DyGen: Learning from Noisy Labels via Dynamics-Enhanced Generative Modeling

Yuchen Zhuang
yczhuang@gatech.edu
Georgia Institute of Technology
Atlanta, GA, USA

Yue Yu
yueyu@gatech.edu
Georgia Institute of Technology
Atlanta, GA, USA

Lingkai Kong
lkkong@gatech.edu
Georgia Institute of Technology
Atlanta, GA, USA

Xiang Chen
xiangche@adobe.com
Adobe Research
San Jose, CA, USA

Chao Zhang
chaozhang@gatech.edu
Georgia Institute of Technology
Atlanta, GA, USA

ABSTRACT

Learning from noisy labels is a challenge that arises in many real-world applications where training data can contain incorrect or corrupted labels. When fine-tuning language models with noisy labels, models can easily overfit the label noise, leading to decreased performance. Most existing methods for learning from noisy labels use static input features for denoising, but these methods are limited by the information they can provide on true label distributions and can result in biased or incorrect predictions. In this work, we propose the Dynamics-Enhanced Generative Model (DyGen), which uses dynamic patterns in the embedding space during the fine-tuning process of language models to improve noisy label predictions. DyGen uses the variational auto-encoding framework to infer the posterior distributions of true labels from noisy labels and training dynamics. Additionally, a co-regularization mechanism is used to minimize the impact of potentially noisy labels and priors. DyGen demonstrates an average accuracy improvement of 3.10% on two synthetic noise datasets and 1.48% on three real-world noise datasets compared to the previous state-of-the-art. Extensive experiments and analyses show the effectiveness of each component in DyGen. Our code is available for reproducibility on GitHub[1].

CCS CONCEPTS

• **Computing methodologies** → **Supervised learning by classification**.

KEYWORDS

Noisy Label Learning, Training Dynamics, Generative Modeling

ACM Reference Format:
Yuchen Zhuang, Yue Yu, Lingkai Kong, Xiang Chen, and Chao Zhang. 2023. DyGen: Learning from Noisy Labels via Dynamics-Enhanced Generative Modeling. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3580305.3599318

[1]https://github.com/night-chen/DyGen

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08.
https://doi.org/10.1145/3580305.3599318

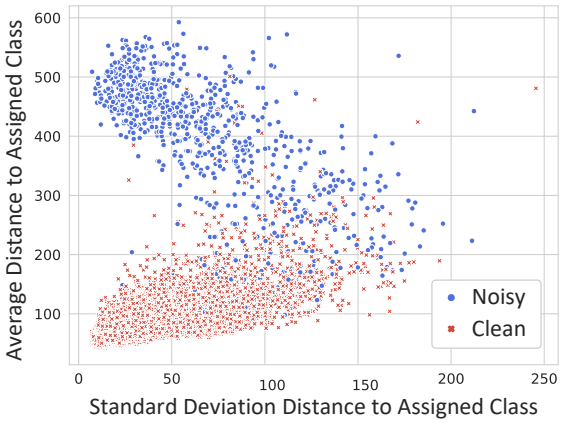


Figure 1: The Euclidean distances between instances and their corresponding label cluster centroids in the embedding space on the 20newsgroup dataset with 20% symmetric noise on labels. The x-axis represents the standard deviation of these distances over epochs during BERT fine-tuning, and the y-axis displays their mean. The patterns of the training dynamics clearly distinguish noisy and clean samples.

1 INTRODUCTION

In many applications, collecting clean labeled data can be much more costly compared to obtaining noisy labeled data. Noisy labels can be cheaply obtained in large quantities from sources such as crowdsourcing [39, 46], web annotations [8, 28], labeling rules [11, 60], and search engines [51, 58]. Using large-scale noisy labeled data holds the potential of training powerful deep learning models with reduced data curation costs. Particularly, fine-tuning pretrained language models (PLMs) with noisy labels have gained interest for a wide range of text analysis tasks [1, 41, 65]. However, the over-parameterized PLMs, due to their large size, are prone to overfitting the label noise, leading to decreased performance [3, 9, 63]. This has become a critical challenge that hinders PLMs from delivering satisfactory results when trained with noisy supervision.

The problem of learning from noisy supervision has been widely studied in the machine learning community. Existing approaches

to this issue can be broadly classified into three categories. 1) *Data Cleaning* methods [2, 6, 25, 34, 49, 52, 55, 65] detect noisy samples using specific criteria such as Area Under Margin [37] and Data Cartography [41] and remove, reweigh, or correct these samples for subsequent model training. 2) *Regularization* methods design regularized loss functions [14, 29, 31, 44, 48, 64] or train multiple models to regularize each other [15, 16, 42, 45, 55, 65], with the goal of improving robustness under label noise. 3) *Noise Transition Estimation* methods [7, 36, 50, 53, 54, 63] estimate the transition matrix $p(\tilde{y}|y, \mathbf{x})$ that maps clean labels y to noisy labels $\tilde{y}$, conditioned on input features $\mathbf{x}$. Noisy Prediction Calibration [5] is a recent approach that models the transition from noisy predictions to the true labels $p(y|\hat{y}, \mathbf{x})$. Each of these categories has its own advantages and drawbacks, and their performance depends on the specific nature of the noise and the input features being used.

A major challenge in existing methods for learning from noisy supervision is their dependence on either the original input features or the embeddings learned with noisy labels. Both scenarios pose limitations when fine-tuning PLMs with noisy labels. First, the original input features $\mathbf{x}$ from PLMs have limited expressivity and therefore cannot effectively distinguish between noisy and clean labels [24]. This can hurt the efficiency of data cleaning methods and the models that learn the noise-to-truth transitions based on $\mathbf{x}$. Furthermore, the input features may hold some information about the true labels y, however, they only encompass a limited understanding of the relationship between the true labels y and the noisy labels $\tilde{y}$. This limitation leads to a reduced capability for generalization to all types of noise. Second, fine-tuning PLMs with noisy labels can also hinder the effectiveness of denoising, as label noise can compromise the quality of the learned embeddings. Over-fitting to the label noise can cause the model to memorize incorrect labels and mistakenly consider some noisy samples as clean ones, even with metrics in regularization methods during fine-tuning. This also impedes noise transition estimation methods from accurately modeling the generation of noise. Consequently, many existing studies are grounded in strong assumptions or are hindered by imprecise noise estimation, resulting in inconsistent performance across varying types of label noise [40].

In this work, we have discovered that noisy and clean samples exhibit distinct behaviors in the embedding space during PLM fine-tuning with noisy labels. During the early stages of fine-tuning, we found that the noisy samples tend to be closer to the cluster associated with the true label y. However, as training progresses, these noisy samples are gradually drawn towards the cluster associated with the assigned noisy label $\tilde{y}$. Therefore, the noisy samples tend to have relatively larger distances to their assigned label clusters due to this training dynamics pattern. Such dynamic patterns can be quantified by the Euclidean distance between each sample and its assigned cluster center at each training epoch. In Figure 1, we visualize the computed distance in the embedding space with the mean (y-axis) and standard deviation (x-axis) over epochs. This plot clearly illustrates that noisy samples tend to have larger means and standard deviations as they move from the true label cluster to the noisy label cluster during training.

We thus propose a dynamics-enhanced generative model DyGen for denoised fine-tuning of PLMs. Our model is based on the observation that noisy and clean samples have different dynamics in the embedding space during the fine-tuning process. To take advantage of this dynamic pattern, our model treats the true labels as latent variables and infers them from the dynamic patterns and the noisy labels. Our model differs from previous generative denoising models [50, 53, 54] in its use of features and modeling of how the features and noisy labels are generated. Unlike these previous models, which generate both the noisy label $\hat{y}$ and the input feature $\mathbf{x}$ conditioned on the true label y ($p(\mathbf{x}, \tilde{y}|y)$), our model leverages dynamic training patterns w and treats the true label y as the latent encoding of $\hat{y}$. This makes it easier to learn, as it only requires generating the noisy label $\hat{y}$, and allows for inference of the posterior $p(y|\hat{y}, \mathbf{w})$ using the variational auto-encoding framework. Furthermore, we can use the discriminative power of the dynamic patterns to induce the prior distribution $p(y|\mathbf{w})$ of our generative model. To improve robustness in inferring the true label, we also employ a co-regularization loss that encourages multiple branches of our generative model to reach a consensus for the posterior $p(y|\hat{y}, \mathbf{w})$.

We have conducted thorough experiments on two datasets with various synthetic noise types and three datasets from the WRENCH benchmark [60] with real-world weak label noise. Our method DyGen consistently surpasses the state-of-the-art baselines, with an average improvement of 2.13% across various noise types and ratios on both synthetic and real-world datasets. Furthermore, DyGen demonstrates remarkable robustness even under extreme label noise ratios, as high as 50%. Additionally, DyGen enhances model calibration by generating predicted probabilities that are more accurately aligned with the true label distribution due to its dynamics-based probabilistic denoising approach. Our contributions are as follows:

- We have discovered that dynamic training patterns in the hidden embedding space during PLM fine-tuning can effectively distinguish between clean and noisy samples. Utilizing this insight, we have devised a denoised fine-tuning approach for PLMs. To our knowledge, this is the first time that dynamic training patterns are used to achieve robust fine-tuning of PLMs with noisy labels.
- We design a generative model that models the reconstruction of the noisy label $\hat{y}$ from the latent true label y and the training dynamics $\mathbf{w}$. We induce a prior distribution for the latent variable y based on the dynamics $\mathbf{w}$ and present a training procedure based on variational auto-encoding.
- To enhance robustness, we employ multiple branches that co-regularize each other to reach consensus for the posterior $p(y|\hat{y}, \mathbf{w})$.
- We have conducted a comprehensive analysis of the noisy learning problems in text data, covering both synthetic and real-world noise scenarios. Our proposed method, DyGen, consistently outperforms other approaches across different types and levels of noise.

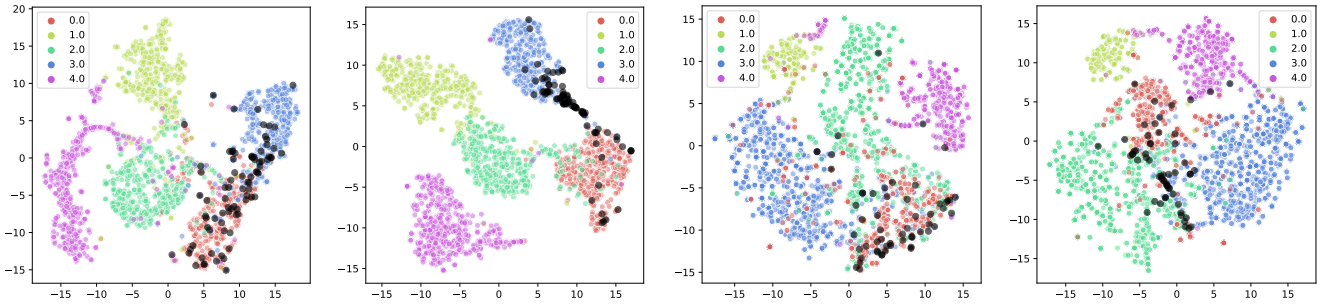
2 RELATED WORK

In this section, we briefly introduce several related lines of works on learning from noisy labeled data.

Noise Transition Matrix Estimation. Most existing techniques in this line model the generation of noise from true to noisy labels as a transition matrix

$$\mathbf{T}_{j,k}(\mathbf{x}) = p(\tilde{y} = j|y = k, \mathbf{x}), \quad \forall j, k = 1, \cdots, c. \tag{1}$$

where c is the total number of classes. If the transition matrix is estimated correctly, classifiers can be trained on noisy data and converge to the optimal solution with theoretical guarantees[26].



(a) 20newsgroup (b) ChemProt

Figure 2: Examples of the observed training dynamic pattern in the corrupted dataset, 20newsgroup, and a real-world noisy dataset, ChemProt [21]. We select 5 categories of the dataset and the black points are the corrupted samples from the true labels (red) to the noisy assigned labels (blue). The left and right figures in each group are t-SNE [43] visualizations on training embeddings obtained from the 2-nd and 10-th epoch, respectively.

However, the noise transition matrix is difficult to estimate. To improve its modeling, recent works propose various assumptions on the nature of noise. For example, [36, 54, 63] assume the noise is instance-independent, namely $p(\tilde{y}|y, \mathbf{x}) = p(\tilde{y}|y)$. This assumption is often unrealistic for real-world noises, where labeling errors can depend on the input features $\mathbf{x}$. Xia et al. [50] assume that the noise generation is dependent on different parts of an instance; Yao et al. [53] introduce an auxiliary latent variable $\mathbf{z}$ that works with true label y together to generate the instance feature $\mathbf{x}$. Nevertheless, these assumptions are too specific and cannot be readily applied to real scenarios where the noise patterns can be diverse and complicated. Thus, Bae et al. [5] consider the true label y as the latent variable, and infer the posterior:

$$\mathbf{H}_{j,k}(\mathbf{x}) = p(y = j|\hat{y} = k, \mathbf{x}), \quad \forall j, k = 1, \cdots, c. \tag{2}$$

Our approach adopts the same formulation but improves the generative modeling with training dynamic patterns. These patterns enhance the latent variable modeling and provide more accurate prior and posterior distributions for noisy-to-true label transitions.

Regularization from Multiple Models. Deep neural networks are often sensitive to the stochasticity (e.g., weight random initialization and data orders) involved during training. This issue is especially exacerbated for noisy label learning, as noisy examples may further disturb model training. To alleviate this, several works proposed adaptive training strategies that involve *multiple model branches* to improve robustness over noisy labels. Jiang et al. [16] propose two networks, MentorNet and StudentNet, where the MentorNet adopts a reweighting scheme to favor samples that are more likely to be correct to guide the training of the StudentNet. The Decoupling strategy [32] simultaneously trains two networks and updates parameters only when their predictions disagree. Co-teaching [55] also involves two networks, where one network learns from the other network's most confident samples. However, the aforementioned methods only select a part of training examples for training without explicit denoising, and also neglect the information from mislabeled data. JoCoR [45] addresses this problem by incorporating consistency between different models during training, instead of blindly trusting the noisy labels. DyGen takes advantage of existing approaches by establishing an agreement objective among multiple generative branches with identical structures but different initializations. By regularizing these branches towards this consensus, we can mitigate the negative effects of noisy labels and potentially imperfect prior knowledge.

Training Dynamics for Data Cleaning. Training dynamics depict the behaviors of the model predictions over instances as training progresses. The main idea of using training dynamics for noisy learning is to consider the patterns as criteria to detect and correct noisy labeled instances. Along this line, the most straightforward way is to identify samples with lower training loss as the clean subset [2, 25, 55, 56], but this approach is often too simple and rigid, which end up choosing easy-to-learn examples only. To better harness the training dynamics from intermediate iterations, Pleiss et al. [38] introduce a new metric that measures the average gap between the logits of a sample's assigned class and its highest non-designated class, where a negative margin suggests the potential label noise; Swayamdipta et al. [41] hypothesize that noisy samples have smaller probabilities in the assigned category during the whole training process. All these existing metrics for data cleaning are based on heuristics and strong assumptions. In addition, they are easy to make biased judgments as they depend solely on noisy classifiers' posterior information.

3 PROBLEM DEFINITION

We study fine-tuning PLMs with noisy labels for classification, formulated as follows: given a noisy training dataset $\mathcal{D}_{\text{train}} = (\mathbf{x}_i, \tilde{y}_i)_{i=1}^n$ consisting of potentially corrupted labels $\tilde{y}$, the ultimate goal is to minimize the true risk $R_L(f) := \mathbb{E}[L(f(x;\theta), y)]$ between the model predictions $f(\mathbf{x};\theta)$ and the underlying true labels y, where $L(\cdot)$ is a loss function. Since the true labels y are not accessible, the only available risk function is the noisy empirical risk $\tilde{R}_L^{emp}(f) := \frac{1}{n}\sum_{i=1}^n L(f(\mathbf{x}_i;\theta), \tilde{y}_i)$ based on noisy labels $\tilde{y}$. Thus, the objective during PLM fine-tuning with noisy labels becomes finding a function that minimizes the true risk $R_L(f)$ through the learning process with the noisy empirical risk $\tilde{R}_L^{emp}(f)$.

4 TRAINING DYNAMICS

4.1 Training Dynamics in Embedding Space

We conducted a comprehensive study through over 1,500 experimental trials of fine-tuning various PLMs, including BERT [10],

BioBERT [23], PubMedBERT [12], and RoBERTa [30]. We used noisy labeled datasets for different NLP benchmarks such as 20newsgroup [22], AG News [27, 62], ChemProt [21], TREC [4], and SemEval [66], with both synthetic and real-world noise at various ratios. In our experiments, we modeled the PLM as a two-module model, $f_\theta = g_{\theta_{-1}} \circ h_{\theta_{:-1}}$, where $h_{\theta_{:-1}}$ is the PLM-based encoder and $g_{\theta_{-1}}$ is the final classifier stacked on the encoder. We optimized the parameters θ using gradient descent algorithms to minimize the empirical risk $\tilde{R}_L^{emp}(f)$ over E training epochs with noisy labeled data. During PLM fine-tuning, we observe that the following dynamic patterns consistently occur in the embedding space:

When fine-tuning PLMs with noisy labels, noisy samples gradually shift away from the true-label cluster towards their assigned-label cluster in the embedding space, leading to a larger Euclidean distance between the noisy samples and their assigned label cluster centroids across epochs. Clean samples, on the other hand, exhibit smaller mean and deviation of distances, resulting in a dynamic contrast in their training patterns compared to the noisy samples.

Figure 2 visualizes the dynamic patterns of noisy samples using t-SNE [43] on two example settings: 20newsgroup [22] with 20% synthetic symmetric noise and ChemProt [21] with 22.88% real noise from weak labeling rules. It shows the embeddings of instances at early and late stages of fine-tuning a BERT-base model. Clean samples are represented by colored points, with colors denoting their true labels, and noisy samples are represented by black points, which have moved from their true-label cluster (red) to their assigned-label cluster (blue) during fine-tuning with noisy labels.

The pattern observed is likely due to the *memorization effect* [3, 48] of deep neural networks, which tend to fit clean label patterns first and then overfit noise. When fine-tuning PLMs with noisy labeled data, all samples tend to remain closer to their true label clusters in the early stages, as PLMs encode semantic knowledge [57, 67] in their embeddings. However, as training continues, the model begins to learn correlations between features and assigned labels, causing noisy samples to gradually move from true-label clusters to assigned-label clusters and overfitting to noise in later epochs. This fitting dynamic still occurs even with large noise ratios, as the randomness of the noise is unlikely to overpower the collective signal of the clean data. Our hypothesis is that this training dynamic will persist as long as there is no systematic bias that dominates the clean data signal.

4.2 Quantitative Measurements of Pattern

To quantify the pattern observed, we measure the Euclidean distances between instance hidden embeddings and the centroids of their assigned label clusters. We fine-tuned the PLM model for E epochs using noisy labeled data and represent the training dynamics of instance i using statistics in the embedding spaces over E epochs. To do this, we first compute the cluster centroids $c_k^{(e)}$ of each class k at each epoch e:

$$c_k^{(e)} = \frac{1}{n_{\text{train}}} \sum_{i=1}^{n_{\text{train}}} h_{\theta_{:-1}^{(e)}}(\mathbf{x}_i) \cdot \mathbb{1}(\tilde{y}_i = k), \tag{3}$$

where $\theta_{:-1}^{(e)}$ denotes the parameters of the feature encoder h at the e-th epoch. Then, we compute the average embedding distance between the samples and the assigned label cluster centroids:

$$\mu_i = \frac{1}{E} \sum_{e=1}^{E} \| h_{\theta_{:-1}^{(e)}}(\mathbf{x}_i), c_{\tilde{y}_i}^{(e)} \|_2. \tag{4}$$

In addition, we compute the standard deviation of distances over E epochs to measure the magnitude of distance variations:

$$\sigma_i = \sqrt{\frac{1}{E} \sum_{e=1}^{E} (\| h_{\theta_{:-1}^{(e)}}(\mathbf{x}_i), c_{\tilde{y}_i}^{(e)} \|_2 - \mu_i)^2}. \tag{5}$$

The differences between noisy and clean data are clearly illustrated in Figure 1. The noisy samples exhibit larger mean and standard deviations in their distances to assigned label clusters compared to clean samples.

5 METHOD

For PLM fine-tuning with noisy labels, the ultimate goal is to learn a model that produces the distribution over the true label y for any input $\mathbf{x}$, namely $p(y|\mathbf{x})$. As we have only noisy labeled data during training, we decompose this objective as:

$$p(y|\mathbf{x}) = \sum_{\hat{y}} p(\hat{y}|\mathbf{x}) p(y|\hat{y}, \mathbf{x}), \tag{6}$$

where $\hat{y}$ is the observed noisy label for instance $\mathbf{x}$.

In the above equation, the $p(\hat{y}|\mathbf{x})$ is the biased model learned with the noisy labeled data $\mathcal{D}_{\text{train}}$ using standard fine-tuning. The challenge is to infer the true labels' posterior distribution, $p(y|\hat{y}, \mathbf{x})$, which serves as a calibration term that debiases $p(\hat{y}|\mathbf{x})$. According to the observation in § 4, we propose to use the training dynamics $\mathbf{w}$ in lieu of $\mathbf{x}$, as $\mathbf{w}$ contains rich information about both noisy predictions $\hat{y}$ and clean labels y. Based on this insight, the objective is reformulated as:

$$p(y|\mathbf{x}) \propto \sum_{\hat{y}} p(\hat{y}|\mathbf{x}) p(y|\hat{y}, \mathbf{w}). \tag{7}$$

To model the two distributions $p(\hat{y}|\mathbf{x})$ and $p(y|\hat{y}, \mathbf{w})$ in Eq. 7, we propose a two-stage learning process (also see Figure 3): (1) *Stage I*: Learn the standard noisy-supervised model to estimate $p(\hat{y}|\mathbf{x})$ and encode the training trajectories $\mathbf{w}$ as compositions of hidden embeddings obtained from each epoch during fine-tuning; (2) *Stage II*: Learn the deep generative model to estimate the transition from noisy predictions to true labels and model the function $p(y|\hat{y}, \mathbf{w})$. The rest of this section describes: (a) Training trajectory-based deep generative model in § 5.1, (b) Co-regularization mechanism in § 5.2, and (c) Training objective in § 5.3.

5.1 Deep Generative Model on Training Patterns

5.1.1 Probabilistic Model Structure. To model distribution $p(y|\hat{y}, \mathbf{w})$, we introduce a deep generative model that can encode and reconstruct noisy labels $\hat{y}$, conditioned on the training trajectories $\mathbf{w}$. Specifically, we consider the true labels y as the latent variables and define the following generative process: first, y is drawn conditioned on the training trajectories $\mathbf{w}$; then, $\hat{y}$ is generated based on y and $\mathbf{w}$. By following this generative process, we can factorize the joint distribution, modeling the relationship between observations $\hat{y}$ and latent variables y:

$$p(y, \hat{y}|\mathbf{w}) = p(y|\mathbf{w}) p(\hat{y}|y, \mathbf{w}). \tag{8}$$

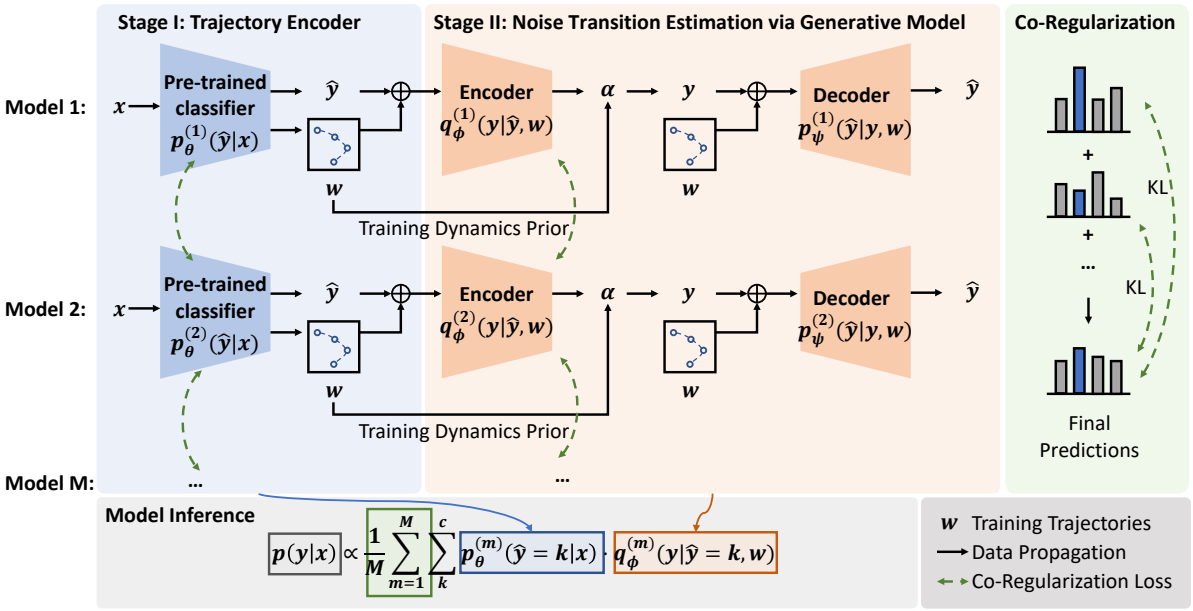


Figure 3: The DyGen framework, containing (1) the noisy-supervised model for training trajectory pattern encoding; (2) the generative process, considering true label y as a latent variable and reconstructing $\hat{y}$; (3) co-regularization loss between multiple branches of models; (4) the inference process to predict true labels from noisy predictions.

Since true labels y are typically assumed to be categorical, we treat y as random probability vectors sampled from a Dirichlet distribution:

$$y \sim \text{Dirichlet}(\alpha_{\mathbf{w}}),\ \hat{y} \sim \text{Multi}(\pi_{\mathbf{w},y}), \tag{9}$$

where $\alpha_{\mathbf{w}} \in \mathbb{R}_+^c$ represents the instance-dependent parameters of the prior probability distribution for all c categories, given a training trajectory $\mathbf{w}$; $\text{Dirichlet}(\alpha_{\mathbf{w}})$ is a Dirichlet distribution parameterized by $\alpha_{\mathbf{w}}$, which is also a conjugate prior of the corresponding multinomial distribution; and $\pi_{\mathbf{w},y}$ is the probability of selecting a class for the noisy label.

5.1.2 Dynamics-Based Prior. Since the prior function $p(y|\mathbf{w})$ in Eq. 8 is unknown from the training stage, we approximate it as $p_\theta(y|\mathbf{w})$ using the observed training dynamics patterns (derived in § 4), where θ is the trajectory encoder parameter in Stage I. First, to effectively distinguish between noisy and clean samples, we sum up the mean and standard deviation computed from Eq.4 and Eq.5 as a scoring function: $s_i = \mu_i + \sigma_i$. Second, we assume that the top β percent of instances with the highest s_i are potentially noisy, denoted as $\hat{\mathcal{D}}_{\text{train}}^{\text{noisy}}$, where β is the estimated error rate. The remaining instances with lower s_i can be considered clean, denoted as $\hat{\mathcal{D}}_{\text{train}}^{\text{clean}}$. We use K Nearest Neighbor (KNN) algorithm on $\hat{\mathcal{D}}_{\text{train}}^{\text{noisy}}$, with $\hat{\mathcal{D}}_{\text{train}}^{\text{clean}}$ as the reference set to sample K neighbors from. Third, we combine the most selected labels $\bar{y}$ from neighbors for $\hat{\mathcal{D}}_{\text{train}}^{\text{noisy}}$ and the remaining assigned labels $\tilde{y}$ for $\hat{\mathcal{D}}_{\text{train}}^{\text{clean}}$ to update $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, \tilde{y}_i, y_i^{\text{prior}})\}_{i=1}^{n_{\text{train}}}$, where y_i^{prior} indicates the prior knowledge of possible true labels.

With the prior knowledge y_i^{prior}, we can define the Dirichlet distribution parameter α_w as:

$$\alpha_w^k = \begin{cases} \delta, & k \neq y_i^{\text{prior}} \\ \delta + \rho, & k = y_i^{\text{prior}} \end{cases},\ k = 1, 2, \cdots, c. \tag{10}$$

where δ and ρ are hyper-parameters to setup α for Dirichlet distribution. The prior function $p_\theta(y|\mathbf{w})$ can then be defined as:

$$p_\theta(y|\mathbf{w}) = \text{Dirichlet}(\alpha_w). \tag{11}$$

Algorithm 1 outlines the computation of dynamics-based priors.

5.1.3 Deep Generative Model Architecture. As our main estimation target $p(y|\hat{y}, \mathbf{w})$ in Eq. 7 is intractable, we apply variational inference to approximate the desired posterior distribution. To this end, we first introduce a variational distribution $q_\phi(y|\hat{y}, \mathbf{x})$. We then minimize the Kullback-Leibler (KL) divergence between the true posterior $p(y|\hat{y}, \mathbf{x})$ and the variational distribution $q_\phi(y|\hat{y}, \mathbf{x})$. Specifically, we adopt the structure of Variational Autoencoder (VAE) to parameterize the encoder $q_\phi(y|\hat{y}, \mathbf{w})$ and decoder $p_\psi(\hat{y}|y, \mathbf{w})$. Thus, we can write the KL divergence function:

$$\begin{aligned} &\text{KL}(q_\phi(y|\hat{y}, \mathbf{w}) \| p(y|\hat{y}, \mathbf{w})) = \int q_\phi(y|\hat{y}, \mathbf{w}) \log \frac{q_\phi(y|\hat{y}, \mathbf{w})}{p(y|\hat{y}, \mathbf{w})} dy \\ &= \log p(\hat{y}|\mathbf{w}) - \mathbb{E}_{y \sim q_\phi(y|\hat{y}, \mathbf{w})}[\log p_\psi(\hat{y}|y, \mathbf{w})] \\ &\quad + \text{KL}(q_\phi(y|\hat{y}, \mathbf{w}) \| p(y|\mathbf{w})) \\ &= \log p(\hat{y}|\mathbf{w}) - \text{ELBO}. \end{aligned} \tag{12}$$

Different from the normal distribution prior in traditional VAE, we apply the reparameterization trick for Dirichlet distribution from

Algorithm 1: Computation of Dynamics-Based Prior.

Input: $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^{n_{\text{train}}}$: noisy training data ; f_θ: noisy supervised model; E: Number of f_θ training epochs.

// Step 0: *Initialization*

Prepare training trajectory set $\mathcal{S}_{\text{dist}} = \emptyset$

for $e = 1, 2, \cdots, E$ **do**

  // Step 1: *Gather Information for Training Trajectories Encoding.*

  Compute the centroid point of each category on embedding space $c_k^{(e)}$ via Eq. 3.

  Compute Euclidean Distances between Samples $\mathbf{x}$ and Assigned Class $c_{\hat{y}}^{(e)}$: $\mathcal{S}_{\text{dist}} \leftarrow \{\|h_{\theta_{:-1}^{(e)}}(\mathbf{x}), c_{\hat{y}}^{(e)}\|_2\}$.

// Step 2: *Compute Scoring Function as Quantitative Pattern.*

Compute the statistics via Eq. 4 and Eq. 5 on $\mathcal{S}_{\text{dist}}$.

Compute the scoring function $s_i = \mu_i + \sigma_i,\ 0 \le i < n_{\text{train}}$.

Separate the $\mathcal{D}_{\text{train}}$ into possibly-clean set $\hat{\mathcal{D}}_{\text{train}}^{\text{clean}}$ (smaller s_i) and possibly-noisy set $\hat{\mathcal{D}}_{\text{train}}^{\text{noisy}}$ (larger s_i).

// Step 3: *Generate True Label Prior Information.*

Apply KNN with $\hat{\mathcal{D}}_{\text{train}}^{\text{clean}}$ as reference set to correct labels in $\hat{\mathcal{D}}_{\text{train}}^{\text{noisy}}$ to obtain prior knowledge $\{y_i^{\text{prior}}\}_{i=1}^{n_{\text{train}}}$.

Compute the parameters to the prior distribution via Eq. 10

Compute the approximated prior distribution $p_\theta(y|\mathbf{w})$ via Eq. 11.

Output: The approximated prior distribution $p_\theta(y|\mathbf{w})$.

Dirichlet VAE [17]:

$$\begin{aligned}\text{ELBO} = \sum_{k=1}^{c}(&\underbrace{\hat{y}^k \log \hat{y^*}^k + (1-\hat{y}^k)\log(1-\hat{y^*}^k)}_{\text{Reconstruction Loss}} \\ &\underbrace{-\log\Gamma(\alpha_{\mathbf{x}}^k) + \log\Gamma(\hat{\alpha}_{\mathbf{x},\hat{y}}) - (\hat{\alpha}_{\mathbf{x},\hat{y}}^k - \alpha_{\mathbf{x}}^k)\psi(\hat{\alpha}_{\mathbf{x},\hat{y}}^k)}_{\text{Prior Regularizer}}),\end{aligned} \tag{13}$$

where $\hat{y^*}$ is the reconstructed $\hat{y}$ after decoder $p_\psi(\hat{y}|y, \mathbf{x})$; $\Gamma(\cdot)$ and $\psi(\cdot)$ represent the gamma and digamma function, respectively.

5.1.4 Model Inference. To perform model inference, we compute $p(y|\mathbf{w}, \mathbf{x})$ in Eq. 7. After training the whole architecture with ELBO in Eq. 13, we obtain the posterior distribution from the encoder. Thus, we can directly use the mode of Dirichlet distribution to compute H:

$$H = q_\phi(y|\hat{y}, \mathbf{w}) = \frac{\hat{\alpha}_{w,\hat{y}}^y - 1}{\sum_{y=1}^{c}\hat{\alpha}_{w,\hat{y}}^y - c}, \tag{14}$$

where $\hat{\alpha}_{w,\hat{y}}^y$ is the predicted posterior Dirichlet distribution by VAE. We can then rewrite Eq. 7 for inference with $q_\phi(y|\hat{y}, \mathbf{w})$ in Eq. 14 :

$$p(y|\mathbf{x}) \propto \sum_{k=1}^{c} q_\phi(y|\hat{y} = k, \mathbf{w})p(\hat{y} = k|\mathbf{x}). \tag{15}$$

5.2 Co-Regularization Mechanism

Despite efforts to mitigate the negative impact of noisy samples (§ 5.1), the guidance from labels and prior is still imperfect. Small deviations in $p(\hat{y}|x)$ and $p(y|\hat{y}, x)$ could potentially carry over into later stages and affect the overall $p(y|x)$. To address the limitations of imperfect guidance and prevent error propagation, we incorporate multiple branches with identical structures but differing initializations into our model. We use a co-regularization loss across branches to promote consensus and prevent over-reliance on potentially corrupted labels.

The learning process of the co-regularization mechanism involves the learning of M copies of the first-stage model $p_\theta^{(m)}(\hat{y}|\mathbf{x})$ and second-stage generative models $q_\phi^{(m)}(y|\hat{y}, \mathbf{x})$ and $p_\psi^{(m)}(\hat{y}|y, \mathbf{x})$, where m ranges from 1 to M ($M > 2$). To begin, we input the instances $\mathbf{x}_i$ into different models to obtain corresponding prediction probabilities $p_i^{(m)}$ of each model m. The aggregated probability q_i is then computed by averaging these predictions, $r_i = \frac{1}{M}\sum_{m=1}^{M} p_i^{(m)}$, representing the consensus of the models on the true label prediction. The co-regularization loss is calculated as the KL Divergence between the consensus probability r_i and each of the model predicted probabilities $p_i^{(m)}$:

$$\begin{aligned}\ell_{\text{cr}} &= \frac{1}{MN}\sum_{i=1}^{N}\sum_{m=1}^{M}\text{KL}(r_i\|p_i^{(m)}) \\ &= \frac{1}{MN}\sum_{i=1}^{N}\sum_{m=1}^{M}\sum_{j=1}^{C} r_{ij}\log(\frac{r_{ij}+\epsilon}{p_{ij}^{(m)}+\epsilon}),\end{aligned} \tag{16}$$

where ϵ indicates a small positive number to avoid division by zero. Specifically, for Stage I, we define the consensus probabilities and co-regularization loss as follows:

$$\begin{aligned}r(\hat{y}_i|\mathbf{x}_i) &= \frac{1}{M}\sum_{m=1}^{M} p_\theta^{(m)}(\hat{y}_i|\mathbf{x}_i), \\ \ell_{\text{cr}-1} &= \frac{1}{MN}\sum_{i=1}^{N}\sum_{m=1}^{M}\text{KL}(r(\hat{y}_i|\mathbf{x}_i)\|p_\theta^{(m)}(\hat{y}_i|\mathbf{x}_i)).\end{aligned} \tag{17}$$

Similarly, the consensus probabilities and co-regularization loss for the deep generative model in Stage II can be represented as:

$$\begin{aligned}r(y|\hat{y}_i, \mathbf{w}_i) &= \frac{1}{M}\sum_{m=1}^{M} q_\phi^{(m)}(y_i|\hat{y}_i, \mathbf{w}_i), \\ \ell_{cr-2} &= \frac{1}{MN}\sum_{i=1}^{N}\sum_{m=1}^{M}\text{KL}(r(y_i|\hat{y}_i, \mathbf{w}_i)\|q_\phi^{(m)}(y_i|\hat{y}_i, \mathbf{w}_i)).\end{aligned} \tag{18}$$

5.3 Training Objective

The training objective of DyGen is to optimize a joint loss that combines the task-specific loss and the co-regularization loss. For the first stage of dynamics pattern encoding, the task-specific loss $\ell_{\text{task}-1}$ is computed as the cross-entropy loss for classification:

$$\ell_{\text{task}-1} = \frac{1}{M}\sum_{m=1}^{M}\sum_{k=1}^{c} -\tilde{y}^k \log \hat{y}^{(m)}, \tag{19}$$

where $\hat{y}^{(m)}$ indicates the predicted label from the m-th model. Similarly, task-specific loss $\ell_{\text{task}-2}$ for the second stage is calculated as the average negative ELBO in Eq. 13, across all branches of the model. Consequently, the training objectives of Stage I and II are defined as:

$$\begin{aligned}\ell_1 &= \ell_{task-1} + \lambda_1\ell_{cr-1}, \\ \ell_2 &= \ell_{task-2} + \lambda_2\ell_{cr-2},\end{aligned} \tag{20}$$

where λ_1 and λ_2 are positive hyper-parameters.

To further enhance the training process, we implement a warm-up phase for λ_1 and λ_2. During this phase, λ is temporarily set to 0 to guarantee proper model initialization. Upon completion of the warm-up phase, λ will return to its pre-determined positive value. Finally, to obtain the final model predictions, the outputs from each model branch are averaged.:

$$p(y|\mathbf{x}) \propto \sum_{\hat{y}} \frac{1}{M} p_\theta^{(m)}(\hat{y}|\mathbf{x}) q_\phi^{(m)}(y|\hat{y}, \mathbf{w}). \tag{21}$$

We present the learning procedure of DyGen in Algorithm 2.

Algorithm 2: Training procedure of DyGen.

Input: $\mathcal{D}_{\text{train}} = \{(x_i, \tilde{y}_i)\}_{i=1}^{n_{\text{train}}}$: noisy training data ; θ: model parameter for pattern encoder; E: number of stage I training epochs; ϕ and ψ: model parameters in VAE; T: number of stage II training iterations; γ: warm-up ratio; λ_1 and λ_2: hyper-parameters; M: the number of model branches.

// Step 1: *Encode Training Dynamics Pattern.*

for $e = 1, 2, \cdots, E$ **do**

  for $m = 1, 2, \cdots, M$ **do**

    Compute $p_\theta^{(m)}(\hat{y}|x)$.

  Compute the task-specific loss $\ell_{\text{task}-1}$ via Eq. 19.

  Compute the co-regularization loss $\ell_{\text{cr}-1}$ via Eq. 17.

  if $e < \gamma \times E$ **then**

    $\ell_1 = \ell_{\text{task}-1}$.

  else

    $\ell_1 = \ell_{\text{task}-1} + \lambda_1 \ell_{\text{cr}-1}$.

  Update the stage I model parameters $\theta \leftarrow \nabla \ell_1$.

// Step 2: *Compute Dynamics-Enhanced Prior.*

for $m = 1, 2, \cdots, M$ **do**

  Compute dynamics-enhanced prior for each model branch $y^{\text{prior},(m)}$.

// Step 3: *Generative Model for NPC.*

for $t = 1, 2, \cdots, T$ **do**

  for $m = 1, 2, \cdots, M$ **do**

    Compute $q_\phi^{(m)}(y|\hat{y}, x)$ and $p_\psi^{(m)}(\hat{y}|y, x)$.

  Compute the task-specific loss $\ell_{\text{task}-2}$ via Eq. 13.

  Compute the co-regularization loss $\ell_{\text{cr}-2}$ via Eq. 18.

  if $t < \gamma \times T$ **then**

    $\ell_2 = \ell_{\text{task}-2}$.

  else

    $\ell_2 = \ell_{\text{task}-2} + \lambda_2 \ell_{\text{cr}-2}$.

  Update the Stage II model parameters $\{\phi, \psi\} \leftarrow \nabla \ell_2$

// Step 4: *Model Inference.*

Compute and average the predictions as $p(y|x, w)$ via Eq. 21.

Output: The inferred true label for each instance .

6 EXPERIMENTS

6.1 Experimental Setup

6.1.1 Datasets. To verify the efficacy of DyGen, we first experiment on two synthetic-noise datasets: 20newsgroup [22] and AG NEWS [27, 62]. Three different types of synthetic label noise are generated and injected into the datasets following the setups of existing works on learning from noisy supervision [5, 36, 53, 54]: (1) **Symmetric Noise (SN)** flips labels uniformly to other classes [5, 15, 36, 48]; (2) **Asymmetric Noise (ASN)** flips labels within similar classes [5, 15, 42, 48]; and (3) **Instance-Dependent Noise (IDN)** flips label with a probability proportional to the features of the sample [5, 9, 50, 53]. Furthermore, we conduct experiments on three real-world datasets: ChemProt [21], TREC [4], and SemEval [66]. ChemProt is a chemical-protein interaction dataset with 10 classes; TREC is a question classification dataset with 6 classes; and SemEval is a relation extraction dataset with 9 classes. For these three datasets, we use the pre-defined heuristic rules from prior works [59–61] as weak supervision to generate noisy labels. The noise ratio of ChemProt, TREC, and SemEval are 22.88%, 38.56%, and 16.00%, respectively. See Appendix. A for details.

6.1.2 Baselines. We compare with the most relevant state-of-the-art noisy label learning baselines from different categories: (1) *Basic Performances* without specific design for robust learning with label noise [10]; (2) *Multi-Model Training Strategies*, including **Co-Teaching** [15], **JoCoR** [45], **CR** [65]; (3) *Generative Models for Noisy Matrix Estimation*, including **DualT** [54], **CausalNL** [53], **NPC** [5], and **CR w/ NPC**. See Appendix. D for details.

6.1.3 Evaluation Protocol. All experiments are evaluated using accuracy on a clean test set, and the reported test performance is selected according to the performance on a clean development set. This applies to both DyGen and all baselines. We report the average performance as well as standard deviations using 5 random seeds.

6.1.4 Implementation Details. We implement DyGen using PyTorch [35] and HuggingFace [47]. In the experiments on ChemProt, BioBERT [23] is used as the backbone model for the first stage training dynamics pattern encoder θ, while for the rest datasets, we use BERT [10]. We use the same backbone for all baseline methods. See Appendix. E for more details.

6.2 Main Results

Performance Comparison. Table 1 and Table 2 show the main results for the synthetic and the real-world noisy datasets. From these results, we have the following observations:

(1) DyGen significantly outperforms all the baselines on synthetic datasets with varying noise types and ratios. Additionally, DyGen also shows superiority in performance on real-world noisy datasets. Compared with the strongest baseline, CR w/ NPC, directly concatenating the Co-Regularized classifier with the generative model for noisy prediction calibration, DyGen achieves 3.10% gain on average on synthetic datasets and 1.48% on real-world datasets.

(2) The results show that DyGen has larger performance gains on synthetic datasets compared to real-world datasets. This is because real-world datasets often contain a more intricate mix of noises, making it a bit of a challenge for the model to arrive at accurate estimates. Additionally, the noise ratio on real-world noisy datasets is lower than that of synthetic datasets, resulting in less room for improvement with our proposed method.

(3) Compared to the 20newsgroup dataset, the gains of Noisy Transition Matrix estimation-based methods over other baselines are more pronounced for the AG News dataset. This discrepancy can be attributed to the difference in the number of classes between the two datasets. Specifically, the AG News dataset has only four classes, which is much smaller than 20newsgroup. This makes the

Table 1: Main results on synthetic noise datasets.

Dataset (→)	20newsgroup						AG NEWS					
Method (↓) / Noise (→)	20% **SN**	40% **SN**	20% **ASN**	40% **ASN**	20% **IDN**	40% **IDN**	20% **SN**	40% **SN**	20% **ASN**	40% **ASN**	20% **IDN**	40% **IDN**
Basic Performances												
Base	78.84±0.59	70.81±2.13	74.44±1.09	56.18±2.82	77.33±1.12	69.81±0.16	82.08±1.80	78.17±2.69	81.43±0.92	77.15±2.50	85.59±1.91	75.86±3.09
Multi-Model Training Strategies												
Co-Teaching	77.66±1.06	69.25±3.62	77.51±1.84	67.26±1.94	77.45±0.32	73.76±1.63	82.99±1.15	78.79±2.41	81.96±0.77	78.07±1.68	87.85±0.82	76.52±2.21
JoCoR	80.92±0.64	73.27±4.25	81.01±0.20	69.40±3.39	81.57±0.76	74.19±0.83	83.82±1.08	81.26±1.83	85.88±0.98	77.98±1.04	87.04±1.27	79.93±1.46
CR	81.61±0.53	74.33±0.92	80.62±0.69	67.63±2.82	82.58±0.34	76.33±1.17	89.10±1.86	78.40±0.4	89.03±1.20	74.52±1.29	87.48±2.17	75.29±1.11
Generative Models for Noisy Transition Matrix Estimation												
DualT	78.92±0.56	73.39±1.19	74.66±1.24	67.82±2.33	77.16±0.76	70.61±0.56	83.66±0.97	80.84±2.05	82.11±1.27	79.03±2.59	86.47±1.55	78.84±2.45
CausalNL	81.08±0.54	74.43±1.93	81.22±1.04	71.25±2.70	82.57±0.64	78.91±2.33	86.44±1.98	82.74±2.11	89.87±1.64	79.80±2.22	89.00±1.67	84.62±2.58
NPC	79.82±0.70	72.96±1.84	78.88±0.97	61.69±4.07	79.97±0.46	75.19±0.11	82.83±3.43	75.04±5.53	83.94±1.97	77.69±2.87	86.28±1.17	77.38±4.09
CR w/ NPC	83.09±0.11	77.96±1.00	83.13±0.39	73.50±3.61	83.47±0.16	80.47±0.71	89.69±0.53	83.21±1.06	89.01±1.22	82.54±2.69	88.25±1.49	86.41±1.93
DyGen (Our Method)	**83.82**±0.04	**79.56**±0.93	**83.63**±0.23	**81.98**±0.80	**84.07**±0.17	**81.54**±0.44	**91.42**±0.70	**89.80**±0.58	**91.37**±0.98	**90.43**±1.11	**91.41**±0.49	**88.90**±1.66

Table 2: Main results on real-world noise datasets.

Method	ChemProt	TREC	SEMEVAL
Noise Ratio	22.88%	38.56%	16.00%
Base	64.84±0.28	67.33±0.83	71.44±0.10
Co-Teaching	65.98±0.63	66.61±0.35	72.07±0.76
JoCoR	65.32±0.24	66.50±1.44	70.33±1.10
CR	65.53±0.22	68.33±0.31	71.11±1.07
DualT	65.30±2.18	69.33±1.02	70.88±1.07
CausalNL	67.29±1.37	69.83±2.71	72.22±0.26
NPC	65.15±0.51	70.44±0.39	72.17±0.17
CR w/ NPC	66.46±0.23	71.02±0.43	72.72±0.70
DyGen (Our Method)	**69.07**±0.38	**72.39**±0.82	**73.17**±0.29

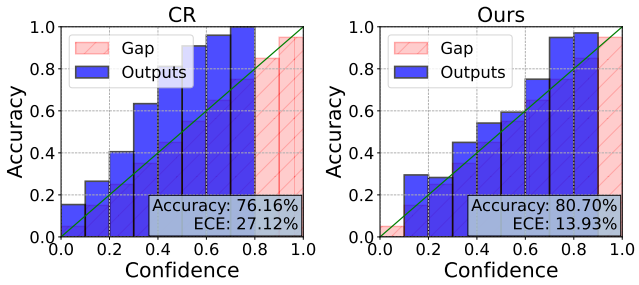


Figure 4: The reliability diagrams of CR baseline and DyGen on 20newsgroup with 40% symmetric noise.

estimation of the corresponding transition matrix simpler. Our observation suggests that the ability to better estimate the transition matrix leads to improved performance in the AG News dataset.

Model Calibration. As a probabilistic denoising method, we find that DyGen also improves model calibration when fine-tuning PLMs with noisy labels [19]. Figure 4 shows the calibration results of the strong CR baseline and DyGen on the 20newsgroup dataset corrupted with 40% symmetric noise. The results suggest that while CR shows some robustness in noisy label learning, it suffers from under-confidence. This is because some model branches with lower confidences will regularize the other ones, leading to an also lower average for prediction. In contrast, DyGen not only improves the classification accuracy on the clean test set but also better calibrates the predictions and reduces the expected calibration error (ECE) from 27.12% to 13.93% (see details in Appendix. F).

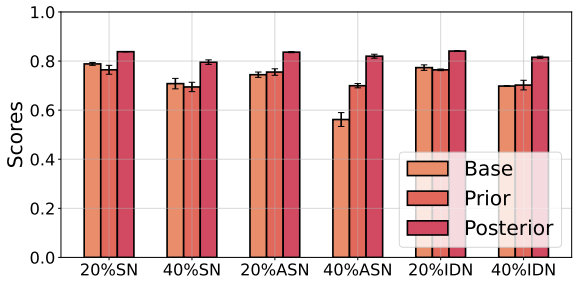
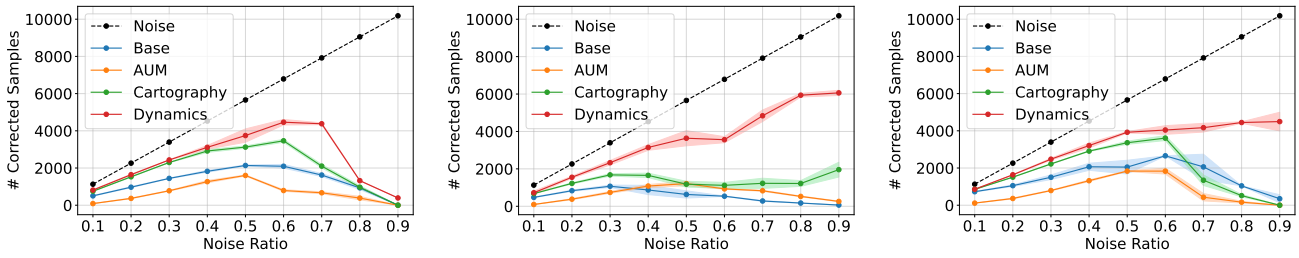


Figure 5: Performance comparison on 20newsgroup dataset between applying prior or posterior for true label prediction.

6.3 Ablation Study

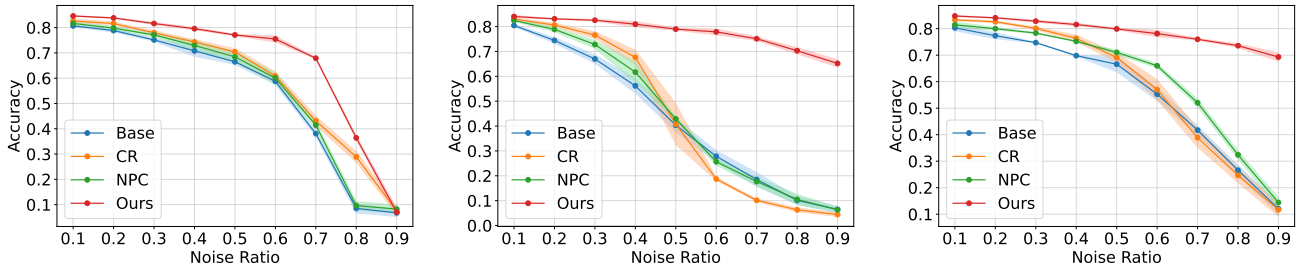
Effect of Different Components. To investigate the effectiveness of each model component, we remove some components of the model to test how the performance varies: (1) Removing the co-regularization mechanism from the Stage I model, degenerating to a simple noisy-supervised classifier; (2) Removing the training dynamics from the dynamics-based prior function, leading to the degeneration to a vanilla KNN prior function; (3) Removing the co-regularization mechanism from the Stage II model, resulting in the degeneration to the original NPC architecture. Table 3 shows the impact of removing components from DyGen on both synthetic (20newsgroup) and real-world (ChemProt) datasets. The results reveal that as more components are taken away, the performance of the model deteriorates, emphasizing the significant contribution of each component to the overall performance. The co-regularization mechanism in the second stage proves to be more effective when it is also applied in the first stage. This is because multiple branches of the co-regularized second stage model generate consistent $q_\phi(y|\hat{y}, \mathbf{x})$ estimates based on the same input $p(\hat{y}|\mathbf{x})$ and prior knowledge.

Comparing Prior and Posterior. Figure 5 compares the performance of the posterior distribution of the generative model against that of the prior distribution used directly for posterior inference. It indicates that while using prior for inference can produce comparable or even better results than the noisy-supervised classifier in the first stage, it still falls short compared to the posterior produced by the generative model in the second stage. This highlights the importance of the second stage, which uses a co-regularized generative model for calibrating noisy predictions and refining the imperfect prior knowledge to only retain the key information.



(a) SN (b) ASN (c) IDN

Figure 6: Numbers of samples corrected by various training dynamic patterns as prior on 20newsgroup dataset.



(a) SN (b) ASN (c) IDN

Figure 7: Performance curves under different noise ratios on 20newsgroup dataset.

Table 3: Ablation studies in DyGen on synthetic noise dataset 20newsgroup. "I" indicates the stage I model; "P" represents the prior function; "II" refers to the stage II model.

I	P	II	20% SN	40% SN	20% ASN	40% ASN	20% IDN	40% IDN	ChemProt
✗	✗	✗	78.84±0.59	70.81±2.13	74.44±1.09	56.18±2.82	77.33±1.12	69.81±0.16	65.15±0.51
✗	✗	✓	79.77±0.51	72.93±0.20	78.84±1.01	61.76±4.19	80.00±0.37	75.20±0.08	67.05±0.40
✗	✓	✗	81.44±0.31	77.06±0.04	82.05±0.66	79.49±0.49	81.34±0.22	77.99±0.44	65.03±0.51
✓	✗	✗	83.09±0.11	77.96±1.00	83.13±0.39	73.50±3.61	83.47±0.16	80.68±0.62	65.46±0.22
✗	✓	✓	81.64±0.06	76.34±0.92	82.05±0.66	79.51±0.57	81.37±0.23	78.00±0.65	67.23±0.84
✓	✓	✗	83.58±0.16	79.67±0.21	83.04±0.5	81.54±0.82	83.35±0.62	81.08±0.49	67.44±0.31
✓	✗	✓	83.56±0.10	78.45±1.05	83.57±0.57	75.22±0.04	83.46±0.39	80.89±0.35	66.48±0.34
✓	✓	✓	**83.82**±0.04	**79.90**±0.51	**83.63**±0.23	**82.31**±0.23	**84.07**±0.17	**81.54**±0.44	**69.07**±0.38

6.4 Quality Analysis of Prior

To evaluate the quality of the dynamics-based prior (**Dynamics**) in DyGen, we compare with a set of state-of-the-art training dynamic patterns and treat them as the prior knowledge: (1) **Base** [5] is the original KNN-based prior used in NPC; (2) **AUM** [37] measures the average difference between the logit values for a sample's assigned class and its highest non-assigned class; (3) **Cartography** [41] observes that instances with smaller mean and standard deviations on output probabilities during the training process often correspond to labeling errors. Figure 6 shows the numbers of samples corrected with different prior knowledge. The results demonstrate that our proposed dynamics-based prior consistently achieves superior performance across varying noise types and ratios, highlighting its effectiveness in supplying high-quality true-label information for the subsequent generative model for noisy prediction calibration.

6.5 Performance with Large Noise Ratio

Figure 7 displays the evaluation of models under large noise ratios (≥ 50%). The results demonstrate the robustness of DyGen to large noise ratios. Besides, we can also observe that DyGen shows increased performance gains, compared with the other methods, as the magnitude of label noise increases. This observation also exists in Table 1. As is also shown in Figure 6, this can be attributed to its dynamics-based prior function, which provides high-quality true-label information to the generative model in the second stage.

7 CONCLUSION

In this paper, we focus on leveraging training dynamics to correct noisy predictions by considering the larger distances between noisy samples and their assigned label clusters. In comparison to clean samples, the noisy samples consistently exhibit a higher mean and deviation in distances throughout 1,500 experiments, resulting in a noticeable discrepancy in their training patterns during PLM fine-tuning. To enhance the quality of prior knowledge and improve robustness to noisy labels, we propose DyGen, a framework for noisy label learning that integrates training dynamics patterns with deep generative models. We leverage the agreement from multiple branches optimized by the co-regularization loss, as opposed to solely relying on potentially unreliable noisy labels. Our proposed method, DyGen, demonstrates an average accuracy improvement of 2.55% on five benchmark datasets with both synthetic and real-world noise. Moreover, we conducted extensive experiments to validate the effectiveness of each component. We believe that this study opens up new possibilities in the topics of using training trajectories to handle noisy labels, especially in calibrating noisy predictions under large noise ratios.

ACKNOWLEDGMENTS

This work was supported in part by NSF (IIS2008334, IIS-2106961, CAREER IIS-2144338), ONR (MURI N00014-17-1-2656).

REFERENCES

[1] Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED Revisited: A Thorough Evaluation of the TACRED Relation Extraction Task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 1558–1569.

[2] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin Mcguinness. 2019. Unsupervised Label Noise Modeling and Loss Correction. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 312–321. https://proceedings.mlr.press/v97/arazo19a.html

[3] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A Closer Look at Memorization in Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 233–242. https://proceedings.mlr.press/v70/arpit17a.html

[4] Abhijeet Awasthi, Sabyasachi Ghosh, Rasna Goyal, and Sunita Sarawagi. 2020. Learning from Rules Generalizing Labeled Exemplars. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SkeuexBtDr

[5] Heesun Bae, Seungjae Shin, Byeonghu Na, Joonho Jang, Kyungwoo Song, and Il-Chul Moon. 2022. From Noisy Prediction to True Label: Noisy Prediction Calibration via Generative Model. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*. PMLR, 1277–1297.

[6] Dara Bahri, Heinrich Jiang, and Maya Gupta. 2020. Deep k-NN for Noisy Labels. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 540–550.

[7] Antonin Berthon, Bo Han, Gang Niu, Tongliang Liu, and Masashi Sugiyama. 2021. Confidence Scores Make Instance-dependent Label-noise Learning Possible. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 825–836.

[8] Avrim Blum, Adam Kalai, and Hal Wasserman. 2003. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)* 50, 4 (2003), 506–519.

[9] Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. 2021. Learning with Instance-Dependent Label Noise: A Sample Sieve Approach. In *International Conference on Learning Representations*. https://openreview.net/forum?id=2VXyy9mIyU3

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186.

[11] Garrett B. Goh, Charles Siegel, Abhinav Vishnu, and Nathan Hodas. 2018. Using Rule-Based Labels for Weak Supervised Learning: A ChemNet for Transferable Chemical Property Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (London, United Kingdom) *(KDD '18)*. Association for Computing Machinery, New York, NY, USA, 302–310. https://doi.org/10.1145/3219819.3219838

[12] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. arXiv:arXiv:2007.15779

[13] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*. PMLR, 1321–1330.

[14] 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 *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 4006–4016.

[15] 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 *Advances in Neural Information Processing Systems*, Vol. 31. Curran Associates, Inc.

[16] 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 *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*. PMLR, 2304–2313.

[17] Weonyoung Joo, Wonsung Lee, Sungrae Park, and Il-Chul Moon. 2020. Dirichlet variational autoencoder. *Pattern Recognition* 107 (2020), 107514.

[18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[19] Lingkai Kong, Haoming Jiang, Yuchen Zhuang, Jie Lyu, Tuo Zhao, and Chao Zhang. 2020. Calibrated Language Model Fine-Tuning for In-and Out-of-Distribution Data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1326–1340.

[20] Kamran Kowsari, Donald E Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, , Matthew S Gerber, and Laura E Barnes. 2017. HDLTex: Hierarchical Deep Learning for Text Classification. In *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on*. IEEE.

[21] Martin Krallinger, Obdulia Rabal, Saber A Akhondi, Martın Pérez Pérez, Jesús Santamaría, Gael Pérez Rodríguez, Georgios Tsatsaronis, Ander Intxaurrondo, José Antonio López, Umesh Nandal, et al. 2017. Overview of the BioCreative VI chemical-protein interaction Track. In *Proceedings of the sixth BioCreative challenge evaluation workshop*, Vol. 1. 141–146.

[22] Ken Lang. 1995. NewsWeeder: Learning to Filter Netnews. In *Machine Learning Proceedings 1995*. Morgan Kaufmann, 331–339.

[23] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 4 (2020), 1234–1240.

[24] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864* (2020).

[25] Junnan Li, Richard Socher, and Steven C.H. Hoi. 2020. DivideMix: Learning with Noisy Labels as Semi-supervised Learning. In *International Conference on Learning Representations*.

[26] Xuefeng Li, Tongliang Liu, Bo Han, Gang Niu, and Masashi Sugiyama. 2021. Provably end-to-end label-noise learning without anchor points. In *International Conference on Machine Learning*. 6403–6413.

[27] Xin Li and Dan Roth. 2002. Learning Question Classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

[28] Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. BOND: BERT-Assisted Open-Domain Named Entity Recognition with Distant Supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 1054–1064. https://doi.org/10.1145/3394486.3403149

[29] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. 2020. Early-Learning Regularization Prevents Memorization of Noisy Labels. In *Advances in Neural Information Processing Systems*, Vol. 33. 20331–20342.

[30] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[31] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. 2020. Normalized Loss Functions for Deep Learning with Noisy Labels. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 6543–6553.

[32] Eran Malach and Shai Shalev-Shwartz. 2017. Decoupling" when to update" from" how to update". *Advances in neural information processing systems* 30 (2017).

[33] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 29.

[34] Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. 2020. SELF: Learning to Filter Noisy Labels with Self-Ensembling. In *International Conference on Learning Representations*.

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[36] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[37] Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. 2020. Identifying Mislabeled Data using the Area Under the Margin Ranking. In *Advances in Neural Information Processing Systems*, Vol. 33. 17044–17056.

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

[39] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. *Commun. ACM* 64, 9 (aug 2021), 99–106.

[40] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[41] Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset Cartography: Mapping and Diagnosing Datasets with Training Dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 9275–9293.

[42] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2018. Joint Optimization Framework for Learning With Noisy Labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

[44] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. 2019. Symmetric Cross Entropy for Robust Learning With Noisy Labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[45] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. 2020. Combating Noisy Labels by Agreement: A Joint Training Method with Co-Regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[46] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 1112–1122.

[47] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, et al. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 38–45.

[48] Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. 2021. Robust early-learning: Hindering the memorization of noisy labels. In *International Conference on Learning Representations*.

[49] Xiaobo Xia, Tongliang Liu, Bo Han, Mingming Gong, Jun Yu, Gang Niu, and Masashi Sugiyama. 2022. Sample Selection with Uncertainty of Losses for Learning with Noisy Labels. In *International Conference on Learning Representations*.

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

[51] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. 2015. Learning From Massive Noisy Labeled Data for Image Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[52] Ran Xu, Yue Yu, Hejie Cui, Xuan Kan, Yanqiao Zhu, Joyce Ho, Chao Zhang, and Carl Yang. 2023. Neighborhood-Regularized Self-Training for Learning with Few Labels. In *Thirty-Seventh AAAI Conference on Artificial Intelligence*.

[53] Yu Yao, Tongliang Liu, Mingming Gong, Bo Han, Gang Niu, and Kun Zhang. 2021. Instance-dependent Label-noise Learning under a Structural Causal Model. In *Advances in Neural Information Processing Systems*, Vol. 34. Curran Associates, Inc., 4409–4420.

[54] Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. 2020. Dual t: Reducing estimation error for transition matrix in label-noise learning. *Advances in neural information processing systems* 33 (2020), 7260–7271.

[55] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. 2019. How does Disagreement Help Generalization against Label Corruption?. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 7164–7173.

[56] Yue Yu, Lingkai Kong, Jieyu Zhang, Rongzhi Zhang, and Chao Zhang. 2022. AcTune: Uncertainty-Based Active Self-Training for Active Fine-Tuning of Pretrained Language Models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1422–1436.

[57] Yue Yu, Chenyan Xiong, Si Sun, Chao Zhang, and Arnold Overwijk. 2022. COCO-DR: Combating Distribution Shifts in Zero-Shot Dense Retrieval with Contrastive and Distributionally Robust Learning. *arXiv preprint arXiv:2210.15212*.

[58] Yue Yu, Yuchen Zhuang, Rongzhi Zhang, Yu Meng, Jiaming Shen, and Chao Zhang. 2023. ReGen: Zero-Shot Text Classification via Training Data Generation with Progressive Dense Retrieval. arXiv:2305.10703 [cs.CL]

[59] Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. 2021. Fine-Tuning Pre-trained Language Model with Weak Supervision: A Contrastive-Regularized Self-Training Approach. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 1063–1077.

[60] Jieyu Zhang, Yue Yu, Yinghao Li, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. 2021. WRENCH: A Comprehensive Benchmark for Weak Supervision. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. https://openreview.net/forum?id=Q9SKS5k8io

[61] Rongzhi Zhang, Yue Yu, Pranav Shetty, Le Song, and Chao Zhang. 2022. Prboost: Prompt-based rule discovery and boosting for interactive weakly-supervised learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 745–758.

[62] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems* 28 (2015).

[63] Yivan Zhang, Gang Niu, and Masashi Sugiyama. 2021. Learning Noise Transition Matrix from Only Noisy Labels via Total Variation Regularization. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 12501–12512.

[64] Zhilu Zhang and Mert Sabuncu. 2018. Generalized Cross Entropy Loss for Training Deep Neural Networks. In *Advances in Neural Information Processing Systems*, Vol. 31. Curran Associates, Inc.

[65] Wenxuan Zhou and Muhao Chen. 2021. Learning from Noisy Labels for Entity-Centric Information Extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 5381–5392.

[66] Wenxuan Zhou, Hongtao Lin, Bill Yuchen Lin, Ziqi Wang, Junyi Du, Leonardo Neves, and Xiang Ren. 2020. Nero: A neural rule grounding framework for label-efficient relation extraction. In *Proceedings of The Web Conference 2020*. 2166–2176.

[67] Yuchen Zhuang, Yinghao Li, Junyang Zhang, Yue Yu, Yingjun Mou, Xiang Chen, Le Song, and Chao Zhang. 2022. ReSel: N-ary Relation Extraction from Scientific Text and Tables by Learning to Retrieve and Select. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 730–744.

Table 5: Main results on synthetic noise dataset WOS.

Dataset (→)	WOS					
Method (↓)	20% **SN**	40% **SN**	20% **ASN**	40% **ASN**	20% **IDN**	40% **IDN**
Basic Performances						
Base	78.35±0.14	76.01±0.82	78.03±0.24	65.13±0.42	78.19±0.21	76.36±0.31
Multi-Model Training Strategies						
Co-Training	79.31±0.53	77.79±1.60	79.10±0.87	67.61±1.45	78.43±0.39	76.03±1.35
JoCoR	78.45±0.63	77.96±0.42	78.46±0.41	66.38±0.92	78.49±0.49	76.47±1.36
CR	78.25±0.13	76.35±0.17	77.41±0.82	62.32±4.38	77.48±0.93	75.47±2.03
Generative Models for Noisy Transition Matrix Estimation						
DualT	78.99±0.98	76.74±1.22	78.58±0.68	67.27±0.79	78.35±0.29	76.79±0.71
CausalNL	78.56±1.06	76.97±0.88	79.19±1.25	65.43±0.75	78.46±0.59	76.53±0.96
NPC	78.80±0.34	77.80±0.86	79.21±0.80	68.36±0.30	78.86±0.25	77.50±0.51
CR w/ NPC	79.18±0.47	78.15±0.98	79.28±0.33	70.79±0.96	79.34±0.42	77.87±1.13
DyGen	**79.95**±0.10	**78.68**±0.26	**79.55**±0.23	**72.50**±3.17	**79.62**±0.19	**78.09**±0.37

A DATASET DETAILS

Table 4: Detailed dataset statistics.

Datasets	# Training	# Validation	# Test	# Label
20newsgroup	9051	2263	7532	20
AG NEWS	40000	7600	7600	4
ChemProt	12861	1607	1607	10
TREC	4965	500	500	6
SemEval	1749	200	692	9
WOS	22552	5639	18794	134

In this work, we select **20newgroup** [22] and **AG NEWS** [27, 62] for news topic classification for experiments on synthetic noise datasets. We also conducted experiments on an additional synthetic noise dataset, **WOS** [20]. The results of these experiments are available in appendix B. Table. 4 introduces detailed statistics about datasets used in our experiments with both synthetic and real-world noise. Since these datasets are assumed to have no noisy labels, we manage three types of noisy label for noisy label injection. In this part, we explain the details of these noisy generation processes. Since we include a detailed explanation of symmetric and asymmetric noise in § 6.1, we will emphasize on the instance-dependent noise (IDN) in the following. Specifically, we follow the noise generation process in existing literature [5, 9, 50, 53] for IDN generation in our experiments. The detailed algorithm of IDN noisy label generation is summarized in Alg. 3:

Algorithm 3: Instance Dependent Noise Generation.

Input: Clean samples $(\mathbf{x}_i, y_i)_{i=1}^n$; Noise ratio τ.
Sample instance flip rates $q \in \mathbb{R}^n$ from the truncated normal distribution $\mathcal{N}(\tau, 0.1^2, [0, 1])$.
Independently sample $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_c$ from the standard normal distribution $\mathcal{N}(0, 1^2)$.
for $e = 1, 2, \cdots, n$ **do**
  $p = \mathbf{x}_i \times \mathbf{v}_{y_i}$.
  $p_{y_i} = -\inf$.
  $p = q_i \times \text{softmax}(p)$.
  $p_{y_i} = 1 - q_i$.
  Randomly choose a label from the label space according to probabilities p as noisy label $\tilde{y}_i$.
Output: Noisy samples $(\mathbf{x}_i, \tilde{y}_i)_{i=1}^n$.

In addition, we conduct experiments on three real-world datasets, including: **ChemProt** [21] for chemical-protein interaction classification, **TREC** [4] for question classification, and **SemEval** [66] for relation extraction. The generation of noisy labels on TREC and ChemProt datasets are introduced in Appendix. C. We follow the same process as [66] to generate noisy labels for SemEval.

B ADDITIONAL RESULTS ON WOS

Table. 5 displays the additional experimental result on **WOS** [20].

C DETAILS FOR WEAK SUPERVISION

The examples of semantic rules on ChemProt are given in Table. 6.

D BASELINES DETAILS

We compare with the most relevant state-of-the-art baselines on learning with noisy labels, including: (1) **Base** [10] is the $\text{BERT}_{\text{base}}$ model fine-tuned only with standard cross-entropy loss and early stopping; (2) **Co-Teaching** [15] trains two different networks and feeds the samples with small loss to each other for parameter updating; (3) **JoCoR** [45] also trains two networks and selects the samples, for which the sum of the losses from two networks is small, as clean samples; (4) **CR** [65] is another method of training multiple networks and uses a soft target to regularize each model; (5) **DualT** [54] factorizes the transition probability matrix into the product of two independently-estimated matrices to mitigate the error in the estimation; (6) **CausalNL** [53] introduces an auxiliary latent variable $\mathbf{z}$, generating $\mathbf{x}$ together with y, and proposes a structural causal model for instance-dependent noise learning; (7) **NPC** [5] proposes a deep generative model to estimate the transition from noisy predictions to true labels; (8) **CR w/ NPC** treats NPC as a post-processing module and links it after the CR method, which achieves the best post-performance of NPC-based methods we obtain empirically.

E IMPLEMENTATION DETAILS

All experiments are conducted on CPU: Intel(R) Core(TM) i7-5930K CPU @ 3.50GHz and GPU: NVIDIA GeForce RTX A5000 GPUs using python 3.8 and PyTorch 1.12. Table. 7 shows the detailed hyper-parameter configuration. In addition, we search the batch size in {8, 16, 32, 64} and use Adam [18] as optimizer.

F MODEL CALIBRATION

Machine learning applications usually require trustworthy predictions that need to be not only accurate but also well-calibrated. Specifically, a well-calibrated model is expected to output prediction confidence comparable to its classification accuracy. More precisely, for a data point $\mathbf{x}_i$, we denote by y_i the ground-truth label, $\hat{y}_i$ the prediction made by the model, and $\hat{p}(\mathbf{x})$ the output probability associated with the prediction. The calibration error of the predictive model for a given confidence $p \in (0, 1)$ is defined as:

$$\epsilon_p = |\mathbb{P}(\hat{y} = y | \hat{p}(\mathbf{x}) = p) - p|. \tag{22}$$

We partition all data points into M bins of equal size according to their prediction confidences bounded l_m and u_m. Then for any $p \in [l_m, u_m)$, we define the empirical calibration error [13] as:

$$\hat{\epsilon}_p = \hat{\epsilon}_m = \frac{1}{|\mathcal{B}_m|} \sum_{i \in \mathcal{B}_m} [\mathbb{1}(\hat{y}_i = y_i) - \hat{p}_i]|, \tag{23}$$

Table 6: Examples of semantic rules on Chemprot.

Rule	Example
`HAS (x, [amino acid,mutant, mutat, replace] ) → part_of`	A major part of this processing requires endoproteolytic cleavage at specific pairs of basic [CHEMICAL]amino acid[CHEMICAL] residues, an event necessary for the maturation of a variety of important biologically active proteins, such as insulin and [GENE]nerve growth factor[GENE].
`HAS (x, [bind, interact, affinit] ) → regulator`	The interaction of [CHEMICAL]naloxone estrone azine[CHEMICAL] (N-EH) with various [GENE]opioid receptor[GENE] types was studied in vitro.
`HAS (x, [activat, increas, induc, stimulat, upregulat] ) → upregulator/activator`	The results of this study suggest that [CHEMICAL]noradrenaline[CHEMICAL] predominantly, but not exclusively, mediates contraction of rat aorta through the activation of an [GENE]alphalD-adrenoceptor[GENE].
`HAS (x, [downregulat, inhibit, reduc, decreas] ) → downregulator/inhibitor`	These results suggest that [CHEMICAL]prostacyclin[CHEMICAL] may play a role in downregulating [GENE]tissue factor[GENE] expression in monocytes, at least in part via elevation of intracellular levels of cyclic AMP.
`HAS (x, [ agoni, tagoni]* ) → agonist` * (note the leading whitespace in both cases)	Alprenolol and BAAM also caused surmountable antagonism of [CHEMICAL]isoprenaline[CHEMICAL] responses, and this [GENE]beta 1-adrenoceptor[GENE] antagonism was slowly reversible.
`HAS (x, [antagon] ) → antagonist`	It is concluded that [CHEMICAL]labetalol[CHEMICAL] and dilevalol are [GENE]beta 1-adrenoceptor[GENE] selective antagonists.
`HAS (x, [modulat, allosteric] ) → modulator`	[CHEMICAL]Hydrogen sulfide[CHEMICAL] as an allosteric modulator of [GENE]ATP-sensitive potassium channels[GENE] in colonic inflammation.
`HAS (x, [cofactor] ) → cofactor`	The activation appears to be due to an increase of [GENE]GAD[GENE] affinity for its cofactor, [CHEMICAL]pyridoxal phosphate[CHEMICAL] (PLP).
`HAS (x, [substrate, catalyz, transport, produc, conver] ) → substrate/product`	Kinetic constants of the mutant [GENE]CrAT[GENE] showed modification in favor of longer [CHEMICAL]acyl-CoAs[CHEMICAL] as substrates.
`HAS (x, [not] ) → not`	[CHEMICAL]Nicotine[CHEMICAL] does not account for the CSE stimulation of [GENE]VEGF[GENE] in HFL-1.

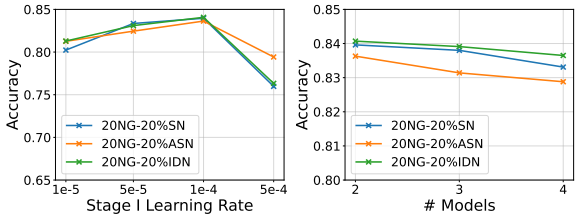


Figure 8: Parameter studies on learning rate of training dynamic pattern encoding and the number of model branches under different noise types on 20newsgroup dataset.

Table 7: Hyper-parameter configurations.

	20newsgroup	AG NEWS	WOS	ChemProt	TREC	SemEval
Max Length	150	256	256	512	64	128
Batch Size	64	32	32	16	64	32
Learning Rate	1e-4	1e-4	1e-4	2e-5	5e-6	5e-6
# Stage I Epochs			10			
# Stage II Iterations			10			
$(\lambda_1, \lambda_2, \delta, \rho)$			(5,1,1,2)			

where y_i, $\hat{y}_i$, and $\hat{p}_i$ are the true label, prediction, and confidence of sample i. $|\mathcal{B}_m|$ is the sample size of m-th bin. To evaluate the overall calibration error of the predictive model, we can further take a weighted average of the calibration errors of all bins, which is also known as the **Expected calibration error (ECE)** [33]. The metric is defined as:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} |\text{acc}(B_m) - \text{conf}(B_m)| = \sum_{m=1}^{M} \frac{|B_m|}{N} \hat{\epsilon}_m, \qquad (24)$$

where N is the number of samples. Specifically, we set $M = 10$.

F.1 Parameter Study

We conduct experiments to investigate the impact of two hyper-parameters on the performance of DyGen: the learning rate of the training pattern encoding, which may affect the quality of the prior, and the number of branches in the co-regularization mechanism. The other hyper-parameters remain the same as the default.

Effect of Stage I Learning Rate. The choice of the learning rate plays a crucial role in encoding the training dynamics patterns. When the learning rate is set too low, the training trajectory pattern may be extended during the learning process, leading to a relatively slight impact on the model performance. On the other hand, if the learning rate is set too high, the pattern may change too rapidly, causing overfitting to the noisy labels and a decrease in performance. Furthermore, a high learning rate can also produce low-quality probability distributions $p(\hat{y}|\mathbf{x})$, hindering the ability of the subsequent model to perform noisy prediction calibration.

Effect of Model Branches. Generally, the number of branches does not severely influence the model performance. We hypothesize that including more branches could be gradually more difficult for models to reach a consensus on incoming noisy samples.