.4%**, where** † **indicates the subset in which correct labels are identified.**

Figure 2 shows the change in MR and MP on the two *realistic* label noises over the training period. These two metrics were observed to follow the *monotonicity* after a few warm-up epochs for the following reasons (see Section 4.1.2 for theoretical analysis):

- *MR* monotonically increases because a DNN eventually memorizes all the true-labeled samples as the training progresses.
- *MP* monotonically decreases because a DNN tends to memorize true-labeled samples first and then gradually memorizes all the false-labeled samples.

Under the monotonicity, the best transition point is the one *cross-point* of the two metrics, i.e., $MP(t) = MR(t)$, because it is the *best trade-off* between them. Then, by finding the answer of $MP(t) = MR(t)$ in Eq. (3), the best transition point t_{tr} satisfies

$$|\mathcal{M}_{t_{tr}}| = |\{(x, \tilde{y}) \in \tilde{\mathcal{D}} : \tilde{y} = y^*\}| = (1 - \tau)|\tilde{\mathcal{D}}|$$
$$\therefore |\mathcal{M}_{t_{tr}}| = (1 - \tau)|\tilde{\mathcal{D}}|, \tag{4}$$

where τ is the true noise rate. Because τ is typically unknown, it is *automatically* estimated by MORPH to check the condition in Eq. (4) for the phase transition.

Regarding the noise rate estimation, MORPH fits a two-component Gaussian Mixture Model (GMM) to model the loss distribution of true- and false-labeled samples because the distribution is *bi-modal* [1, 27]. At each epoch, MORPH accumulates the loss of all the training samples and fits the GMM to the accumulated loss by using the Expectation-Maximization (EM) algorithm. The probability of a sample x being false-labeled is obtained through its posterior probability. Accordingly, the noise rate τ is estimated by

$$\hat{\tau} = \mathbb{E}_{(x, \tilde{y}) \in \tilde{\mathcal{D}}}[p(g|f_{(x, \tilde{y})}(\Theta_t))], \tag{5}$$

where g is the Gaussian component with a larger mean (i.e., larger loss). A thorough analysis of estimating the noise rate is provided in Appendix A.

Therefore, MORPH transitions the learning phase at time t_{tr} when the number of memorized samples is greater than or equal to the estimated number of true-labeled ones (i.e., $|\mathcal{M}_{t_{tr}}| \geq (1 - \hat{\tau})|\tilde{\mathcal{D}}|$). Please note that MORPH requires *neither* a true noise rate *nor* a clean validation set.

4.1.2 Theoretical Analysis. We formally prove that MR and MP change monotonically over the training time t. Let us make an assumption that a set of memorized samples $\mathcal{M}_t$ grows as a loss value

decreases, which generally holds because the loss value implies how well a certain DNN predicts the label of training samples.

ASSUMPTION 4.2. *Let $f_{\tilde{\mathcal{D}}}(\Theta_t)$ be the expected loss of the DNN parameterized by Θ_t for all $(x, \tilde{y}) \in \tilde{\mathcal{D}}$. Then, it is assumed that*

$$f_{\tilde{\mathcal{D}}}(\Theta_t) \geq f_{\tilde{\mathcal{D}}}(\Theta_{t+1}) \rightarrow |\mathcal{M}_t| \leq |\mathcal{M}_{t+1}| \text{ and } \mathcal{M}_t \subseteq \mathcal{M}_{t+1}. \tag{6}$$

Subsequently, the monotonicity of MR and MP is proven by Theorems 4.3 and 4.4 under Assumption 4.2.

THEOREM 4.3. *Let the DNN be trained with gradient descent with a learning rate η. Then, $MR(t)$ is a monotonically increasing function that converges to 1 over the training time t.*

PROOF. Suppose that the loss function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a L-Lipschitz convex function. Then, the loss value monotonically decreases with gradient descent [30],

$$f_{\tilde{\mathcal{D}}}(\Theta_{t+1}) \leq f_{\tilde{\mathcal{D}}}(\Theta_t) - \frac{\eta}{2}||\nabla f_{\tilde{\mathcal{D}}}(\Theta_t)||_2^2 \leq f_{\tilde{\mathcal{D}}}(\Theta_t). \tag{7}$$

Hence, by Assumption 4.2, $|\mathcal{M}_t|$ increases monotonically from the update. That is, $\mathcal{M}_t$ gradually includes more true-labeled samples and eventually contains all of them. Thus, Eq. (8) holds.

$$MR(t) \leq MR(t + 1) \text{ and } \lim_{t \to \infty} MR(t) = 1. \quad \square \tag{8}$$

THEOREM 4.4. *Let the DNN be trained with gradient descent with a learning rate η. Then, $MP(t)$ is a monotonically decreasing function that converges to $(1 - \tau)$ over the training time t, where $\tau \in [0, 1]$ is the ratio of false-labeled samples in the training data.*

PROOF. Let $\mathcal{M}_t = C_t \cup \mathcal{R}_t$, where C_t and $\mathcal{R}_t$ are the sets of true- and false-labeled samples, respectively, memorized at time t. Then, $|\mathcal{R}_{t+1}|/|\mathcal{R}_t| \geq |C_{t+1}|/|C_t|$ because false-labeled samples are memorized rapidly as the training process progresses contrary to true-labeled samples after the DNN stabilizes by a few epochs [2]. Hence, $MP(t) \geq MP(t + 1)$ through the following derivation,

$$|\mathcal{R}_{t+1}| / |\mathcal{R}_t| \geq |C_{t+1}| / |C_t|$$
$$\Leftrightarrow |\mathcal{R}_{t+1}||C_t| + |C_{t+1}||C_t| \geq |C_{t+1}||\mathcal{R}_t| + |C_{t+1}||C_t|$$
$$\Leftrightarrow |C_t|(|C_{t+1}| + |\mathcal{R}_{t+1}|) \geq |C_{t+1}|(|C_t| + |\mathcal{R}_t|) \tag{9}$$
$$\Leftrightarrow |C_t| / |\mathcal{M}_t| \geq |C_{t+1}| / |\mathcal{M}_{t+1}| \Leftrightarrow MP(t) \geq MP(t + 1).$$

By gradient descent under Assumption 4.2, $\mathcal{M}_t$ eventually contains all the false-labeled samples of size $\tau|\tilde{\mathcal{D}}|$. Thus, Eq. (10) holds.

$$MP(t) \geq MP(t + 1) \text{ and } \lim_{t \to \infty} MP(t) = (1 - \tau). \quad \square \tag{10}$$

4.2 Phase II: Evolution

Phase II robustly updates the DNN *only* using the selected clean samples called the *maximal safe set* in Definition 4.5. By the definition of the transition point, the initial maximal safe set obtained in Phase I is quantitatively sufficient and qualitatively clean (i.e., high MR and MP); however, it can be further improved by **(i)** including more hard samples previously indistinguishable and **(ii)** filtering out false-labeled samples incorrectly included. In this regard, we take advantage of the fact that *undesired memorization* of false-labeled samples is easily forgotten by the robust update [8, 39]. Hence, MORPH robustly updates the DNN with the current maximal safe set $\mathcal{S}_t$ as in Eq. (11) and then derives a more refined set $\mathcal{S}_{t+1}$ as in Eq. (12).

Thus, the DNN keeps even more hard true-labeled samples C_{new} owing to the robust update, while forgetting the false-labeled samples R_{new} which were incorrectly memorized earlier. In this way, MORPH becomes better generalized to almost all true-labeled samples through the evolution of the safe set.

Definition 4.5. **(Maximal Safe Set)** A *maximal safe set* S_t is defined as the set of samples expected to be true-labeled at time $t(\geq t_{tr})$, where t_{tr} is the transition point. It starts from $S_{t_{tr}} = M_{t_{tr}}$, and is managed as follows:

1. The refinement of the maximal safe set S_t accompanies the update of the DNN parameter Θ_t. To be free from memorizing false-labeled samples in Phase II, the DNN parameter Θ_{t+1} is robustly learned only using the samples in S_t out of the mini-batch B_t by

$$\Theta_{t+1} = \Theta_t - \alpha \nabla \big(\frac{1}{|B_t \cap S_t|} \sum_{(x,\tilde{y}) \in (B_t \cap S_t)} f_{(x,\tilde{y})}(\Theta_t) \big). \quad (11)$$

2. Then, S_{t+1} reflects the changes by the DNN update at time $t+1$, i.e., newly memorized samples C_{new} and newly forgotten samples R_{new}, as formulated by

$$S_{t+1} = S_t + C_{new} - R_{new}, \text{ where}$$
$$C_{new} = \{(x,\tilde{y}) \in (B_t \cap S_t^c) : \arg\max_y p(y|x, t+1) = \tilde{y}\}, \quad (12)$$
$$R_{new} = \{(x,\tilde{y}) \in (B_t \cap S_t) : \arg\max_y p(y|x, t+1) \neq \tilde{y}\}. \quad \square$$

Furthermore, to avoid the potential risk of overfitting to a small *initial* seed, which is typically observed with a very high noise rate, MORPH adds *consistency regularization* $\mathcal{J}(\Theta_t)$ [16, 45] to the supervised loss in Eq. (11). Without relying on possibly unreliable labels, this regularization effectively helps learn the *dark knowledge* [10] from *all* the training samples by penalizing the prediction difference between the original sample x and its augmented sample $\hat{x}$. (For the experiments, $\hat{x}$'s were generated by random crops and horizontal flips.) Hence, the update rule is finally defined by

$$\Theta_{t+1} = \Theta_t - \alpha \nabla \big(\frac{1}{|B_t \cap S_t|} \sum_{(x,\tilde{y}) \in (B_t \cap S_t)} f_{(x,\tilde{y})}(\Theta_t) + w(t)\mathcal{J}(\Theta_t) \big),$$
$$\text{where } \mathcal{J}(\Theta_t) = \frac{1}{B_t} \sum_{(x,\tilde{y}) \in B_t} ||z(x;\Theta_t) - z(\hat{x};\Theta_t)||_2^2,$$
$$(13)$$

where $z(x;\Theta_t)$ is the softmax output of a sample x and $w(t)$ is a Gaussian ramp-up function to gradually increase the weight to the maximum value w_{max}. According to our ablation study in Section 5.6, the regularization further enhanced the robustness of MORPH when dealing with very high noise rates.

4.3 Algorithm Pseudocode

Algorithm 1 describes the overall procedure of MORPH, which is self-explanatory. First, the DNN is trained on the noisy training data $\tilde{D}$ in the *default* manner (Lines 5–7). During the first phase, the noise rate is estimated and subsequently used to find the moment when the noise-prone period begins. Here, if the transition condition holds, Phase I transitions to Phase II after assigning the clean seed (Lines 9–14). Subsequently, during the second phase, the mini-batch samples in the current maximal safe set are selected to update the DNN parameter with the consistency regularization (Lines 18–20). The rest mini-batch samples are excluded to pursue the robust

---

Algorithm 1 MORPH

INPUT: $\tilde{D}$: data, *epochs*: total number of epochs, q: history length, $\quad w_{max}$: maximum weight for $\mathcal{J}$
OUTPUT: Θ_t: DNN parameters, S_t: final safe set
1: $\quad t \leftarrow 1; S_t \leftarrow \emptyset; \Theta_t \leftarrow$ Initialize DNN parameters;
2: $\quad$ /* **I. Seeding during Noise-Robust Period** */
3: $\quad$ **for** $i = 1$ **to** *epochs* **do**
4: $\qquad$ **for** $j = 1$ **to** $|\tilde{D}|/|B_t|$ **do**
5: $\qquad\quad$ Draw a mini-batch B_t from $\tilde{D}$;
6: $\qquad\quad$ /* Standard update by Eq. (1) */
7: $\qquad\quad$ $\Theta_{t+1} = \Theta_t - \alpha \nabla \big(\frac{1}{|B_t|} \sum_{(x,\tilde{y}) \in B_t} f_{(x,\tilde{y})}(\Theta_t) \big);$
8: $\qquad\quad$ $t \leftarrow t + 1;$
9: $\qquad$ /* Noise rate estimation by Eq. (5) */
10: $\qquad$ $\hat{\tau} \leftarrow \mathbb{E}_{(x,\tilde{y}) \in \tilde{D}} [p(g|f_{(x,\tilde{y})}(\Theta_t))];$
11: $\qquad$ /* Checking the phase transition condition */
12: $\qquad$ **if** $|M_t| \geq (1 - \hat{\tau})|\tilde{D}|$ **then**
13: $\qquad\quad$ /* Assigning the initial maximal safe set */
14: $\qquad\quad$ $t_{tr} \leftarrow t; S_{t_{tr}} \leftarrow M_{t_{tr}};$ break;
15: $\quad$ /* **II. Evolution during Noise-Prone Period** */
16: $\quad$ **for** $i = t_{tr}|B_t|/|\tilde{D}| + 1$ **to** *epochs* **do**
17: $\qquad$ **for** $j = 1$ **to** $|\tilde{D}|/|B_t|$ **do**
18: $\qquad\quad$ Draw a mini-batch B_t from $\tilde{D}$;
19: $\qquad\quad$ /* Robust update by Eq. (13) */
20: $\qquad\quad$ $\Theta_{t+1} = \Theta_t - \alpha \nabla \big(\frac{1}{|B_t \cap S_t|} \sum_{(x,\tilde{y}) \in (B_t \cap S_t)} f_{(x,\tilde{y})}(\Theta_t)$
$\qquad\qquad\qquad +w(t)\mathcal{J}(\Theta_t) \big);$
21: $\qquad\quad$ /* Updating the maximal safe set by Eq. (12) */
22: $\qquad\quad$ $S_{t+1} \leftarrow S_t + C_{new} - R_{new};$
23: $\qquad\quad$ $t \leftarrow t + 1;$
24: $\quad$ **return** $\Theta_t, S_t;$

---

learning. Finally, the maximal safe set is refined by reflecting the change in DNN memorization resulting from the update (Lines 21–22). This alternating process is repeated until the end of the remaining learning period.

The main *additional* costs of MORPH are **(i)** the estimation of the noise rate (Line 10) and **(ii)** the additional inference step for the consistency regularization (Line 20). Because the noise rate is estimated using the EM algorithm once per epoch, its cost is negligible compared with that of the inference steps of a complex DNN. Thus, the additional inference in Phase II is the only part that increases the time complexity. Nevertheless, the additional cost is relatively *cheap* considering that other sample selection methods [5, 9, 31, 33, 42] require either multiple additional DNNs or multiple training rounds.

5 EVALUATION

Our evaluation was conducted to support the followings:

- MORPH is **much robust** and **efficient** than other state-of-the-art sample selection methods (Section 5.2 and 5.4).
- MORPH consistently identifies true-labeled samples from noisy data with **high recall** and **precision** (Section 5.3).
- The transition point estimated by MORPH is the **optimal point** for the best test error (Section 5.5).
- The consistency regularization is **much useful** at a very high noise rate (Section 5.6).
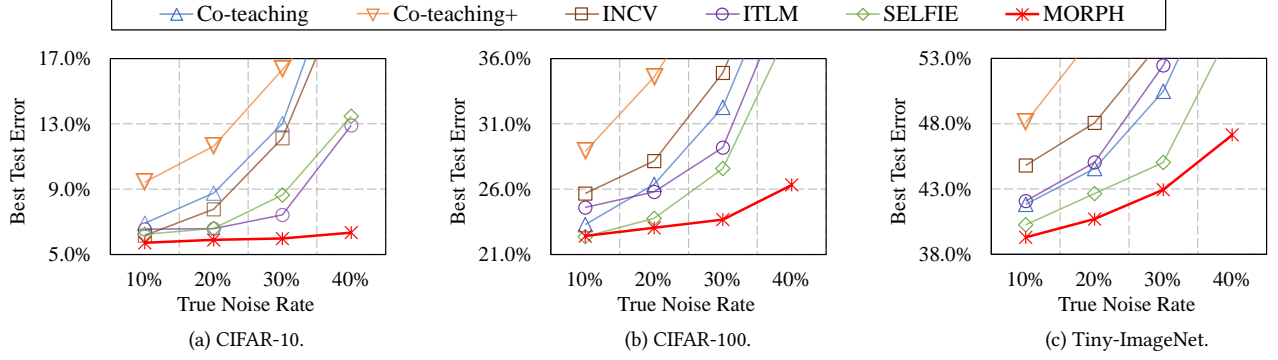
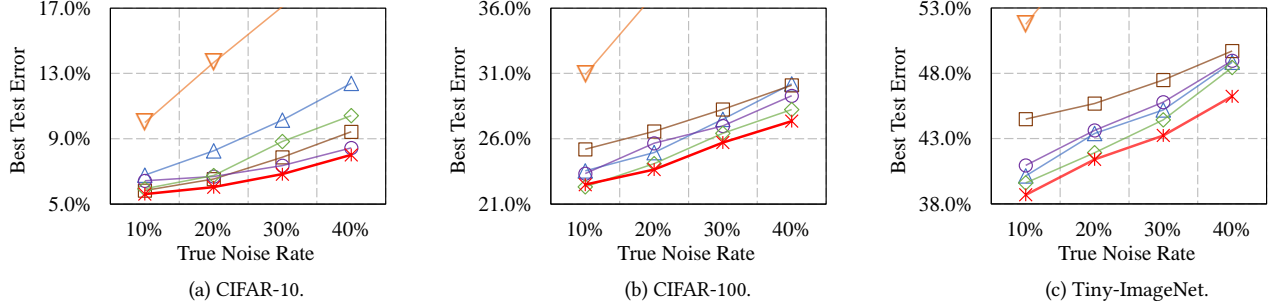**Figure 3: Best test errors on three datasets using WideResNet with varying asymmetric noise rates.**

Figure 4: Best test errors on three datasets using WideResNet with varying symmetric noise rates.

5.1 Experiment Setting

5.1.1 Datasets. To verify the superiority of MORPH, we performed an image classification task on five benchmark datasets: CIFAR-10 and CIFAR-100, a subset of 80 million categorical images [14]; Tiny-ImageNet [15], a subset of ImageNet ILSVRC12 [6]; WebVision 1.0 [21], real-world noisy data of large-scale web images crawled from Flickr and Google Images search; and FOOD-101N [17], real-world noisy data of crawled food images annotated by their search keywords in the Food-101 Taxonomy. Random crops and horizontal flips were applied for data augmentation. Please see Appendix C.1 for the details of the benchmark datasets.

5.1.2 Noise Injection. As all the labels in CIFAR and Tiny-ImageNet are clean, we artificially corrupted the labels in these datasets with two synthetic noises [9, 33, 42]. For k classes, we applied the label transition matrix $\mathbf{T}$: **(i)** *symmetric noise*: $\forall_{j \neq i} \mathbf{T}_{ij} = \frac{\tau}{k-1}$, where a true label is flipped into other labels with equal probability; **(ii)** *asymmetric noise*: $\exists_{j \neq i} \mathbf{T}_{ij} = \tau \wedge \forall_{k \neq i, k \neq j} \mathbf{T}_{ik} = 0$, where a true label is flipped into a certain label. Here, $\mathbf{T}_{ij}$ is the probability of the true label i being flipped to the corrupted label j, and τ is the noise rate. It is known that asymmetric noise is more realistic than symmetry noise because labelers may make mistakes only within very similar classes [33]. To evaluate the robustness on varying noise rates from light noise to heavy noise, we tested four noise rates $\tau \in \{0.1, 0.2, 0.3, 0.4\}$. In contrast, we did not inject any label noise into WebVision 1.0 and FOOD-101N because they contain real label noise whose rate is estimated at 20.0% and 18.4% [35].

5.1.3 Hyperparameter Configuration. MORPH requires two additional hyperparameters: the history length q and the maximum regularization weight w_{max}. We used the best history length $q = 10$

and maximum weight $w_{max} = 5.0$, which were obtained via a grid search in Appendix B.1. The w value gradually increased from 0 to w_{max} using a Gaussian ramp-up function as the DNN with a large w gets stuck in a degenerate solution at the beginning [16]. Similarly, the hyperparameters of the compared algorithms were configured to the best values, as detailed in Appendix C.3.

5.1.4 Reproducibility. For CIFARs and Tiny-ImageNet, we trained a WideResNet-16-8 from scratch using SGD with a momentum of 0.9, a batch size of 128, a dropout of 0.1, and a weight decay of 0.0005. The DNN was trained for 120 epochs with an initial learning rate of 0.1, which was decayed with cosine annealing.

Regarding WebVision 1.0 and FOOD-101N, which contain the real-world noise, we followed exactly the same configuration in the previous work. For WebVision 1.0, we trained an Inception ResNet-V2 [36] from scratch for the first 50 classes of the Google image subset [5]; for FOOD-101N, we fine-tuned a ResNet-50 with the ImageNet pretrained weights for the entire training set [17]. Please see Appendix C.2 for the details of the configurations.

All the algorithms were implemented using TensorFlow 2.1.0 and executed using 16 NVIDIA Titan Volta GPUs. Note that the compared algorithms were re-implemented by us for fair comparison. In support of reliable evaluation, we repeated every task *thrice* and reported the average test (or validation) error as well as the average training time. The source code and trained models are publicly available at https://github.com/kaist-dmlab/MORPH.

5.2 Robustness Comparison

5.2.1 Results with Synthetic Noise. We compared MORPH with the five state-of-the-art sample selection methods using the small-loss trick: *Co-teaching* [9], *Co-teaching+* [42], *INCV* [5], *ITLM* [31],

Table 1: Comparison with state-of-the-art methods trained on Webvision 1.0. The value outside (inside) the parentheses denotes the top-1 (top-5) classification error (%) on the WebVision validation set and the ImageNet ILSVRC12 validation set. The results for baseline methods are borrowed from [5].

Method	WebVision Val.	ILSVRC12 Val.
F-correction [26]	38.88 (17.32)	42.64 (17.64)
Decouple [25]	37.46 (15.26)	41.74 (17.74)
Co-teaching [9]	36.42 (14.80)	38.52 (15.30)
MentorNet [12]	37.00 (18.60)	42.20 (20.08)
D2L [23]	37.32 (16.00)	42.20 (18.64)
INCV [5]	34.76 (14.66)	38.40 (15.02)
MORPH	**29.98 (11.33)**	**33.09 (12.62)**

Table 2: Comparison with state-of-the-art methods trained on FOOD-101N. The value denotes the top-1 classification error (%) on the FOOD-101 validation set. The results for baseline methods are borrowed from [17, 20]. † indicates that extra clean (or verification) labels were used for supervision.

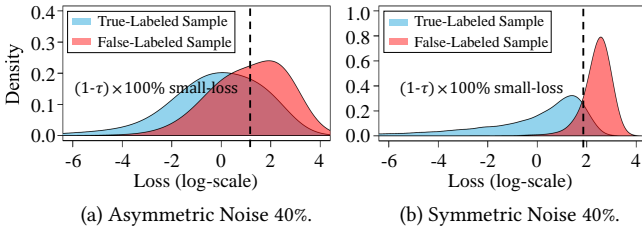| Method	FOOD-101 Val.
Cross-Entropy [17]	18.56
Weakly Supervised [46]	16.57
CleanNet $(w_{hard})^{\dagger}$ [17]	16.53
CleanNet $(w_{soft})^{\dagger}$ [17]	16.05
Guidance Learning† [20]	15.80
MORPH	**14.71**

Figure 5: Loss distribution of true- and false-labeled samples at the training accuracy of 50% **on noisy CIFAR-100.**

and *SELFIE* [33]. In addition, the comparison with *DM-DYR-SH* [1] and *DivideMix* [18], which are in fact not directly comparable with MORPH, is discussed in Appendix B.2. Figures 3 and 4 show the test errors of the six sample selection methods with varying *asymmetric* and *symmetric* noise rates. See Appendix D.1 for the tabular reports.

Asymmetric Noise: MORPH generally achieved the lowest test errors with respect to a wide range of noise rates. The error reduction became larger as the noise rate increased, reaching $6.6pp$–$27.0pp$ at a heavy noise rate of 40%. In contrast, the performance of the other methods worsened rapidly with an increase in the noise rate. As shown in Figure 5(a), because the loss distributions of true- and false-labeled samples are closely overlapped in asymmetric noise, the philosophy of selecting small-loss samples could not distinguish well true-labeled samples from false-labeled samples. Thus, the other methods simply discard a lot of hard training samples even if they are true-labeled. On the other hand, MORPH gradually includes them by its alternating process in Phase II.

Symmetric Noise: MORPH generally outperformed the other methods again, though the error reduction was relatively small, i.e., $0.42pp$–$26.3pp$ at a heavy noise rate of 40%. The small-loss trick was turned out to be appropriate for symmetric noise because the loss distributions are clearly separated, as illustrated in Figure 5(b).

Putting them together, we contend that only MORPH realizes *noise type robustness*, evidenced by consistently low test errors in both asymmetric and symmetric noises.

5.2.2 Results with Real-World Noise. Tables 1 and 2 summarize the results on Webvision 1.0 and FOOD-101N. MORPH maintained its dominance over multiple state-of-the art methods for *real-world*
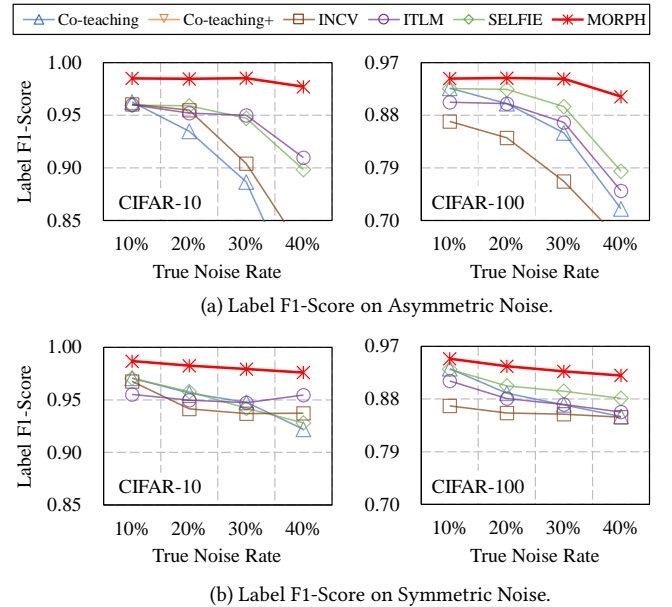

Figure 6: Label F1-Scores on two CIFARs data using WideResNet with varying noise rates.

label noise as well. It improved the top-1 validation error by $4.44pp$–$8.90pp$ and $1.09pp$–$3.85pp$ in Webvision 1.0 and FOOD-101N, respectively. The lowest error of MORPH in FOOD-101N was achieved even without extra supervision from the verification labels.

5.3 In-depth Analysis on Selected Samples

5.3.1 F1-Score of Selected Clean Samples. The superior robustness of MORPH is attributed to its high *label recall* (LR) and *precision* (LP), which are calculated by replacing $\mathcal{M}$ for MR and MP in Eq. (3) with the set of selected clean samples [9]. LR and LP are the performance indicators that respectively represent the quantity and quality of the samples selected as true-labeled ones [9]. Hence, we compare MORPH with the other five sample selection methods in terms of the *label F1-score* $= (2 \cdot LP \cdot LR)/(LP + LR)$.

To evaluate the label F1-score, MORPH used its final maximal safe set, *Co-teaching(+)* and *SELFIE* used the samples selected during their last epoch, and *INCV* and *ITLM* used the samples selected
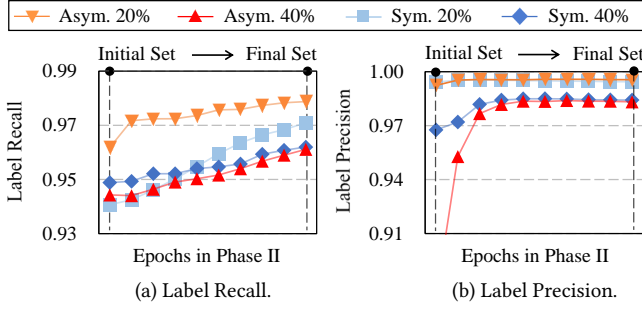
Figure 7: Evolution of the maximal safe set in Phase II using WideResNet on CIFAR-10 with two synthetic noises.

Table 3: Best test errors (%) of MORPH with early or late transition based on the best point, where α is added to the estimated noise rate in Eq. (4) to force early or late transition. CIFARs with two synthetic noises of 40% were used.

Transition Point	Early		Best	Late	
A value α	+10%	+5%	+0%	-5%	-10%
CIFAR-10 (Sym.)	8.71	8.37	**8.01**	9.39	9.75
CIFAR-10 (Asym.)	7.53	**6.20**	6.34	7.43	8.08
CIFAR-100 (Sym.)	29.8	28.5	**27.4**	28.1	28.5
CIFAR-100 (Asym.)	28.3	27.2	**26.3**	27.0	28.9

for their final training round. Figure 6 shows their label F1-scores. Only MORPH achieved consistently high label F1-scores of over 0.91 in all the cases. This result corroborates that MORPH identifies true-labeled samples with high recall and precision regardless of the noise type and rate.

5.3.2 Evolution of the Maximal Safe Set. Figure 7 shows the LR and LP values on the maximal safe set obtained at each epoch since Phase II begins. At the beginning of Phase II, both LR and LP already exhibited fairly high values because the initial set is derived from the samples memorized at the transition point, which is the best compromise between MR and MP. Moreover, as the training in Phase II progressed, LR increased gradually by adding more hard true-labeled ones previously indistinguishable (i.e., C_{new}), and LP also increased gradually by filtering out false-labeled samples incorrectly included (i.e., $\mathcal{R}_{new}$). Notably, their improvement was consistently observed regardless of the noise type and rate, thereby achieving remarkably high LR and LP at the end. The high F1-score of MORPH in Figure 6 is well supported by this evolution of the maximal safe set.

5.4 Training Efficiency Comparison

Another advantage of MORPH is its efficiency in training the DNN. Differently from the other methods that require additional DNNs or training rounds, MORPH needs only a single training round for a single DNN. Figure 8 shows the training time of the six sample selection methods as well as the *non*-robust method ("default") on two CIFAR datasets. MORPH added only 24.2–38.0% overhead, mostly caused by the consistency regularization, compared with the non-robust method. Among the six robust methods, MORPH was the fastest in each case because of its relatively cheap additional
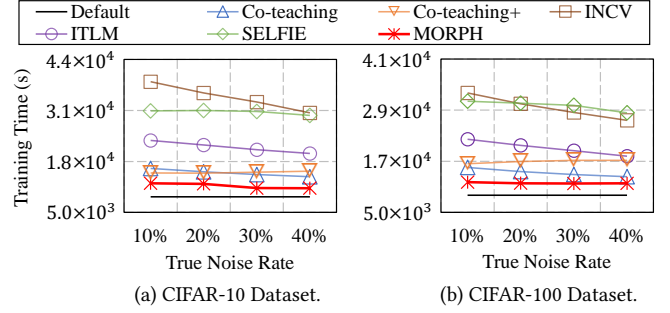

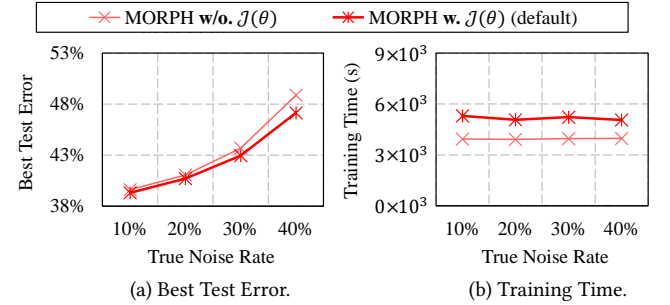Figure 8: Training time on CIFAR-10 and CIFAR-100 datasets with asymmetric noise.

Figure 9: Effect of the consistency loss $\mathcal{J}(\theta)$ on Tiny-ImageNet with asymmetric noise.

costs. Overall, MORPH was 1.13–3.08 times faster than the other methods. The difference between the training time of the robust baselines tended to be determined by the total number of training rounds, which is represented by one of their hyperparameters. See Table 6 in Appendix D.2 for the result of all datasets.

5.5 Optimality of the Best Transition Point

It is of interest to verify the optimality of the best transition point. Hence, we investigated the performance change of MORPH when forcing early or late transition. Table 3 shows the test errors of MORPH with early or late transition based on the estimated best transition point. The best test error was generally achieved at the estimated best point other than those around the best point. The more deviated a transition point from the best point, the higher the test error. Overall, this empirical result confirms the optimality of the best transition point estimated by MORPH.

5.6 Effect of Consistency Regularization

Figure 9 shows the effect of the consistency regularization. Interestingly, as shown in Figure 9(a), the test error of MORPH was further improved by adding the consistency loss in Eq. (13). The higher the noise rate, the greater the benefit, because the overfitting issue caused by a smaller size of the initial maximal safe set is mitigated. Regarding the training time, as shown in Figure 9(b), the regularization slightly slowed down the training speed owing to the extra inference steps. Because MORPH achieved lower test error than the other robust methods even *without* the regularization (see Appendix D.1 for details), practitioners may skip employing the regularization if their time budgets are restricted.

6 CONCLUSION

We proposed a novel *self-transitional* learning scheme called **MORPH** for noisy training data. The first phase exploits all training samples and estimates the optimal transition point. The second phase completes the rest of the training process using only a maximal safe set with high label recall and precision. MORPH can be easily applied to many real-world cases because it requires neither a true noise rate nor a clean validation set. Through extensive experiments using various real-world and simulated noisy datasets, we verified that MORPH consistently exhibited significant improvement in both robustness and efficiency compared with state-of-the-art methods. Overall, we believe that the division of the training process into two phases unveils a new approach to robust training and can inspire subsequent studies.

ACKNOWLEDGEMENT

REFERENCES

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin Mcguinness. 2019. Unsupervised label noise modeling and loss correction. In *ICML*. 312–321.

[2] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *ICML*. 233–242.

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019. MixMatch: A holistic approach to semi-supervised learning. In *NeurIPS*. 5049–5059.

[4] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. 2017. Active Bias: Training more accurate neural networks by emphasizing high variance samples. In *NeurIPS*. 1002–1012.

[5] Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. 2019. Understanding and utilizing deep neural networks trained with noisy labels. In *ICML*. 1062–1070.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*. 248–255.

[7] Benoît Frénay and Michel Verleysen. 2013. Classification in the presence of label noise: A Survey. *IEEE Transactions on Neural Networks and Learning Systems* 25, 5 (2013), 845–869.

[8] Bo Han, Gang Niu, Xingrui Yu, Quanming Yao, Miao Xu, Ivor Tsang, and Masashi Sugiyama. 2020. SIGUA: Forgetting may make learning with noisy labels more robust. In *ICML*. 4006–4016.

[9] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*. 8536–8546.

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *CoRR* (2015).

[11] Jinchi Huang, Lie Qu, Rongfei Jia, and Binqiang Zhao. 2019. O2U-Net: A simple noisy label detection approach for deep neural networks. In *ICCV*. 3326–3334.

[12] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*. 2309–2318.

[13] Mohammad Mahdi Kamani, Sadegh Farhang, Mehrdad Mahdavi, and James Z Wang. 2020. Targeted data-driven regularization for out-of-distribution generalization. In *KDD*. 882–891.

[14] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. 2014. CIFAR-10 and CIFAR-100 datasets. https://www.cs.toronto.edu/~kriz/cifar.html.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *NeurIPS*. 1097–1105.

[16] Samuli Laine and Tim Aila. 2017. Temporal ensembling for semi-supervised learning. In *ICLR*.

[17] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. 2018. CleanNet: Transfer learning for scalable image classifier training with label noise. In *CVPR*. 5447–5456.

[18] Junnan Li, Richard Socher, and Steven CH Hoi. 2020. DivideMix: Learning with noisy labels as semi-supervised learning. In *ICLR*.

[19] Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. 2020. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *AISTATS*. 4313–4324.

[20] Qing Li, Xiaojiang Peng, Liangliang Cao, Wenbin Du, Hao Xing, Yu Qiao, and Qiang Peng. 2020. Product image recognition with guidance learning and noisy supervision. *Computer Vision and Image Understanding* (2020), 102963.

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

[22] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. 2020. Early-learning regularization prevents memorization of noisy labels. In *NeurIPS*.

[23] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah M Erfani, Shu-Tao Xia, Sudanthi Wijewickrema, and James Bailey. 2018. Dimensionality-driven learning with noisy labels. In *ICML*. 3361–3370.

[24] Zhanyu Ma and Arne Leijon. 2011. Bayesian estimation of beta mixture models with variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 11 (2011), 2160–2173.

[25] Eran Malach and Shai Shalev-Shwartz. 2017. Decoupling "when to update" from "how to update". In *NeurIPS*. 960–970.

[26] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*. 2233–2241.

[27] Geoff Pleiss, Tianyi Zhang, Ethan R. Elenberg, and Kilian Q. Weinberger. 2020. Detecting noisy training data with loss curves. https://openreview.net/forum?id=HyenUkrtDB

[28] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2015. Training deep neural networks on noisy labels with bootstrapping. In *ICLR*.

[29] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *ICML*. 4334–4343.

[30] Ohad Shamir and Tong Zhang. 2013. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *ICML*. 71–79.

[31] Yanyao Shen and Sujay Sanghavi. 2019. Learning with bad training data via iterative trimmed loss minimization. In *ICML*. 5739–5748.

[32] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. In *NeurIPS*. 1917–1928.

[33] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. 2019. SELFIE: Refurbishing unclean samples for robust deep learning. In *ICML*. 5907–5915.

[34] Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. 2020. How does early stopping help generalization against label noise?. In *ICMLW*.

[35] Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. 2020. Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199* (2020).

[36] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*.

[37] Yina Tang, Fedor Borisyuk, Siddarth Malreddy, Yixuan Li, Yiqun Liu, and Sergey Kirshner. 2019. MSURU: Large scale e-commerce image classification with weakly supervised search data. In *KDD*. 2518–2526.

[38] Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*. 1195–1204.

[39] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. 2019. An empirical study of example forgetting during deep neural network learning. In *ICLR*.

[40] Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. 2018. Iterative learning with open-set noisy labels. In *CVPR*. 8688–8696.

[41] Han Xiao, Huang Xiao, and Claudia Eckert. 2012. Adversarial label flips attack on support vector machines. In *ECAI*. 870–875.

[42] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. 2019. How does disagreement help generalization against label corruption?. In *ICML*. 7164–7173.

[43] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Michael C Mozer, and Yoram Singer. 2020. Identity Crisis: Memorization and generalization under extreme overparameterization. In *ICLR*.

[44] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *ICLR*.

[45] Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. 2020. Consistency regularization for generative adversarial networks. In *ICLR*.

[46] Bohan Zhuang, Lingqiao Liu, Yao Li, Chunhua Shen, and Ian Reid. 2017. Attend in groups: A weakly-supervised deep learning framework for learning from web data. In *CVPR*. 1878–1887.
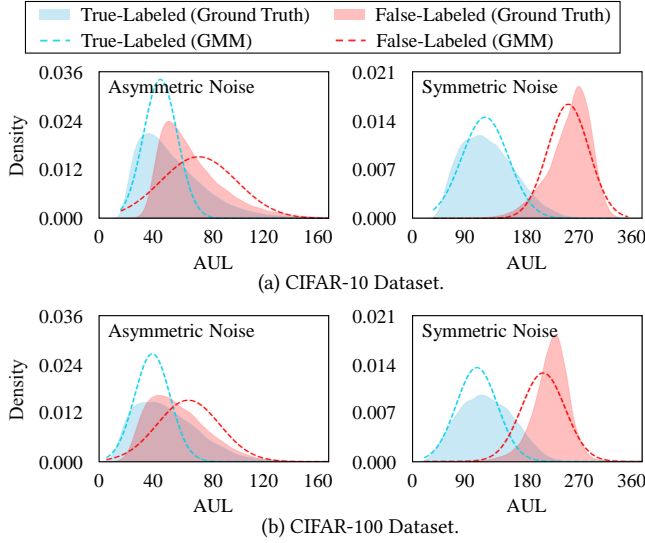
Figure 10: AUL distributions of true- and false-labeled samples using the ground-truth label and the GMM on two CI-FAR datasets with two synthetic noises of 40%.

A NOISE RATE ESTIMATION

In order to accurately estimate the noise rate, we compared *two* widely-used methods. The first one was adopted in all experiments because it showed better performance than the second one.

1. **Gaussian Mixture Model (GMM):** The first method is exploiting a one-dimensional and two-component GMM to model the loss distribution of true- and false-labeled samples [1, 27]. Because the loss distribution tends to be bimodal, the probability of a sample being a false-labeled sample is obtained through its posterior probability. Subsequently, the noise rate is estimated by computing the expectation of the posterior probability for all the training samples. However, in considering that the DNN eventually memorizes all the training samples, the training loss becomes less separable by the GMM as the training progresses. Thus, we computed the *Area Under the Loss curve* (AUL) [27], which is the sum of the samples' training losses obtained from all previous training epochs. The main benefit of the AUL is that its distribution remains separable even after the loss signal decays in later epochs. Therefore, as shown in Figure 10, the loss distributions of true- and false-labeled samples are modeled by fitting the GMM to the AULs of all the training samples, and the noise rate at time t is estimated by

$$\hat{\tau} = \mathbb{E}_{(x,\tilde{y}) \in \tilde{\mathcal{D}}} \left[p\big(g | AUL_t(x, \tilde{y})\big) \right],$$

$$\text{where } AUL_t(x, \tilde{y}) = \sum_{i=1}^{t} f_{(x,\tilde{y})}(\Theta_t) \text{ and} \quad (14)$$

g is the Gaussian component with a larger mean (i.e., larger AUL).

2. **Cross-Validation:** The second method is estimating the noise rate by applying cross-validation on two randomly divided noisy training datasets $\tilde{\mathcal{D}}_1$ and $\tilde{\mathcal{D}}_2$ [5]. Under the assumptions that these two datasets share exactly the same label transition matrix, the noise rate quantifies the test accuracy of DNNs, which are
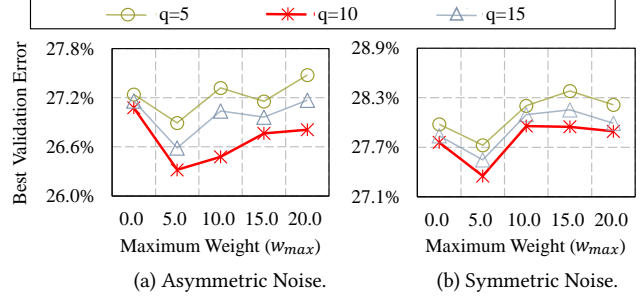

(a) Asymmetric Noise.   (b) Symmetric Noise.

Figure 11: Hyperparameter selection on the CIFAR-100 dataset with two noise types of 40%.

Table 4: Best test errors (%) of MORPH compared with DM-DYR-SH and DivdeMix using CIFAR-100.

Noise Type	Sym. Noise			Asym. Noise	
Noise Rate	20%	40%	70%	20%	40%
DM-DYR-SH	28.0	32.2	46.4	31.1	46.3
DivdeMix	25.3	27.6	**38.9**	25.4	44.1
MORPH	**23.6**	**27.4**	39.5	**23.0**	**26.3**

trained and tested on previously mentioned noisy datasets $\tilde{\mathcal{D}}_1$ and $\tilde{\mathcal{D}}_2$, respectively. In the case of synthetic noises, the test accuracy is approximated by a quadratic function of the noise rate. Therefore, the noise rate can be estimated from the test accuracy obtained by the cross-validation. Refer to [5] for details.

B SUPPLEMENTARY EVALUATION

B.1 Hyperparameter Selection

To ascertain the optimal values of the history length q and the maximum weight w_{max}, we trained a WideResNet-16-8 on CIFAR-100 at a noise rate of 40%. Here, because no validation data exists for the CIFAR-100 dataset, we constructed a small clean validation set by randomly selecting 1,000 images from the original training set of 50,000 images. Then, the noise injection process was applied to only the rest 49,000 training images. Figure 11 shows the validation errors of MORPH obtained by grid search on the noisy CIFAR-100 dataset. These two hyperparameters were chosen in the grid $q \in \{5, 10, 15\}$ and $w_{max} \in \{0.0, 5.0, 10.0, 15.0\}$. The two hyperparameters were not that sensitive to the noise rate. Typically, the lowest validation error depending on q was achieved when the value of q was 10 for both noise types. As for w_{max}, the validation error was observed to be the lowest when the value of w_{max} was 5.0. Therefore, we set the values of q and w_{max} to be 10 and 5.0, respectively, in all experiments.

B.2 Comparison with More Recent Methods

DM-DYR-SH [1] and *DivideMix* [18] are similar to MORPH in that they also use mixture models for handling noisy labels. However, the mixture model is used for a *different* purpose: *DM-DYR-SH* and *DivideMix* use the mixture model to identify clean samples based on the small loss trick; on the other hand, MORPH uses the model to find the best transition point. In addition, MORPH no longer relies on the small-loss trick, and incrementally identifies a better set of clean samples through our alternating scheme in Phase II.

Table 5: Best test errors (%) of seven training methods on two synthetic noises of 40% **in Figures 3 and 4.**

Noise Type	Asymmetric Noise in Figure 3			Symmetric Noise in Figure 4		
Dataset	CIFAR-10	CIFAR-100	Tiny-Image	CIFAR-10	CIFAR-100	Tiny-Image
Co-teaching	24.7	44.8	61.8	12.4	30.2	48.8
Co-teaching+	21.3	53.3	66.5	20.3	53.7	64.2
INCV	22.2	47.0	63.8	9.43	30.1	49.7
ITLM	12.9	41.9	63.9	8.43	29.3	49.0
SELFIE	13.5	38.6	55.4	10.4	28.2	48.4
MORPH w/o. $\mathcal{J}(\Theta)$	7.25	27.1	48.9	8.27	27.8	47.0
MORPH w. $\mathcal{J}(\Theta)$	**6.34**	**26.3**	**47.2**	**8.01**	**27.4**	**46.2**

Table 6: Training time (sec) of seven training methods on two synthetic noises of 40% **in Figures 3 and 4.**

	Asymmetric Noise in Figure 3			Symmetric Noise in Figure 4		
CIFAR-10	CIFAR-100	Tiny-Image	CIFAR-10	CIFAR-100	Tiny-Image
14,107	13,386	60,262	13,171	11,886	53,660
15,527	17,244	82,982	15,365	17,459	80,038
30,376	26,628	104,597	28,915	24,668	99,314
20,068	18,212	69,037	18,763	16,749	60,469
29,730	28,352	115,208	29,436	27,931	113,380
9,646	**9,477**	**39,728**	**9,598**	**9,488**	**40,154**
11,198	11,828	50,594	11,331	12,049	49,412

Nevertheless, we compared MORPH with *DM-DYR-SH* and *DivideMix* for a more in-depth analysis. The results are summarized in Table 4. Only MORPH achieved the noise type robustness; the performance of *DM-DYR-SH* and *DivideMix* was considerably worse in asymmetric noise because their underlying philosophy is based on the *small-loss* trick. In fact, *DM-DYR-SH* and *DivideMix* are favored because they are equipped with an unsupervised or semi-supervised method such as *Beta Mixture* [24] and *MixMatch* [3] to further improve the performance, while MORPH deals with only sample selection. Please note that this superior performance of MORPH was achieved even without adopting the unsupervised or semi-supervised method. MORPH is expected to perform even much better when combined with such an additional technique.

C DETAILS ON EXPERIMENT SETTING

C.1 Benchmark Datasets

For the experiment, we prepared *five* synthetic or real-world noisy datasets: CIFAR-10[3] (10 classes) [14] and CIFAR-100[3] (100 classes) [14], a subset of 80 million categorical images with 50K training images and 10K test images; Tiny-ImageNet[4] (200 classes) [15], a subset of ImageNet with 100K training images and 10K test images; Webvision 1.0[5] (1, 000 classes) [21], real-world noisy data of crawled images using the concepts in ImageNet with 2.4M training images, 50K WebVision validation images, and 50K ImageNet ILSVRC12 validation images; FOOD-101N[6] (101 classes) [17], real-world noisy data of crawled food images with 310K training images and 25K FOOD-101 validation images. Following the previous work [5], we used only the first 50 classes of the Google image subset in Webvision 1.0.

C.2 Training Setup for Real-World Data

To verify the practical usability of MORPH on real-world noisy labels, we performed a classification task on Webvision 1.0 and FOOD-101N. We followed exactly the same configurations in the previous work [5, 17, 20]. For Webvision 1.0, we trained an InceptionResNet-V2 from scratch for 120 epochs using SGD with a momentum of 0.9 and an initial learning rate of 0.1, which was divided by 10 after 40 and 80 epochs (refer to [5]). For FOOD-101N, we trained a ResNet-50 with the ImageNet pretrained weights for 60 epochs using SGD with a momentum of 0.9 and an initial learning rate of 0.01, which was divided by 10 after 30 epochs (refer to [17]). Regardless of the dataset, we used a batch size of 64, a dropout of 0.4, and a weight

decay of 0.001. Random crops and horizontal flips were applied for data augmentation.

C.3 Algorithm Hyperparameters

For reproducibility, we clarify the hyperparameter setup of all compared algorithms. Please note, if the true noise rate τ was needed as one of the hyperparameters (in *Co-teaching*, *Co-teaching+*, *ITLM*, and *SELFIE*), then it was set to be the noise rate $\hat{\tau}$ estimated by MORPH for fair comparison. The other hyperparameters for each method are configured favorably as follows:

- *Co-teaching(+)*: To decrease the number of selected samples gradually at the beginning of the training, the warm-up epoch is required as a hyperparameter; it was set to be 15, which is reported to achieve the best performance in the original paper [9].
- *INCV*: Following the original paper [5], the total number of training rounds was set to be 4; the DNN was trained for 50 epochs using the Adam optimizer; an initial learning rate was set to be 0.001, which was divided by 2 after 20 and 30 epochs and finally fixed to be 0.0001 after 40 epochs. Subsequently, all the training samples selected by *INCV* were used to retrain the DNN using *Co-teaching* with the same configuration.
- *ITLM*: Because it iterates the training process multiple times as well, the total number of training rounds was set to be 5. As mentioned in the original paper [31], the training process for the first 4 rounds was early stopped because it may help filter out false-labeled samples. Subsequently, without early stopping, the DNN was retrained using the samples selected from the 4th round during the last training round.
- *SELFIE*: Four hyperparameters are required for *SELFIE*. The warm-up epoch for sample selection was set to be 15 similar to *Co-teaching*. The uncertainty threshold and the history length for loss correction were set to be 0.05 and 15, respectively, which are the best values obtained from a grid search in the original paper [33]. The training process was restarted twice according to the authors' recommendation.

D COMPLETE EXPERIMENT RESULTS

D.1 Test Error with Synthetic Noises

Table 5 shows the *test error* of seven training methods using WideResNet-16-8 on *simulated* noisy datasets with 40% noise rate. Two variants of MORPH depending on the existence of the consistency regularization were included for comparison.

D.2 Training Time with Synthetic Noises

Table 6 shows the *training time* of seven training methods using WideResNet-16-8 on *simulated* noisy datasets with 40% noise rate.

---

[3]https://www.cs.toronto.edu/~kriz/cifar.html
[4]https://www.kaggle.com/c/tiny-imagenet
[5]https://data.vision.ee.ethz.ch/cvl/webvision/dataset2017.html
[6]https://kuanghuei.github.io/Food-101N