Learning on Imbalanced Noisy Data via Debiased Sample Selection and LLM-Driven Annotation

Bo Yuan[1,2*], **Yulin Chen**[3], **Yin Zhang**[3†]

[1]Test Center, National University of Defense Technology, Xi'an, China
[2]Shanxi Key Laboratory of Intelligence Coordination Networks, Xi'an, China
[3]Zhejiang University, Hangzhou, China
byuan.186@qq.com, yulinchen@zju.edu.cn, yinzh@zju.edu.cn

Łearning with Noisy Labels (LNL) is a challenge that arises in many scenarios where training data can contain noisy labels. While various methods, such as active learning for small language models (SLMs), have been proposed to re-annotate samples, they still require human efforts. The prevalent in-context learning (ICL) of large language models (LLMs) can perform text annotation tasks, but their efficiency hinges on the precise selection of clean versus noisy samples from noisy data. Meanwhile, this paper observes that existing sample selection suffers from selection and training bias on class imbalance noisy datasets, leading to decreased accuracy in sample selection. To bridge it, we propose a debiased sample selection and an innovative collaborative learning framework DeCo based on active learning for LNL. During collaborative learning, we first conduct debiased sample selection by designing a robust expert model based on SLMs and introducing a dynamic class-wise threshold strategy, then feed selected clean samples to active annotator LLMs for re-annotating noisy samples using ICL, with the re-annotation results reinforcing SLMs' training for subsequent accurate selection. Ultimately, we employ distinct loss functions adept at managing subsets with varying degrees of label noise. Extensive experimental results on synthetic and real-world datasets demonstrate the effectiveness and superiority of our proposed method.

1 Introduction

The core of deep neural networks' success lies in data scale and annotation quality. However, obtaining large-scale high-quality datasets is expensive and time-consuming in practical scenarios. To obtain large-scale data under a limited cost, some researchers collect data by web-crawling (Li et al., 2017) or crowd-sourcing (Yan et al., 2014), which could inevitably incur wrong (noisy) labels. Noisy labels will mislead the learning patterns and subsequently result in incorrect predictions. To mitigate the side-effect of label noise, LNL (Zhang et al., 2018; Ma et al., 2020; Zhang et al., 2021) has attracted huge attention from the community. Among them, the most straightforward and effective way is active learning (Zhao et al., 2011; Younesian et al., 2021), querying experts to provide accurate labels for the identified noisy samples. These methods are a prominent solution but still require human effort to annotate partial samples.

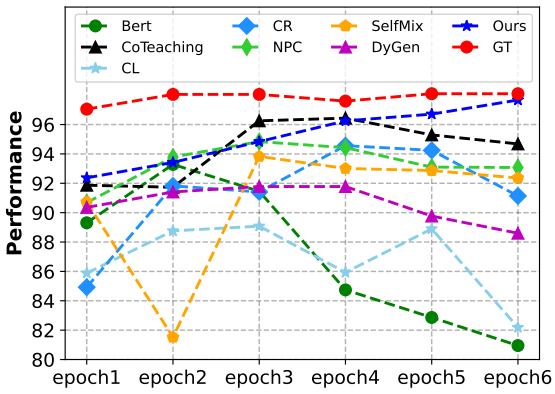


Figure 1: Comparions of our method with current LNL methods on the imbalanced dataset R8 under 40% asymmetric noise. Benefiting from debiased sample selection, our method outperforms others and nearly matches the performance of training with ground truth labels (GT).

To avoid human costs in the annotation, we resort to LLMs, such as ChatGPT. LLMs have shown remarkable performance on downstream tasks by ICL. With only a few task demonstrations, ICL outperforms zero-shot inference on various tasks (*e.g.*, classification and generation), making it a compelling alternative to supervised fine-tuning or human annotating (Shu et al., 2019; Gilardi et al., 2023). In particular, the success of ICL heavily relies on the quality of demonstrations selected from a large set of annotated examples. For them, input-label mappings solicited from humans or LLMs can

*Corresponding Author
†Corresponding Author

Findings of the Association for Computational Linguistics: ACL 2026, pages 30504–30542

often be noisy, especially in complex tasks (Kossen et al., 2023; Gao et al., 2024). This gives rise to the importance of *selecting clean samples to construct high-quality demonstrations and identifying noisy samples to active query LLMs*.

Existing sample selection methods usually select clean samples from noisy data by setting a global fixed threshold for loss value, then regard the samples with small losses as clean samples. However, real-world scenarios contain not only noisy labels but also class imbalances. Most samples belong to the majority classes (*i.e.*, head classes), while others (*i.e.*, tail classes) have very few samples. In such scenarios, samples from tail classes tend to be overlooked and have large loss values (see Figure 5). This prevents current methods from accurately selecting clean samples from the tail classes, resulting in a highly imbalanced selected set (selection bias)(see Figure 4). Based on further analysis (see Appendix A), we observed that each class has different loss distributions, and the loss distributions dynamically evolve during training. So it is not ideal to handle all classes equally and set a global fixed loss threshold. Besides, current methods always adopt a self-training manner and exist accumulated error (*i.e.*, training bias)(Li et al., 2020), which makes model memorize itself mistakes and hard to discern noisy labels by itself (Xiao et al., 2023). This means that training models on selected class-imbalanced biased sets will continuously lead to sub-optimal results (see Figure 1). While current methods can divide datasets, they struggle to accurately select clean samples in scenarios when label noisy coexists with class imbalance, *thus failing to construct high-quality demonstrations for LLMs.*

To circumvent the above problems, we introduce DeCo, a novel collaborative learning paradigm that combines LLMs with SLMs through active learning, using SLMs to filter noisy data and distilling knowledge from LLMs. Specifically, we first design a robust model (MEM) that integrates the SLM with multiple expert layers to reduce the training bias. Our motivation is to perform SLM learning and sample selection on different output layers, which helps prevent SLM from training on its own incorrect selections, thereby reducing the accumulated error. Then we propose a dynamic class-wise threshold strategy based on our observation (each class has distinct and dynamically evolving loss distributions, see Figure 12) to mitigate selection bias. With the output of MEM and the proposed strategy, we perform debiased sample selection to select clean and noisy samples for the LLM. To integrate SLMs and LLMs as a whole, a collaborative learning framework is designed where SLMs act as filters to divide noisy datasets into different subsets, and the LLM serves as an active annotator to correct noisy samples from subsets. During collaborative learning, SLMs can learn the knowledge of LLMs to boost their performance, while LLMs can also benefit from the divided clean samples to boost ICL. Overall, our main contributions are:

- Based on the LNL problem under class imbalance, we innovatively utilize MEM and a dynamic class-wise threshold strategy, alleviating training bias and selection bias, to select clean and noisy samples from noisy data.
- We propose a novel collaborative learning framework called DeCo to employ the SLMs as filters and the LLMs as active annotators to learn from class-imbalance noisy datasets.
- We provide extensive experimental results on synthetic and real-world noisy datasets, demonstrating the superiority of our framework compared to existing baselines.

2 Related Work

Learning With Noisy Labels. Previous sample selection methods typically utilize the small-loss criterion (Han et al., 2018; Shu et al., 2019; Qiao et al., 2022), applying a global fixed loss threshold to segregate the noisy training set and considering samples with smaller losses as clean. However, these methods usually rely on the class-balanced hypothesis, rendering them inadequate for addressing noisy and imbalanced datasets in real-world scenarios. Based on our observation that the loss distributions dynamically evolve during training and each class has a different loss distribution, we propose a dynamic class-wise threshold strategy to improve these methods tackling the concurrent label noise and class imbalance issues.

Class Imbalance. Real-world scenarios simultaneously contain noisy labels and class imbalances, posing a more challenging problem. Prior works mainly resort to the sample re-weighting strategy for addressing class imbalance (Ren et al., 2018; Jiang et al., 2022; Huang et al., 2022). These methods usually assign larger weights to tail classes while smaller weights to head classes. However, existing approaches are usually vulnerable when training with noisy and imbalanced data. It should be noted that noisy and tail class samples exhibit high losses. Noisy samples require smaller weights,

while tail class samples require larger weights. Instead of following the re-weighting paradigm, we introduce a class-wise threshold strategy in our method to ensure that tail classes are sufficiently learned during training.

3 Background

Given a training data $\mathcal{D}$=$\{(x_i, y_i)\}_{i=1}^{N}$ with N samples and K classes, where x is the text, $y \in [1, K]$ is possibly incorrect label. Denote the output of the final layer in the model for text x_i as $z(x_i) \in \mathbb{R}^K$. The confidence of x_i for each class k can be represented as follows: $p(k; x) = \frac{e^{z(k;x)}}{\sum_{k=1}^{K} e^{z(k;x)}}$. We follow Li et al. (2023) to quantify SLM's memorization strength through a confidence metric and regard $p(k; x)$ as the value of memorization strength. For text classification tasks, if a model memorizes a text, its confidences $p(k; x)$ of k exceed a certain threshold or reach the maximum.

4 Methodology

In this section, we introduce our proposed framework, DeCo, which aims to address LNL problems under class imbalance in the LLMs era. While LLMs can generate new labels for noisy samples, they still rely on SLMs to effectively distinguish and separate noisy data. In each training loop, we alternate the following steps: (1) Training a MEM and performing debiased sample selection to divide the noisy data into three subsets. (2) Selecting clean and noisy samples from subsets, active querying LLMs correct the noisy samples, and the clean samples are used to prompt ICL. (3) Learning from these subsets in different ways. Figure 2 shows the overall framework of DeCo. In what follows, we will elaborate on our proposed DeCo framework.

4.1 Debiased Sample Selection

In this step, MEM and a dynamic class-wise threshold strategy are introduced to address training and selection biases separately, aiming to achieve debiased sample selection.

4.1.1 MEM: Multiple Expert Model

The training bias (confirmation bias) of current sample selection stems from a self-training manner (Tarvainen and Valpola, 2017; Li et al., 2020), which will maintain or exacerbate the class imbalance rate of selected clean samples in the training process (see Figure 14). Motivated by the Mixture-of-experts (Rokach, 2010), we propose a robust architecture MEM that integrates multiple expert layers $\{g_1, \ldots, g_m\}$ with size m into the classifier layer f of SLM, which independently conducts SLM learning and sample selection on different layers. Specifically, the sample selection stage only takes place at different expert layers, avoiding the involvement of f. This prevents f from being trained on its own chosen noisy samples, thereby reducing training bias caused by self-training. Meanwhile, different expert layers offer diverse evidence (confidence) to memorize samples, which can be emsembled to prompt more robust selection.

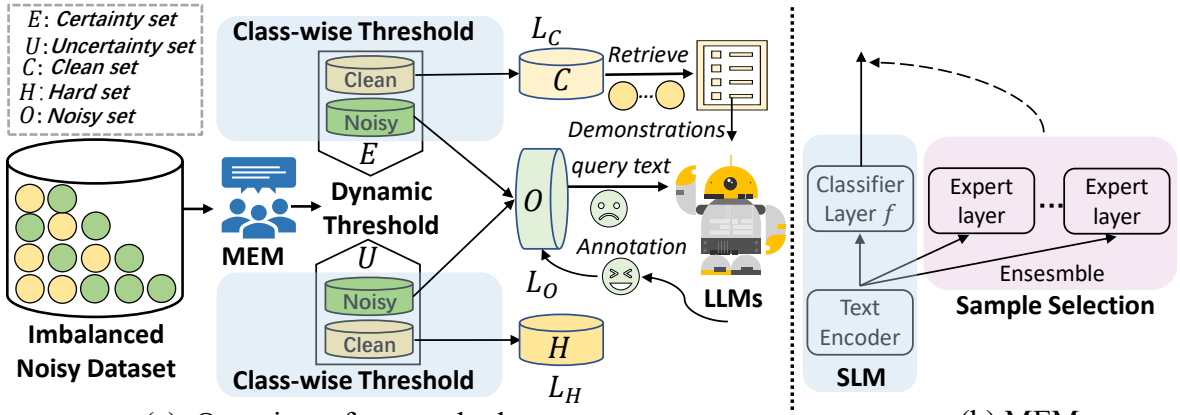
4.1.2 Dynamic Class-wise Threshold Strategy

As analyzed in Sec. 1, current sample selection methods fail to select clean samples from the tail classes and cause the selected set to be highly class-imbalanced (selection bias) in the class-imbalanced noisy dataset (see Figure 4), which further affects the model's generalization and results in extremely low test accuracy on tail classes (see Figure 14). In this scenario, it is inappropriate to set a global fixed threshold. Thus, we introduce a novel dynamic class-wise threshold strategy to provide both dynamic and class-wise thresholds.

Dynamic Threshold. For the tail class, we observe that the loss value of noisy samples decreases during training (see Appendix A). The observed phenomena could potentially be attributed to the memory effect of SLMs, *i.e.*, the memory strength of the model for given labels towards each sample is getting stronger with the increase of learning (Li et al., 2023). That is to say, as training continues, the noisy samples will be memorized gradually, and their loss value accordingly decreases. So, we contend that the testing curriculum for evaluating SLMs' memory for given labels should be accordingly enhanced rather than setting a fixed value.

Specifically, we set a threshold $\tau(t; x)$ for each sample x: $\tau(t; x) = \lambda p(t; x) + (1 - \lambda)\tau(t - 1; x)$, $\tau(0) = 0$, where $p(t; x) = max(p(k; x))$, $p(t; x)$ is the maximum confidence of current epoch t, λ is a hyperparameter controlling threshold stability. The idea of $\tau(t; x)$ is that the threshold for determining whether a model memorizes x should increase accordingly with the increase of historical confidence. But, the confidence of a single epoch may be unstable, especially in early training epochs. So, we use the momentum maximum confidence of each sample, computed based on all previous epochs, $\tau(t; x)$ as the dynamic threshold.

Class-wise Threshold. Since our empirical find-



(a) Overview of our methods. (b) MEM.

Figure 2: (a) The overview of DeCo. DeCo first employs the MEM to obtain diverse confidence for text classification. Based on these confidences, a dynamic threshold is introduced by considering whether the sample is simultaneously remembered, which is used to select certainty set E and uncertainty set U from the noisy dataset. Then, a class-wise threshold is used to finely group these two sets into the clean set C, hard set H, and noisy set O. Meanwhile, the LLM serves as an active annotator imbuing its knowledge: (1) construct high-quality demonstrations by selecting clean samples from C; (2) query noisy samples in O and generate annotation. Finally, we use different loss functions to learn three subsets. (b) The framework of MEM. MEM consists of SLM and a set of expert layers.

ings (see Appendix A) show that loss distributions of head and tail classes are different, we argue that the loss values of samples with different observed labels may not be comparable. So, it would be more appropriate to select samples class by class with the same observed labels rather than setting a global threshold. Specifically, the training data $\mathcal{D}$ is split to K set according to labels, *i.e.*, $\mathcal{D}_k = \{(x_i, y_i) \in \mathcal{D} | y_i = k\}$. For the k-th set $\mathcal{D}_k$, we compute the loss $\mathcal{L}_i$ for each sample of $\mathcal{D}_k$ and pick top $\phi(k) = min(\lceil \frac{N}{K} \times \mathcal{R} \rceil, N_k)$ small-loss samples, where $\mathcal{R}$ is the filter rate that is identical for all classes. Compared to the original small-loss selection, DeCo set the class-wise threshold $\phi(k)$ and pick $\phi(k)$ samples with the smallest loss from each class k to constitute a set $\mathcal{D}_{\phi(k)}$. Finally, the integral set $\mathcal{D}_\phi$ is merged as: $\mathcal{D}_\phi = \cup_{k=1}^{K} \mathcal{D}_{\phi(k)}$. The samples in $\mathcal{D}_\phi$ have a high clean probability.

Overall, the proposed threshold strategy can better divide the noisy dataset, and select clean samples from all classes without ignoring tail classes (see Figure 14), thereby mitigating selection bias.

4.1.3 Selection

DeCo combines dynamic thresholds $\tau(t)$ with sets $\mathcal{D}_\phi$ segmented based on class-wise thresholds $\phi(k)$ to conduct our debiased sample selection.

Based on the diverse confidence of multiple expert layers in MEM, $\tau(t)$ is used to select two subsets from $\mathcal{D}$. Specifically, for a sample x_i with label y_i, the confidence from expert layers can be denoted as $\{p_{g_1}(y_i; x_i), \ldots, p_{g_m}(y_i; x_i)\}$. If these confidences are all higher than $\tau(t; x_i)$, x_i is put into the certainty set E:

$$E = \{(x_i, y_i) | p_{g_1}(y_i; x_i) > \tau_{g_1}(t; x_i)\} \cap \ldots \cap \{(x_i, y_i) | p_{g_m}(y_i; x_i) > \tau_{g_m}(t; x_i)\}. \quad (1)$$

In other words, if a sample can be simultaneously memorized by multiple experts with high confidence, it should be included in the certainty set E. Otherwise, if only one or more of the experts' confidence surpasses $\tau(t; x_i)$, the sample is grouped into the uncertainty set U:

$$U = \{(x_i, y_i) | p_{g_1}(y_i; x_i) > \tau_{g_1}(t; x_i)\} \cup \ldots \cup \{(x_i, y_i) | p_{g_m}(y_i; x_i) > \tau_{g_m}(t; x_i)\} - E. \quad (2)$$

However, along with training, models eventually memorize the noisy label (Li et al., 2023), which means multiple experts may simultaneously memorize a sample with noisy labels during training. Hence, it is necessary to further distinguish whether the sample in the certainty set E may be clean or noisy. To achieve it, another threshold is introduced to distinguish clean samples from noisy samples.

For any sample in the set E, if its loss value falls within the lowest quantile of its respective class, it is then allocated to the clean set C: $C = \{(x_i, y_i) | x_i \in E \cap D_\phi\}$. Briefly, a sample that is concurrently recognized by multiple experts and has a low loss value is classified as a clean sample.

Similarly, it is valuable to further exploit the useful information from the uncertainty set. Qiao

et al. (2022) found that different network architectures remember noise samples with different rates. When one expert layer starts to memorize the noisy samples, its confidence for clean samples will decrease, but other expert layers may still maintain high confidence for these clean samples. Thus, we argue that some samples in the uncertainty set U could be clean and may potentially augment the generalization capabilities of our approach. These samples are categorized into the hard set H: $H = \{(x_i, y_i)|x_i \in U \cap D_\phi\}$. Following the segregation into C and H, the residual samples are collated into the noisy set O: $O = \{(x_i, y_i)|x_i \in D - (C \cup H)\}$.

4.2 Active Querying by LLMs

In this step, the LLMs are leveraged to generate new labels for noisy samples and the core challenge lies in constructing the ICL demonstration. Since the clean set C has a significantly low noisy ratio (see Table 4) and ICL has certain robustness for low label noise in demonstrations on classification tasks (Fei et al., 2023; Cheng et al., 2024), the samples in C can be selected to construct demonstration. Meanwhile, the noisy set O has a substantially high noise ratio that necessitates LLM query correction.

Demonstration Construction. Retrieval-augmented generation (RAG) that augments LLMs with the retrieval of relevant information has become increasingly popular (Gao et al., 2023b). Inspired by it, we first introduce class descriptions from Wikipedia for each class, which can provide class knowledge relevant to the task for enhancing LLMs, especially those tasks that require significant domain expertise. Next, for each class k, the w examples are introduced in prompts, w being the w-shot setting. Specifically, given a sample $x_i \in N$, we calculate the cosine similarity of the text feature between x_i and other samples x_j of each class ($x_j \in C_k, C_k = \{(x_i, y_i) \in C|y_i = k\}$). Then, the top-$w$ similar samples are sampled to form w-shot examples for each class k.

Querying. The proposed LLMs prompt consists of the following three components: (1) *Task description*, which describes the task. (2) *Demonstration*, which consists of the *class descriptions* of each class and its sample sequence from C. (3) *Input*, which is the sample from O to classify. By prompting LLMs to generate high-quality labels for noisy samples, the high noise rate of O is greatly reduced. The details are provided in Appendix J.

4.3 Training MEM

In this step, the selected three subsets are used to train MEM. Since the selection procedure depends on the expert layers' performance, it is necessary to update the parameters of the expert layer. Follow Wei et al. (2024), we randomly assign a layer from the set of $\{f, g_1, \ldots, g_m\}$ as the classifier layer $\widetilde{f}$ and the rest are expert layers. After sufficient training, each layer in MEM contains sterling performance on selection and prediction.

Learning From the Clean Set C. Cross-entropy loss is directly utilized (l_{ce}) on C for MEM

$$\mathcal{L}_C = -\frac{1}{N}\sum_{i=1}^{N_C} \log p_{\widetilde{f}}(y_i; x_i) \tag{3}$$

where N_C denotes the size of the clean set C, N denotes the size of the entire dataset.

Learning From the Noisy Set O. The number of noisy samples in O is significantly reduced following the generation of the labels by LLMs for samples in O. However, LLMs, even the powerful GPT-4, cannot generate accurate labels for every sample in O. Our solution comes from rethinking the way of learning from the O: *can we design an objective such that our model can be optimized with access to the O with a lower noise ratio?* To this end, we resort to a family of noise-robust loss functions (Ghosh et al., 2017; Gao et al., 2023a). In LNL, the demonstration of a loss function's robustness ($\mathcal{L}_{robust}$) is achieved by showing that the estimated loss on datasets with most of the samples correctly labeled ($\mathcal{D}_{noisy}$) is equivalent to the loss calculated using datasets with total clean labels ($\mathcal{D}_{clean}$), *i.e.*, the noise-robust loss functions have the below noise-tolerant property:

$$\arg\min_\theta \mathcal{L}_{robust}(\widetilde{f}_\theta, \mathcal{D}_{clean}) = \arg\min_\theta \mathcal{L}_{robust}(\widetilde{f}_\theta, \mathcal{D}_{noisy}), \tag{4}$$

where θ is the parameter of $\widetilde{f}$. This property also applies under our label noise scenarios (proofs in Appendix H), which inspires us to learn from O by a noise-tolerant function. Specifically, we consider the reversed cross-entropy loss for samples in O:

$$\mathcal{L}_O = -\frac{1}{N}\sum_{i=1}^{N_O}\sum_{k=1}^{K}(p_{\widetilde{f}}(k; x_i)\log q(k|x_i)), \tag{5}$$

where $q(k|\boldsymbol{x})$ is the ground-truth distribution over labels, N_O denotes the size of O.

Remark. The Appendix I provides proof that the reversed cross-entropy loss is noise tolerant.

Learning From the Hard Set H. Due to the sample in H cannot be memorized by multi-

Dataset	Trec				AGNews (IF:1)						AGNews (IF:10)					
Method(↓) / Noise(→)	20%**S**	40%**S**	20%**A**	40%**A**	20%**S**	40%**S**	20%**A**	40%**A**	20%**I**	40%**I**	20%**S**	40%**S**	20%**A**	40%**A**	20%**I**	40%**I**
BERT	94.64	87.45	93.60	85.72	90.68	84.43	90.27	84.30	88.24	85.72	85.97	69.81	83.79	65.08	88.78	73.37
Co-Teaching	95.08	89.30	94.88	87.16	92.03	88.41	92.12	89.38	89.53	88.72	89.51	86.55	89.34	84.01	88.99	82.89
CL	95.64	89.72	95.52	86.24	92.17	88.45	92.30	89.13	89.94	87.03	90.58	88.04	89.51	83.52	88.76	78.15
CR	95.15	89.74	94.87	87.77	92.11	88.52	91.52	89.60	89.12	88.06	89.98	88.07	89.96	79.06	88.40	77.81
NPC	95.10	88.58	94.48	87.25	91.15	87.74	90.87	88.77	88.33	87.07	89.49	86.09	89.68	85.21	89.33	77.69
SelfMix	95.20	89.80	95.16	89.00	91.37	89.28	91.21	87.80	88.32	87.45	87.56	86.83	87.55	86.56	86.09	77.78
DyGen	95.88	89.00	94.96	88.56	91.61	89.88	91.59	86.62	89.15	87.72	87.26	86.19	87.07	85.26	87.64	77.42
Supervised GT (0% Noise)	97.20				94.05						91.87					
ChatGPT (Zero-shot/10-shot)	61.60/72.00				82.92/84.23						82.92/84.23					
DeCo (Ours)	**97.08**	**96.56**	**97.16**	**95.32**	**93.98**	**93.25**	**93.55**	**93.27**	**93.60**	**92.75**	**91.44**	**90.94**	**91.52**	**90.84**	**91.51**	**90.33**

Table 1: Performance (accuracy %) comparison of DeCo with other LNL basslines on synthetic noise datasets. Moreover, we also compare DeCo with the zero-shot/10-shot and supervised counterparts on the test dataset. Supervised GT refers to BERT trained on ground truth data. **Bold** means the best score for each dataset.

Method	**ChemProt**	**TREC**	**SEMEVAL**
Noise Ratio	22.88%	38.56%	16.00%
Base	$64.84_{\pm 0.28}$	$67.33_{\pm 0.83}$	$71.44_{\pm 0.10}$
Co-Teaching	$65.98_{\pm 0.63}$	$66.61_{\pm 0.35}$	$72.07_{\pm 0.76}$
CL	$65.95_{\pm 0.28}$	$71.16_{\pm 0.61}$	$73.63_{\pm 0.58}$
CR	$65.53_{\pm 0.22}$	$68.33_{\pm 0.31}$	$71.11_{\pm 1.07}$
NPC	$65.15_{\pm 0.51}$	$70.44_{\pm 0.39}$	$72.17_{\pm 0.17}$
SelfMix	$65.44_{\pm 0.55}$	$69.96_{\pm 2.16}$	$74.24_{\pm 3.01}$
DyGen	$69.07_{\pm 0.38}$	$72.39_{\pm 0.82}$	$73.17_{\pm 0.29}$
LAFT†	-	72.34	73.56
Ours	**70.52**$_{\pm 0.87}$	**78.80**$_{\pm 1.25}$	**82.63**$_{\pm 0.27}$

Table 2: Main results on real-world noise datasets. For LAFT†, we directly report the results of their versions.

Modules					**Trec**			
MEM	**DSS**	**C**	**N**	**H**	**20%S**	**40%S**	**20%A**	**40%A**
✔	✔	✔	✔	✔	**97.08**	**96.56**	**97.16**	**95.32**
✗	✔	✔	✔	✔	95.88	95.24	96.36	94.48
✔	✗	✔	✔	✔	94.40	90.32	95.20	89.44
✔	✔	✗	✔	✔	95.52	95.04	96.24	94.00
✔	✔	✔	✗	✔	95.72	93.00	95.12	92.64
✔	✔	✔	✔	✗	96.44	93.60	95.96	94.16

Table 3: Ablation study on the Trec dataset.

ple experts simultaneously, some noisy samples from tail classes will inevitably be selected as "clean". To minimize its negative effect, DeCo uses a confidence-based sample method to enhance the reliability of selected samples. Specifically, we randomly choose two samples (x_i, y_i), (x_j, y_j) and assign a larger coefficient for the sample whose prediction confidence is higher and a lower coefficient for the sample with lower prediction confidence. For example, if $max(p(k; x_j) \leq max(p(k; x_i)$, the mixed sample (e'_i, y'_i) can be defined as $e'_i = \lambda' e_i + (1 - \lambda')e_j$, $y'_i = \lambda' y_i + (1 - \lambda')y_j$, e_i =SLMs(x_i), e_j =SLMs(x_j), $\lambda \sim$ Beta(α, α), $\lambda' = \max(\lambda, 1 - \lambda)$. By adopting the proposed confidence-based sample, H is reconstructed as $\widetilde{H}$: $\widetilde{H} = \{(e'_i, y'_i) | (x_i, y_i), (x_j, y_j) \in H\}$. For $\widetilde{H}$, we compute the loss: $\mathcal{L}_H = -\frac{1}{N} \sum_{i=1}^{N_{\widetilde{H}}} \log p_{\widetilde{f}}(y'_i; e'_i)$, where $N_{\widetilde{H}}$ is the size of $\widetilde{H}$.

Finally, the overall training objective $\mathcal{L}$ can be calculated by: $\mathcal{L} = \mathcal{L}_C + \mathcal{L}_O + \mathcal{L}_H$. Like existing methods, DeCo first warm-up MEM for 2 epochs, then performs sample selection for rest epochs.

5 Experiments

5.1 Datasets and Baselines Setup

Synthetic datasets. To construct the class-imbalanced versions of **AGNews** (Gulli, 2005), **MR** (Pang and Lee, 2005), and **20ng** (Lang, 1995), we take an exponential function $n_k = n_0 \mu^k$ to reduce the number of samples per class, where n_k is the sample number of class k ($n_0 \geq \cdots \geq n_k$) and $\mu \in (0, 1]$. Generally, the class imbalance factor (IF) is defined as $\frac{n_0}{n_k}$ to measure the imbalance degree, which is chosen from $\{1, 10, 50\}$ in this paper. Then, we also select some inherently imbalanced dataset: **Trec** (Li and Roth, 2002), **R8** (Lewis, 1992). Three different types of synthetic label noise are injected into these datasets following the setups of existing LNL works (Qiao et al., 2022; Zhu et al., 2022): (1) **Symmetric Noise (S)** flips labels uniformly to other classes (van Rooyen et al., 2015) (2) **Asymmetric Noise (A)** flips labels to other similar classes. (Chen et al., 2019; Zhu et al., 2022) (3) **Instance-dependent Noise (I)** flips

label with a probability proportional to the features of the sample (Algan and Ulusoy, 2020).

Real-world datasets. Further, we conduct experiments on inherently imbalanced datasets with **real-world noise**: ChemProt (Krallinger et al., 2017), SemEval (Zhou et al., 2020), and TREC (Awasthi et al., 2020). More details are in Appendix B.

Baselines. We compare DeCo to the most popular baselines for LNL. Specifically, we compare to: (1) **Base** that performs fine-tuning with cross-entropy loss (Devlin et al., 2019); (2) *Sample Selection methods*, including **Co-Teaching** (Han et al., 2018), **CR** (Zhou and Chen, 2021), **NPC** (Bae et al., 2022), **SelfMix** (Qiao et al., 2022), **LAFT** (Wang et al., 2023), **FreeAL** (Xiao et al., 2023); (3) *Others*, **CL** (Northcutt et al., 2021), **DyGen** (Zhuang et al., 2023). See Appendix C for more details.

The implementation details are in Appendix E.

5.2 Main Results

Results on Trec (synthetic noisy and inherently class-imbalanced dataset) and AGNews (synthetic noisy and synthetic class-imbalanced dataset) are shown in Table 1. Table 2 shows the results for real-world datasets. (See Appendix D for more results). These results illustrate: (1) When the dataset contains only noisy labels (*i.e.*, the IF is 1), all methods achieve robust performance. Notably, our method achieves the best performance. (2) When the dataset is noisy and class-imbalanced, other sample selection methods exhibit significant performance degradation as the IF and the noise rate increase, whereas our method displays robustness and remains competitive. (3) DeCo outperforms all rivals by a notable margin on synthetic datasets and real-world datasets with different settings. (4) We also provide performances of the base model on the ground truth data (upper bounds) and ChatGPT (zero/10-shot). On simple datasets with fewer classes, ChatGPT performs better than certain baselines, yet DeCo still maintains a certain advantage. For upper bounds, the results of DeCo are closest to it compared to others. In short, the above observations clearly verify the effectiveness of DeCo.

5.3 Ablation Studies

In this section, we study the influence of each proposed component (see Table 3) of DeCo.

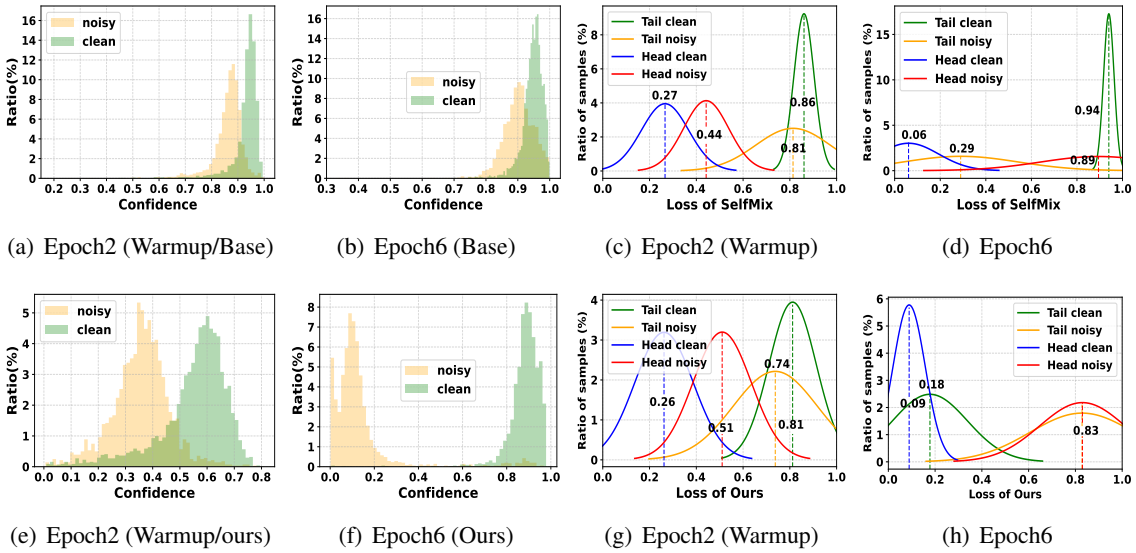
MEM. MEM independently conducts SLM learning and sample selection, which can alleviate training bias (accumulated error) caused by self-training. When MEM is removed, training and sample selection on BERT only with a classifier layer f, the performance of our method will decrease, which indicates that MEM is indeed crucial and contributes to mitigating the impact of noise.

Debiased Sample Selection. Based on the above analysis, the current sample selection tends to cause selection bias in class-imbalanced noisy datasets. Our debiased sample selection (DSS) ensures that the tail classes are not neglected and selected into a clean set during training. Accordingly, the selection bias issue is effectively alleviated. If DSS is removed, the performance degradation is the most significant, which proves this module contributes the most to performance improvement.

The Divided Subsets. Since directly learning from noisy data can significantly degrade the performance of the model, we first divide the noisy data into C, O, and H, then learn from them separately. As shown in Table 3, removing each subset will lead to a decrease in the results. Thus, these subsets are all important for improving the performance of DeCo. More ablations see Appendix F.

5.4 Analysis

Capability of debiased sample selection. For clean and noisy samples, we visualize their confidence distributions of the base model (Figure 3(a-b)) and our model (Figure 3(e-f)) on R8 under 40% asymmetric noise. During training, the confidence generated by our model is getting more polarized while the base model has already overfitted the noisy labels. Although SelfMix can handle label noise, it tends to fail in class-imbalanced datasets. For the head class, the loss values of clean samples are lower, while the loss values of noisy samples are higher. However, the loss distribution is just the opposite for the tail class due to under-learning tail class samples (Figure 3(c-d), Figure 12). This phenomenon avoids SelfMix selecting clean samples of tail classes (selection bias)(Figure 4). Moreover, due to training bias, SelfMix cannot improve itself during training(Figure 14). For DeCo, MEM and DSS are proposed to mitigate training and selection bias to perform debiased sample selection. Benefit from it, DeCo ensures that the tail class is not neglected during training(Figure 4, Figure 14), which encourages the loss distributions of clean and noisy samples to gradually become consistent across both head and tail classes (Figure 3(g-h), Figure 13). In this case, DeCo can better isolate noisy and clean samples based on the small-loss criterion, which can help MEM better perform training on these



(a) Epoch2 (Warmup/Base) (b) Epoch6 (Base) (c) Epoch2 (Warmup) (d) Epoch6

(e) Epoch2 (Warmup/ours) (f) Epoch6 (Ours) (g) Epoch2 (Warmup) (h) Epoch6

Figure 3: The confidence distributions of Base (a-b) and Ours (e-f) on R8 under 40% asymmetric noise in different stages. The loss distributions of SelfMix (c-d) and Ours (g-h) on R8 under 40% asymmetric noise in different stages.

Dataset	Subsets	epoch3	epoch4	epoch5	epoch6
Trec	Clean set (C)	3524/3540	3530/3546	3577/3591	3587/3599
	Noisy set (N)	264/1108 ⇒ 929 /1108 (83.84%)	237/1093 ⇒ 1069/1093 (97.80%)	119/979⇒ 958/979 (97.85%)	108/969⇒ 965/969 (99.59%)
	Hard set (H)	596/704	617/733	688/782	689/804

Table 4: The data statistical distribution (the number of samples with right labels / the number of all samples) of different subsets on Trec under 20%A. The left part of ⇒ is the original distribution of the noisy set, and the right part of ⇒ is the distribution after querying LLMs. The value (%) represents the ratio of correct labels in the subset.

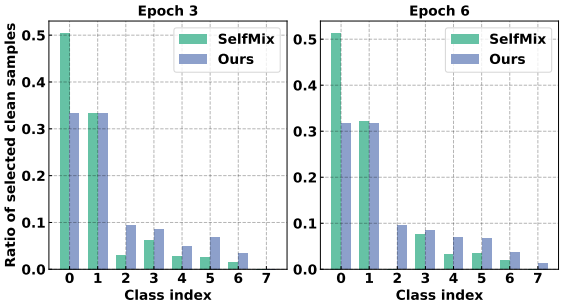


Figure 4: The class ratio distribution of clean samples selected by SelfMix and our method on R8 under 40%A. During training, SelfMix gradually ignored the samples in Class 2 and consistently failed to select clean samples from Class 7 to learn. Our method conducts debiased sample selection without neglecting any class.

selected debiased clean samples and avoid memorizing (overfitting) noisy samples.

Effect of Collaborative Learning. We present the statistical distribution of the different subsets during training. After warming up, we perform the debiased sample selection. From Table 4, we observe that the number of correctly labeled samples in O has significantly increased after querying LLMs, confirming the effectiveness of using LLMs for label denoising. During training, we notice the following trends: (1) For C and H, the number of correctly labeled samples in these subsets is gradually increasing; (2) For O, the ratio of correct labels provided by LLMs is also steadily rising. These observations display that collaborative learning enables SLMs and LLMs to mutually enhance their performance. More analysis see Appendix G.

6 Conclusion

In this paper, we focused on the challenge of learning with noisy and imbalanced datasets. To simultaneously address label noise and class imbalance, we propose a novel and effective framework called DeCo that introduces active learning to combine the SLM and LLM. Specifically, DeCo utilizes the SLM to construct a robust multi-expert model and introduces a dynamic class-wise threshold strategy to select clean and noisy samples for LLM. Then the LLM employs the clean samples to construct demonstrations and query noisy samples for re-annotation, the re-annotation results collaboratively enhancing SLM training for more precise selection. The effectiveness and superiority of our proposed method are verified on diverse noise types and both synthetic and real-world datasets.

Limitations

Our proposed DeCo is a collaborative framework that aims to handle label noise on class-imbalanced datasets for multi-class text classification. Despite its effectiveness, there is still much potential for improvement. First, the annotation quality of noisy samples largely hinges on the strong ability of LLMs. For some domains that are extremely challenging or eccentric, the commonly adopted GPT-3.5-Turbo (even GPT-4-0613) nowadays may fail to provide a qualified initial annotation, even if we introduce external knowledge class descriptions in the demonstrations. Our model is anticipated to be suitable for these circumstances with the advancement of more powerful LLMs across diverse domains.

Ethics Statement

We adhere to the Ethics of Official. This paper will not pose ethical problems or negative social consequences. The datasets used in our paper are all publicly available and are widely adopted by researchers to evaluate models.

References

Görkem Algan and Ilkay Ulusoy. 2020. Label noise types and their effects on deep learning. *CoRR*, abs/2003.10471.

Abhijeet Awasthi, Sabyasachi Ghosh, Rasna Goyal, and Sunita Sarawagi. 2020. Learning from rules generalizing labeled exemplars. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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 *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 1277–1297. PMLR.

Ruichu Cai, Zhenjie Zhang, Anthony K. H. Tung, Chenyun Dai, and Zhifeng Hao. 2014. A general framework of hierarchical clustering and its applications. *Inf. Sci.*, 272:29–48.

Duo Chai, Wei Wu, Qinghong Han, Fei Wu, and Jiwei Li. 2020. Description based text classification with reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1371–1382. PMLR.

Pengfei Chen, Benben Liao, Guangyong Chen, and Shengyu Zhang. 2019. Understanding and utilizing deep neural networks trained with noisy labels. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1062–1070. PMLR.

Chen Cheng, Xinzhi Yu, Haodong Wen, Jingsong Sun, Guanzhang Yue, Yihao Zhang, and Zeming Wei. 2024. Exploring the robustness of in-context learning with noisy labels. *CoRR*, abs/2404.18191.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. 2023. Mitigating label biases for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 14014–14031. Association for Computational Linguistics.

Hongfu Gao, Feipeng Zhang, Wenyu Jiang, Jun Shu, Feng Zheng, and Hongxin Wei. 2024. On the noise robustness of in-context learning for text generation. *Preprint*, arXiv:2405.17264.

Jiahui Gao, Renjie Pi, Yong Lin, Hang Xu, Jiacheng Ye, Zhiyong Wu, Weizhong Zhang, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. 2023a. Self-guided noise-free data generation for efficient zero-shot learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023b. Retrieval-augmented generation for large language models: A survey. *CoRR*, abs/2312.10997.

Aritra Ghosh, Himanshu Kumar, and P. S. Sastry. 2017. Robust loss functions under label noise for deep neural networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 1919–1925. AAAI Press.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd-workers for text-annotation tasks. *CoRR*, abs/2303.15056.

Antonio Gulli. 2005. The anatomy of a news search engine. In *Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005 - Special interest tracks and posters*, pages 880–881. ACM.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi

Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8536–8546.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-16, 2010*, pages 33–38. The Association for Computer Linguistics.

Yingsong Huang, Bing Bai, Shengwei Zhao, Kun Bai, and Fei Wang. 2022. Uncertainty-aware learning against label noise on imbalanced datasets. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 6960–6969. AAAI Press.

Shenwang Jiang, Jianan Li, Ying Wang, Bo Huang, Zhang Zhang, and Tingfa Xu. 2022. Delving into sample loss curve to embrace noisy and imbalanced data. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 7024–7032. AAAI Press.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jannik Kossen, Tom Rainforth, and Yarin Gal. 2023. In-context learning in large language models learns label relationships but is not conventional learning. *CoRR*, abs/2307.12375.

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*, volume 1, pages 141–146.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*, pages 331–339. Morgan Kaufmann.

David D. Lewis. 1992. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Copenhagen, Denmark, June 21-24, 1992*, pages 37–50. ACM.

Junnan Li, Richard Socher, and Steven C. H. Hoi. 2020. Dividemix: Learning with noisy labels as semi-supervised learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. 2017. Webvision database: Visual learning and understanding from web data. *CoRR*, abs/1708.02862.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *19th International Conference on Computational Linguistics, COLING 2002, Howard International House and Academia Sinica, Taipei, Taiwan, August 24 - September 1, 2002*.

Yifan Li, Hu Han, Shiguang Shan, and Xilin Chen. 2023. DISC: learning from noisy labels via dynamic instance-specific selection and correction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 24070–24079. IEEE.

Huafeng Liu, Mengmeng Sheng, Zeren Sun, Yazhou Yao, Xian-Sheng Hua, and Heng Tao Shen. 2024. Learning with imbalanced noisy data by preventing bias in sample selection. *IEEE Trans. Multim.*, 26:7426–7437.

Danqing Luo, Chen Zhang, Yan Zhang, and Haizhou Li. 2024. Crosstune: Black-box few-shot classification with label enhancement. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 4185–4197. ELRA and ICCL.

Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah M. Erfani, and James Bailey. 2020. Normalized loss functions for deep learning with noisy labels. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6543–6553. PMLR.

Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. 2021. Confident learning: Estimating uncertainty in dataset labels. *J. Artif. Intell. Res.*, 70:1373–1411.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *Cornell University - arXiv,Cornell University - arXiv*.

Dan Qiao, Chenchen Dai, Yuyang Ding, Juntao Li, Qiang Chen, Wenliang Chen, and Min Zhang. 2022. Selfmix: Robust learning against textual label noise with self-mixup training. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 960–970. International Committee on Computational Linguistics.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4331–4340. PMLR.

Lior Rokach. 2010. Ensemble-based classifiers. *Artificial Intelligence Review*, page 1–39.

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 *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 1917–1928.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1195–1204.

Brendan van Rooyen, Aditya Krishna Menon, and Robert C. Williamson. 2015. Learning with symmetric label noise: The importance of being unhinged. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 10–18.

Song Wang, Zhen Tan, Ruocheng Guo, and Jundong Li. 2023. Noise-robust fine-tuning of pretrained language models via external guidance. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 12528–12540. Association for Computational Linguistics.

Qi Wei, Lei Feng, Haobo Wang, and Bo An. 2024. Debiased sample selection for combating noisy labels. *CoRR*, abs/2401.13360.

Ruixuan Xiao, Yiwen Dong, Junbo Zhao, Runze Wu, Minmin Lin, Gang Chen, and Haobo Wang. 2023. Freeal: Towards human-free active learning in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 14520–14535. Association for Computational Linguistics.

Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. 2019. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6222–6233.

Yan Yan, Rómer Rosales, Glenn Fung, Subramanian Ramanathan, and Jennifer G. Dy. 2014. Learning from multiple annotators with varying expertise. *Mach. Learn.*, 95(3):291–327.

Taraneh Younesian, Zilong Zhao, Amirmasoud Ghiassi, Robert Birke, and Lydia Y. Chen. 2021. Qactor: Active learning on noisy labels. In *Asian Conference on Machine Learning, ACML 2021, 17-19 November 2021, Virtual Event*, volume 157 of *Proceedings of Machine Learning Research*, pages 548–563. PMLR.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12501–12512. PMLR.

Zhilu Zhang and Mert R. Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8792–8802.

Liyue Zhao, Gita Sukthankar, and Rahul Sukthankar. 2011. Incremental relabeling for active learning with noisy crowdsourced annotations. In *PASSAT/SocialCom 2011, Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference on and 2011 IEEE Third International Conference on Social Computing (SocialCom), Boston, MA, USA, 9-11 Oct., 2011*, pages 728–733. IEEE Computer Society.

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, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5381–5392. Association for Computational Linguistics.

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 *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2166–2176. ACM / IW3C2.

Dawei Zhu, Michael A. Hedderich, Fangzhou Zhai, David Ifeoluwa Adelani, and Dietrich Klakow. 2022. Is BERT robust to label noise? A study on learning with noisy labels in text classification. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP, Insights@ACL 2022, Dublin, Ireland, May 26, 2022*, pages 62–67. Association for Computational Linguistics.

Yuchen Zhuang, Yue Yu, Lingkai Kong, Xiang Chen, and Chao Zhang. 2023. Dygen: Learning from noisy labels via dynamics-enhanced generative modeling. *CoRR*, abs/2305.19395.

A Analysis for Current Sample Selection-Based Method

The traditional sample selection-based method (Han et al., 2018; Shu et al., 2019; Qiao et al., 2022) normally follows the memory effect, *i.e.*, deep learning models first memorize clean samples and then gradually fit noisy ones. For example, a recent study SelfMix (Qiao et al., 2022) has illuminated that during SLM training, the loss distributions of clean and noisy samples typically adhere to two Gaussian Distributions. Notably, clean samples tend to have a smaller mean loss value (the small-loss criterion). Then, they usually set a global fixed loss threshold to separate noisy datasets. While these methods, such as SelfMix, have shown promise in leveraging this insight to separate noisy datasets, our empirical experiments indicate that these global-fixed-loss-value-based methods do not perform well in some scenarios when noisy labels coexist with class imbalance.

Specifically, we generate the synthetic label noise (40% asymmetric and 40% symmetric) and inject it into the inherently imbalanced R8 dataset. Then, we found that samples from tail classes tend to be overlooked and have large loss values (see Figure 5). Therefore, the SelfMix fails to help separate the clean samples of the tail class from the noisy ones (Figure 7). We further empirically analyze why the popular global-fixed-loss-value-based methods, such as SelfMix, do not perform well in noisy scenarios when class imbalance exists. For this experiment, we observe the changes of R8 under 40% asymmetric label noise at a fine-grained level (Table 6 shows the distribution of classes on the R8 under 40% asymmetric label noise). Specifically, as shown in Figure 12, we observe that:

- The loss distribution of clean samples and noisy samples, whether in the head class or tail class, dynamically evolves during the training process;
- Both tail class samples and noisy samples exhibit large losses;
- The losses of some clean samples belonging to the tail are even larger than the losses of some noisy ones from the head class;
- For the head class, the clean samples tend to have a smaller loss value and the noisy samples tend to have a bigger loss value;
- For the tail class, the noisy samples tend to have a smaller loss value and the clean samples tend to have a bigger loss value.

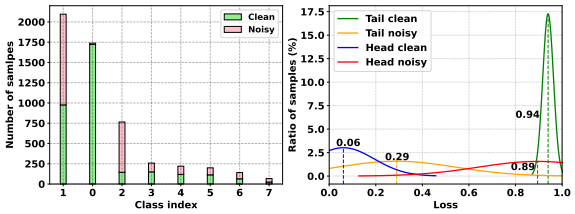


Figure 5: The class distribution (left) and the loss distribution (right) on imbalanced dataset R8 under 40% asymmetric noise. We can find that the losses of some clean samples belonging to tail classes are even larger than the losses of some noisy ones from head classes. Thus, existing small-loss-based sample selection methods that set a global fixed loss threshold to separate clean and noisy samples tend to fail.

Therefore, we think it is not ideal to handle all classes equally and set a global fixed loss threshold to separate noisy data.

Selection bias. Further, we observe that the SelfMix, conducted as *self-training*, will maintain or exacerbate the imbalanced ratio of selected clean subsets during the training process. As shown in Figure 14, the number of the selected samples decreases (see Index 2) or maintains (see Index 7) in the tail class. On the whole, the selected clean samples are always highly class-imbalanced (**selection bias**). Therefore, the model is always trained on these class-imbalanced biased sets, resulting in sub-optimal results. The intuitive reason is that the model's performance on tail classes hardly improves due to the limited number of available samples for training, which further degrades the effectiveness of selection criteria (*i.e.*, the small-loss criterion) on these classes.

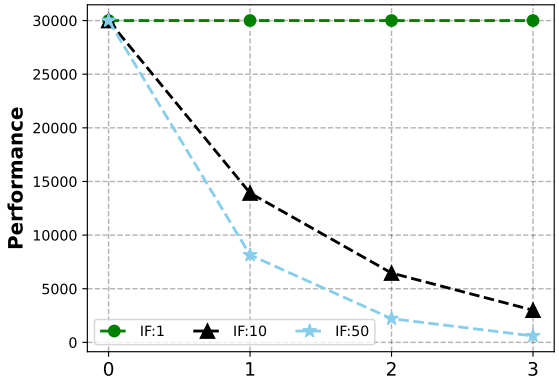


Figure 6: The number of samples belonging to each class (class index is 0, 1, 2, 3) in AGNews under various imbalance factor settings.

B Dataset Details

In this section, we introduce the details of the datasets used in our experiments. Specifically, we first select **20ng**, **AGNews** and **MR** to construct **synthetic imbalanced dataset** with different imbalance factors (*i.e.*, IF is 1, 10 and 50). Figure 6 presents the sample distribution of synthetic AGNews under different imbalance factors. Among them, 20ng and AGNews are news topic classification datasets, MR is a sentiment classification datasets. Then, we select two **inherently imbalanced datasets**: **Trec** and **R8**. The Trec dataset is a question classification with 6 classes. The R8 dataset consisted of news documents from the 8 most popular classes of the Reuters-21578 corpus.

Synthetic datasets. For these above 5 datasets, we manually inject three different types of noise (**Symmetric noise**, **Asymmetric noise** and **Instance-dependent nois**) into them for evaluation in our experiments. we explain the details of synthetic noise generation processes in section B.1.

Real-world datasets. To further verify the effectiveness of our method in practical scenarios, we conduct experiment on inherently imbalanced datasets with **real-world noise**: **ChemProt**, **SemEval**, and **TREC**. Among them, ChemProt is a chemical-protein interaction dataset with 10 classes, SEMEVAL is a relation extraction dataset with 9 classes, and TREC is a question classification with 6 classes. We follow the work DyGen (Zhuang et al., 2023) for these datasets to obtain real-world noise.

Table 5 introduces detailed statistics about all datasets used in our experiments.

#Dataset	#Class	#Training	#Validation	#Test
Trec	6	4952	500	500
R8	8	6674	500	500
20ng	20	9051	7527	2263
AGNews	4	112400	7600	7600
MR	2	8662	1000	1000
TREC	6	4965	500	500
SemEval	9	1749	692	200
ChemProt	10	12861	1607	1607

Table 5: The detailed statistics of all datasets used in our experiments.

B.1 Synthetic Noise Generation

In this part, we explain the details of synthetic noise generation processes.

Asymmetric noise (Asym) Asymmetric noise attempts to simulate the incorrect classification of classes. Modeling such noise can be achieved by flipping the labels of the samples according to a pre-defined noise level $\varepsilon \in [0, 1)$ (Zhu et al., 2022):

$$p_{flip}(\hat{y} = j|y = i) = \begin{cases} 1 - \varepsilon, & i = j \\ \varepsilon, & i \neq j \end{cases}.$$

Due to these noise generation processes are feature independent (*i.e.* $p(\cdot|y = i, x) = p(\cdot|y = i)$), we describe them by an asymmetric noise transition matrix, which can be used to generate noisy labels.

Symmetric noise (Sym) Modeling such noise can be achieved by uniformly flipping the labels of the samples to other classes according to a pre-defined noise level $\varepsilon \in [0, 1)$ (van Rooyen et al., 2015):

$$p_{flip}(\hat{y} = j|y = i) = \begin{cases} 1 - \varepsilon, & i = j \\ \frac{\varepsilon}{\mathcal{K}-1}, & i \neq j \end{cases},$$

where $\mathcal{K}$ is the number of classes.

Instance-dependent noise (IDN) We follow the noise generation process in existing literature (Bae et al., 2022; Qiao et al., 2022) for IDN generation in our experiments. The **Trec** dataset comprises only 5452 training samples. Consequently, when considering a high noise ratio, there's a possibility that the count of clean samples might be lower than that of generated noisy samples within the long-tailed class. This circumstance renders the classification task meaningless. As a result, we exclusively generate IDN on the 20ng, AGNews, MR, and R8 datasets. The detailed algorithm of IDN noisy label generation is summarized in Algorithm 1.

C Baselines Details

In this section, we provide the details of the baselines used in our experiments.

- **Base** (Devlin et al., 2019). We train the BERT (base model) model fine-tuned only with standard cross-entropy loss without noise-handling;
- **Co-Teaching** (Han et al., 2018) makes usage of the divergence in the learning errors of two distinct models to remove the influence of label noises. The two models screen out noisy samples for each other, combining their advantages in counteracting noisy labels;
- **CL** (Northcutt et al., 2021) employ confidence learning to quantify ontological class overlap and moderately increase model accuracy by cleaning data prior to training;

Algorithm 1 Instance Dependent Noise Generation

Require: Clean samples $(x_i, y_i)_{i=1}^n$, $y_i \in [1, K]$; Noisy ratio τ;

Train an BiLSTM classifier f;
Get output from an BiLSTM classifier $f_{x_i} \in \mathbb{R}^K$ for all $i = 1, \ldots, n$;
Set $N_{noisy} = 0$;
while $N_{noisy} < n \times \tau$ **do**
    Randomly choose a sample x_i, $argmax(softmax(f_{x_i})) \neq y_i$;
    Set its noisy label $\bar{y_i} = argmax(softmax(f_{x_i}))$;
    $N_{noisy} = N_{noisy} + 1$;
end while

Ensure: Noise samples $(x_i, \bar{y_i})_{i=1}^n$;

- **CR** (Zhou and Chen, 2021), also trains multiple models with a regularization strategy based on a soft target.
- **NPC** (Bae et al., 2022) utilizes a generative model to estimate the transition matrix from noisy predictions to the ground-truth labels of samples and uses it to correct noisy labels.
- **SelfMix** (Qiao et al., 2022) separates samples via GMM and leverages semi-supervised learning to handle label noise.
- **DyGen** (Zhuang et al., 2023) uses the variational auto-encoding framework to infer the posterior distributions of true labels from noisy labels to improve noisy label predictions. Although DyGen also considers dynamic training, it uses dynamic patterns in the embedding space, which is different from ours.
- **LAFT** (Wang et al., 2023) also segregates all training samples into different subsets by generating confidences for each sample of training datasets, which is a way that introduces the external guidance from LLMs. Although the segregation method based on confidence is similar to ours, LAFT ignores the inaccurate of LLM-generated confidences. Compared to LAFT, our methods only utilize the LLM on one subset, which can reduce the cost of LLM expenses. Then, we apply the noise-robust loss functions on LLM-generated labels, which can avoid the additional biases introduced by inaccurate results from LLMs. So our method is more efficient and effective than LAFT. In our experimental results, we directly refer to the results reported by its paper. The comparison results in Table 22.
- **FreeAL** (Xiao et al., 2023) also integrates active learning, alongside an LLM and a small model. Specifically, FreeAL considers unsupervised classification tasks without human annotations. For FreeAL, its input is an unlabeled training dataset and need LLM to annotate all datasets at first. However, our method consider supervised classification tasks with noisy human annotations, the input is a labeled training dataset with noisy labels. We need to select the noisy samples from datasets, then use LLM to annotate these samples. Our task is more intricate than FreeAL, as it involves a process of noise filtering. Moreover, we only need LLM to annotate noisy samples rather than all samples, which greatly reduces costs. We compare the FreeAL with our methods in Section D.1.
- **ChatGPT**. In addition to utilizing ChatGPT's zero-shot capabilities, we also evaluated its few-shot potential (random sampling) in our experiments. For both ChatGPT and DeCo, we uniformly configured the setup to a 10-shot setting in our experiments to ensure a fair comparison.

For these baselines, we perform their public code (except **LAFT**) to implement them.

D More detailed Results

We report the detailed performance (accuracy with standard deviation %) on Trec (refer to Table 14), AGNews (refer to Table 15), AGNews with IF 10 (refer to Table 16), AGNews with IF 50 (refer to Table 17), R8 (refer to Table 18), MR (refer to Table 19), MR with IF 10 (refer to Table 20), MR with IF 50 (refer to Table 21), 20ng (refer to Table 22), 20ng with IF 10 (refer to Table 23), 20ng with IF 50 (refer to Table 24).

D.1 Comparison with FreeAL

To compare with FreeAL, we use LLM to annotate the datasets and treat the annotated datasets (a labeled training dataset with noisy labels) as the input of our method. By doing so, we can ensure that the inputs for both methods remain consistent. The noisy ratio denotes the proportion of incorrect labels among those generated by the LLM. As shown in table 13, we found that our method consistently outperforms FreeAL across most datasets, especially on datasets with multiple categories. While FreeAL demonstrates commendable performance on simple binary sentiment classification datasets (MR), our approach maintains its superiority in more complex datasets. This consistency in performance across various datasets highlights the robustness and adaptability of our method, making it a viable and effective choice for a wide array of classification tasks.

E Implementation Details

In this section, we introduce the implementation details for our experiments.

We use BERT as the text encoder of NEM. The classifier layer and expert layers of NEM are all implemented as the fully-connected layer and randomly initialized at the beginning, while both the encoder and the classifier will be updated via gradient descent during fine-tuning.

Follow previous work (Qiao et al., 2022; Zhuang et al., 2023; Wang et al., 2023; Liu et al., 2024; Wei et al., 2024), 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 DeCo and all baselines. We report the average performance as well as standard deviations using 5 random seeds. We implement our framework with Python 3.7, PyTorch 1.13, and HuggingFace, and train our framework on Nvidia RTX 3090 and Nvidia A100 GPU. In addition, we use Adam (Kingma and Ba, 2015) as an optimizer.

LLMs and Prompts We use GPT-3.5-Tubor-0613 API (e.g., ChatGPT), and run the generation 5 times with a temperature of 0.8 to produce different reasoning paths and predictions. Then we use majority voting to get the final prediction results.

F More Ablation Experiments

F.1 Effect of Dynamic Class-wise Selection

Due to the suboptimal performance of current sample selection methods when dealing with noisy learning in imbalanced datasets, we introduce a dynamic class-wise sample selection to better perform LNL in imbalanced datasets. Here, we conduct an ablation experiment (refer to Table 8) to verify the effectiveness of the dynamic class-wise selection by replacing it with current sample selection methods based on a global-fixed-loss value.

F.2 Effect of Robust Loss Function

We conduct an ablation experiment (refer to Table 9) to verify the effectiveness of reversed cross-entropy loss functions by replacing it with cross-entropy loss functions.

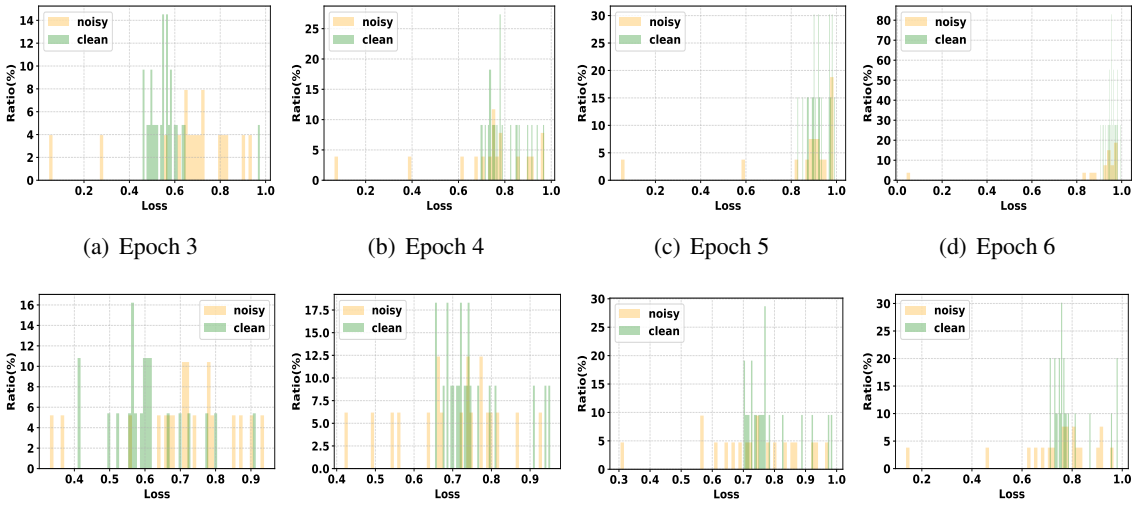
F.3 Effect of Prompt contents

In Table 10 and Table 11, we ablate the prompt contents in the default settings by: (a) removing the Task description (*r.m.* a); (b) removing the Class description (*r.m.* b); (c) removing the Demonstration (*r.m.* c); (d) replacing our example sampling with random example sampling (*r.p.* d). The results yield the subsequent observations: First, the task description is of less importance, indicating that ChatGPT is capable of understanding the task directly from the demonstration. Second, the demonstration is of critical importance to the performance of our DeCo. This is because they carry the necessary information for ChatGPT to understand the classification task. The class description in the demonstration is a crucial aspect, specifically in some tasks that require significant domain expertise. As shown in Table 10 for Trec and Table 11 for ChemProt (ChemProt is a chemical-protein interaction dataset with 10 classes), when we remove the class descriptions, the performance of our method experiences the most significant decline on the Chemprot dataset. Finally, our sampling strategy is important to the performance of our DeCo, especially under a high noise ratio.

G More Analysis

G.1 Capability of class-balanced learning.

We visualize the test accuracy in each class to demonstrate that DeCo achieves relatively balanced performance on varying categories while fulfilling greater generalization. We plot comparison results



(a) Epoch 3 (b) Epoch 4 (c) Epoch 5 (d) Epoch 6

(e) Epoch 3 (f) Epoch 4 (g) Epoch 5 (h) Epoch 6

Figure 7: Loss histogram of R8 under 40% asymmetric label noise (a-d), 40% symmetric label noise (e-h). Clean data and noisy data are marked by green and yellow respectively. We observe that the loss value can not separate the clean data from the noisy ones. Hence, the existing small-loss-based sample selection methods (such as SelfMix), which set a global fixed loss value to separate noisy data, are not applicable in our scenario (class-imbalanced noisy dataset).

Category	earn	acq	crude	trade	money-fx	interest	ship	grain
Index	0	1	2	3	4	5	6	7
Count	1738	2095	765	258	220	200	141	68

Table 6: The details statistics of R8 dataset under 40% asymmetric label noise.

Hyperparameter	**Trec**	**MR**	**20ng**	**AGNews**	**R8**	**TREC**	**ChemProt**	**SemEval**
λ	0.94	0.92	0.90	0.90	0.91	0.96	0.81	0.90
$\mathcal{R}$	0.8	0.8	0.7	0.8	0.90	0.85	0.95	0.70
m	2	5	3	2	3	4	3	2
Batch Size	32	32	32	32	32	32	32	32
learning rate	1e-5	1e-5	1e-5	1e-5	1e-5	1e-5	1e-5	1e-5

Table 7: Main hyperparameter settings of our models.

Dataset	**Trec**			
Loss function(↓) / Noise(→)	20%**S**	40%**S**	20%**A**	40%**A**
Current sample selection methods	95.25	92.00	96.08	91.20
Ours	**97.08**	**96.56**	**97.16**	**95.32**

Table 8: An ablation experiment for dynamic-class-wise sample selection. **Bold** means the best score.

Dataset	**Trec**			
Loss function(↓) / Noise(→)	20%**S**	40%**S**	20%**A**	40%**A**
cross-entropy loss	96.25	94.00	96.08	93.20
reversed cross-entropy loss	**97.08**	**96.56**	**97.16**	**95.32**

Table 9: Ablation study for loss functions on purified set O. **Bold** means the best score.

in Figure 8. First, compared with GT trained on clean samples, DeCo achieves almost unbiased prediction results. Second, compared with the existing method SelfMix, DeCo considers tail classes better. Therefore, DeCo performs better in these categories. The result demonstrates that our proposal has considerable potential for fulfilling class-balanced learning while tackling label noise.

Noise	default	*r.m.* a	*r.m.* b	*r.m.* c	*r.p.* d
20%A	**97.16**	96.85	96.22	96.05	96.55
40%A	**95.32**	95.12	93.86	93.78	94.02

Table 10: Prompt contents (Accuracy on Trec under 20% and 40% asymmetric label noise). The default settings include precisely the necessary information for prompting.

Noise	default	*r.m.* a	*r.m.* b	*r.m.* c	*r.p.* d
22.88%	**70.52**	68.22	66.10	67.05	67.85

Table 11: Prompt contents (Accuracy on ChemProt under 22.88% real-world noise). The default settings include precisely the necessary information for prompting.

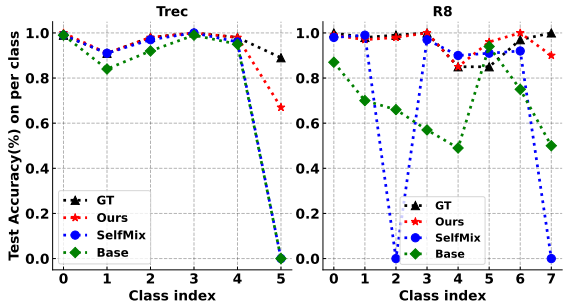


Figure 8: Comparisions of class-level prediction results on Trec with 20% A and R8 with 40% A.

G.2 Hyper-parameters Selection

There are three main hyper-parameters $\mathcal{R}$, λ, and m in our proposed method. We take $\mathcal{R}$ to control the proportion of selected clean samples per class, λ to control threshold stability, and m to control the number of expert layers. We conduct ablation studies to select optimal values for experiments, where $\mathcal{R} \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$, $\lambda \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$, $m \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$. We provide the model performance under different $\mathcal{R}$, λ, and m settings in Figure 10. From these results, we find that the appropriate hyper-parameters are not entirely consistent for different datasets. This is because the data distribution of datasets of different types is complex and inconsistent. Hence, we should comprehensively consider various situations and carefully tune hyper-parameters' values according to actual datasets. Based on our experience with parameter tuning during experiments, we have the following suggestions:

For $\mathcal{R}$ and λ. We found that choosing a large value (not the maximum value) of $\mathcal{R}$ and λ often leads to better model performance.

For m. For the number of expert layers m, we can see that as the value of m increases, the optimal testing accuracy first increases and then decreases. For the R8 or Trec dataset, when m = 3 or m = 5, the performance reaches the peak and fluctuates as m increases. This indicates that too large a value can harm model performance.

Table 7 shows the hyperparameter configurations for different datasets.

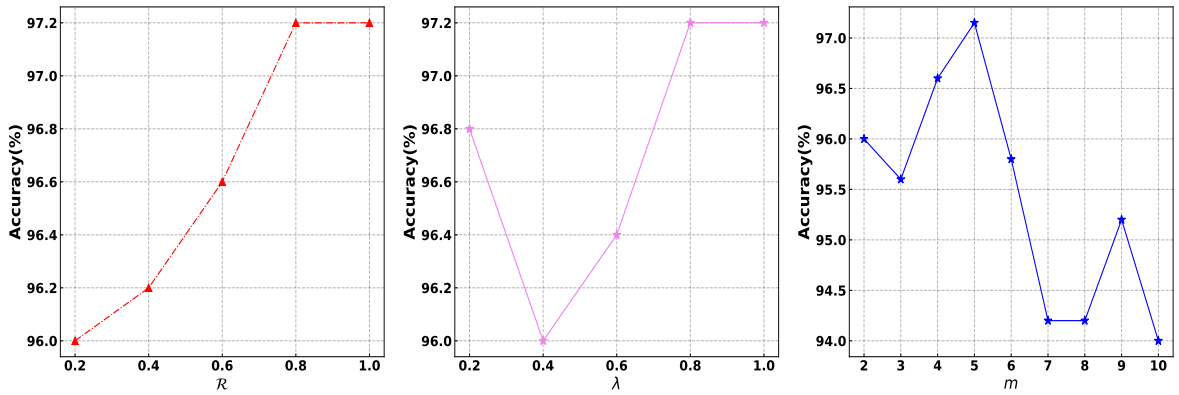
G.3 Effect of In-Context Examples.

We show the effect of different numbers w of in-context examples during the process of ICL on the Trec datasets under 20% asymmetric label noise and R8 datasets under 40% asymmetric label noise. Moreover, we conduct comparison experiments by replacing the ICL with much simpler approaches such as Knn. As shown in Figure 11, our method far exceeds than Knn over a wide range of w from 1 to 20, this further verifies the robustness of our method and we can simply adopt w = 10 for fair comparisons in our experiments.

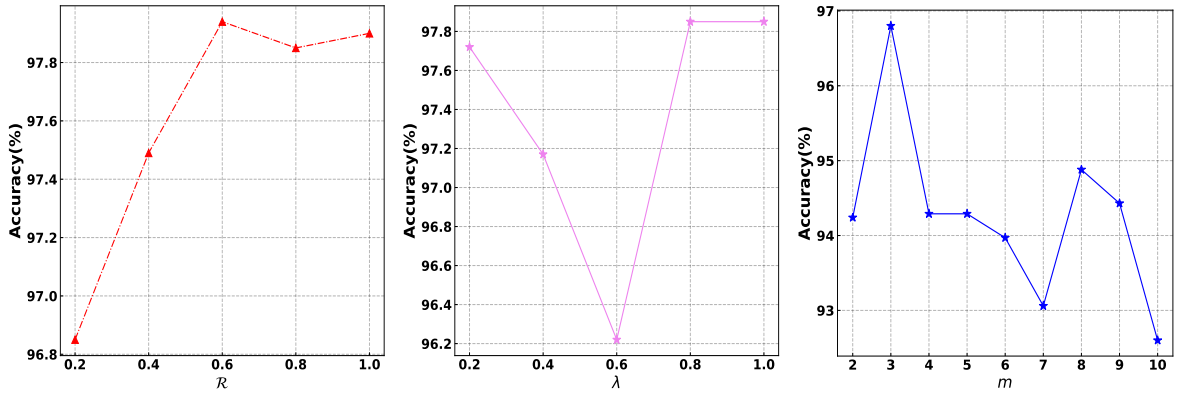
G.4 Experiments with other LLMs

For datasets containing noise, our method selects samples with noisy labels to form a subset O, and then uses an LLM to generate new labels for these noisy samples to create a new subset O. Through this method, although the LLM can significantly reduce the noise rate in the original subset O, the LLM itself also has hallucination issues; it cannot guarantee the generation of correct labels for all samples. To address this problem, we have theoretically proven that as long as the noise rate in subset O is below a certain threshold, we can directly use a robust loss function to learn from subset O, thus ignoring the influence of the noisy samples. Therefore, in our method, the noise rate in subset O is affected by the capabilities of the LLM used. This leads to the following two scenarios:

(1) When the noise ratio is relatively low (e.g., 20% or 40%), there aren't many noisy samples in subset O. For ChatGPT-3.5, even though its performance may not be as good as the more powerful LLM GPT-4, it can still reduce the noise rate in the original subset O to below the threshold. For GPT-4, it might reduce the noise rate in subset P even further. However, since we are using a robust loss function, whether ChatGPT-3.5 or GPT-4 is used, our method can conduct approximately noise-free



(a) Trec under 20% asymmetric label (b) Trec under 20% asymmetric label (c) Trec under 20% asymmetric label



(d) R8 under 40% asymmetric label (e) R8 under 40% asymmetric label (f) R8 under 40% asymmetric label

Figure 9: Hyper-parameter sensitivities of λ, $\mathcal{R}$ and m. Experiments are conducted on the Trec dataset under 20% asymmetric label noise and the R8 dataset under 40% asymmetric label noise.

training on subset O, and the final performance gap is not significant.

(2) Under conditions of extreme noise (>50%), there will be particularly many noisy samples in subset O. For ChatGPT-3.5, it may be a significant challenge to reduce the noise rate in the original subset O to below the threshold. However, for GPT-4, it can still generate correct labels relatively well and reduce the noise rate in subset O to below the threshold. So, in the case of extreme noise, the performance gap between ChatGPT-3.5 and GPT-4 becomes apparent, and our method achieves better results when using the more powerful LLM GPT-4.

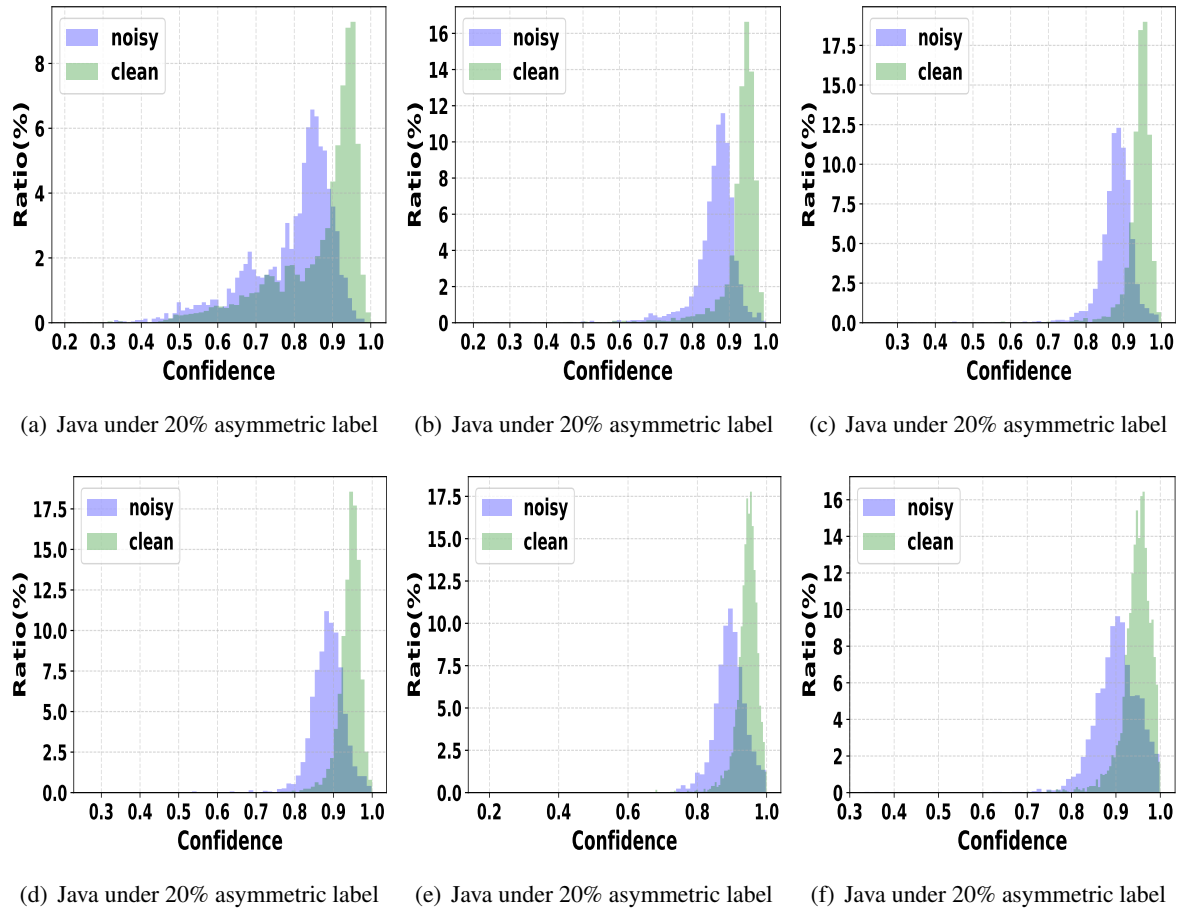
We further conducted preliminary experiments on Trec datasets under different noisy ratios. As shown in Table 12, we found that under circumstances of a smaller noise ratio, using ChatGPT-3.5 or a more powerful LLM such as ChatGPT-4 in our method does not show a significant performance difference. However, with a higher noise ratio, utilizing ChatGPT-4 in our method results in better outcomes.

Model	20%A	40%A	60%A
Ours (ChatGPT-3.5)	97.16	95.32	74.56
Ours (GPT-4)	97.20	95.45	77.32

Table 12: Performance (accuracy%) comparison on Trec datasets under 20% A, under 40% A, under 60% A.

G.5 Case Analyse for Cost Efficiency

In this paper, we incur costs by using ChatGPT. Thus, we provide a single case to calculate how many dollars need to be expensed by multiplying the total number of consumed tokens with the ChatGPT model price ($0.003 per thousand tokens). Specifically, for the R8 dataset with 40% Asymmetric label noise (40%A), all samples (Task + Class descriptions+ ICL examples + Inputs) to query ChatGPT were tokenized into 156523 tokens (please note, the R8 dataset has 8 categories, under which we set 10 samples to form ICL examples), and ChatGPT generated 9122 tokens, resulting in a total of 165645 tokens. Since we run each case



(a) Java under 20% asymmetric label (b) Java under 20% asymmetric label (c) Java under 20% asymmetric label

(d) Java under 20% asymmetric label (e) Java under 20% asymmetric label (f) Java under 20% asymmetric label

Figure 10: Hyper-parameter sensitivities of λ, $\mathcal{R}$ and m. Experiments are conducted on the Trec dataset under 20% asymmetric label noise and the R8 dataset under 40% asymmetric label noise.

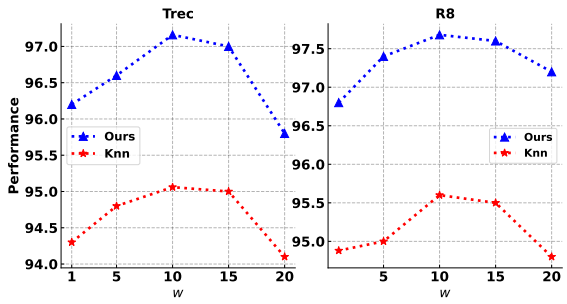


Figure 11: Comparisions of DeCo with KNN on Trec dataset under 20% asymmetric label noise and R8 dataset under 40% asymmetric label noise.

using 5 random seeds and report the average performance, the final tokens are 165645*5=828225 tokens, so we need to spend $2.484675.

G.6 Discussion about Human Annotations

In this section, we discuss the cost of human annotations and LLM annotations. Generally, human annotation tasks can be conducted by crowdworkers on platforms such as MTurk and trained annotators, such as research assistants. However, LLM annotations are generally more efficient than human annotations when dealing with large datasets. This is primarily because LLMs can generate annotations quickly and at scale, without the need for extensive human labor. Moreover, the per-annotation cost of ChatGPT is less than $0.003—about twenty times cheaper than MTurk.

H Theoretical Analysis

H.1 Risk Minimization problem for losses

Generally, for a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ with data size N, given any loss function $\mathcal{L}$ and classifier f_θ, we define the loss on $\mathcal{D}$:

$$\begin{aligned}\mathcal{L}(f_\theta, \mathcal{D}) &= \mathbb{E}_{\mathcal{D}}[\mathcal{L}(f_\theta(x), y)] \\ &= \mathbb{E}_{x,y}[\mathcal{L}(f_\theta(x), y)]. \end{aligned} \tag{6}$$

Under the risk minimization problem for losses, our object is to learn a classifier f, which is a global minimizer of $\mathcal{D}$ depending on the loss function $\mathcal{L}$.

That is to say, we want to obtain the optimal parameters θ^* of f with loss function $\mathcal{L}$ over dataset $\mathcal{D}$, *i.e.*, $\theta^* = \arg\min_\theta \mathcal{L}_{robust}(f_\theta, \mathcal{D})$.

H.2 Noise Robustness of Loss Functions

Let $\mathcal{D}_{clean} = \{(x_i, y_i)\}$ represent the clean training dataset, and $\mathcal{D}_{noisy} = \{(x_i, \hat{y}_i)\}$ represent the noisy training dataset with noise rate ε, where

$$\hat{y}_i = \begin{cases} y_i, & 1-\varepsilon \\ others, & \varepsilon \end{cases}.$$

Previous work (Ghosh et al., 2017; Zhang and Sabuncu, 2018; Xu et al., 2019; Gao et al., 2023a) on noise-robust loss functions has shown that the loss function satisfying following formula:

$$\sum_{k=1}^{\mathcal{K}} \ell_{\text{robust}}(f_\theta(x), k) = C, \forall x, f, \tag{7}$$

is a robust loss $\mathcal{L}_{robust}$, which has the below noise-tolerant property (Gao et al., 2023a):

$$\arg\min_\theta \mathcal{L}_{robust}(f_\theta, \mathcal{D}_{clean}) = \arg\min_\theta \mathcal{L}_{robust}(f_\theta, \mathcal{D}_{noisy}). \tag{8}$$

H.3 Proof For Noise-tolerant Property

We include the aforementioned work here to ensure comprehensiveness. More precisely, we consider three scenarios of label noise: asymmetric noise, symmetric noise, and instance-dependent noise as described in the following.

Symmetric Noise. In a multi-class classification task with $\mathcal{K}$ classes, given a loss function $\mathcal{L}_{robust}$ satisfying property 7. Then $\mathcal{L}_{robust}$ is noise-tolerant under symmetric label noise if noise rate $\varepsilon < 1 - \frac{1}{\mathcal{K}}$, the proof as follows:

$$\begin{aligned}
&\mathcal{L}_{robust}(f_\theta, \mathcal{D}_{noisy}) \\
&= \mathbb{E}_{x,\hat{y}}[\mathcal{L}_{robust}(f_\theta(x), \hat{y})] \\
&= \mathbb{E}_x \mathbb{E}_{y|x} \mathbb{E}_{\hat{y}|y,x}[\mathcal{L}_{robust}(f_\theta(x), \hat{y})] \\
&= \mathbb{E}_x \mathbb{E}_{y|x}[(1-\varepsilon)\mathcal{L}_{robust}(f_\theta(x), y) + \\
&\quad \frac{\varepsilon}{\mathcal{K}-1} \sum_{j \neq y} \mathcal{L}_{robust}(f_\theta(x), j)] \\
&= \mathbb{E}_x \mathbb{E}_{y|x}[(1 - \varepsilon + \frac{\varepsilon}{\mathcal{K}-1} - \frac{\varepsilon}{\mathcal{K}-1}) \\
&\quad \mathcal{L}_{robust}(f_\theta(x), y) + \frac{\varepsilon}{\mathcal{K}-1} \sum_{j \neq y} \mathcal{L}_{robust}(f_\theta(x), j)] \\
&= \mathbb{E}_{x,y}[(1 - \varepsilon + \frac{\varepsilon}{\mathcal{K}-1} - \frac{\varepsilon}{\mathcal{K}-1}) \\
&\quad \mathcal{L}_{robust}(f_\theta(x), y) + \frac{\varepsilon}{\mathcal{K}-1} \sum_{j \neq y} \mathcal{L}_{robust}(f_\theta(x), j)] \\
&= \mathbb{E}_{\boldsymbol{x},y}[\frac{\mathcal{K}-1-\mathcal{K}\varepsilon}{\mathcal{K}-1} \mathcal{L}_{robust}(f_\theta(x), y)] + \frac{\varepsilon C}{\mathcal{K}-1} \\
&= \frac{\mathcal{K}-1-\mathcal{K}\varepsilon}{\mathcal{K}-1} \mathcal{L}_{robust}(f_\theta, \mathcal{D}_{clean}) + \frac{\varepsilon C}{\mathcal{K}-1}
\end{aligned} \tag{9}$$

where C is a constant due to the property 7. Suppose θ^* is the optimal parameter of f over the clean dataset $\mathcal{D}_{clean}$, then for any θ:

$$\begin{aligned}
&\mathcal{L}_{robust}(f_{\theta^*}, \mathcal{D}_{noisy}) - \mathcal{L}_{robust}(f_\theta, \mathcal{D}_{noisy}) \\
&= \frac{\mathcal{K}-1-\mathcal{K}\varepsilon}{\mathcal{K}-1} \\
&\quad (\mathcal{L}_{robust}(f_{\theta^*}, \mathcal{D}_{clean}) - \mathcal{L}_{robust}(f_\theta, \mathcal{D}_{clean})) \\
&\leq 0.
\end{aligned} \tag{10}$$

Thus, when $\varepsilon < 1 - \frac{1}{\mathcal{K}}$, θ^* is also the optimal parameter of f over the noisy dataset $\mathcal{D}_{noisy}$.

Asymmetric Noise. For a loss function $\mathcal{L}_{robust}$ satisfying property 7. Then $\mathcal{L}_{robust}$ is noise-tolerant under asymmetric label noise if noise rate $\varepsilon < \frac{1}{2}$, the proof as follows:

$$
\begin{aligned}
&\mathcal{L}_{robust}(f_\theta, \mathcal{D}_{noisy}) \\
&= \mathbb{E}_{x,\hat{y}}[\mathcal{L}_{robust}(f_\theta(x), \hat{y})] \\
&= \mathbb{E}_x \mathbb{E}_{y|x} \mathbb{E}_{\hat{y}|y,x}[\mathcal{L}_{robust}(f_\theta(x), \hat{y})] \\
&= \mathbb{E}_x \mathbb{E}_{y|x}[(1-\varepsilon)\mathcal{L}_{robust}(f_\theta(x), y) + \\
&\quad \varepsilon \sum_{j \neq y} \mathcal{L}_{robust}(f_\theta(x), j)] \\
&= \mathbb{E}_x \mathbb{E}_{y|x}[(1 - \varepsilon + \varepsilon - \varepsilon \\
&\quad \mathcal{L}_{robust}(f_\theta(x), y) + \varepsilon \sum_{j \neq y} \mathcal{L}_{robust}(f_\theta(x), j)] \\
&= \mathbb{E}_{x,y}[(1 - \varepsilon + \varepsilon - \varepsilon) \\
&\quad \mathcal{L}_{robust}(f_\theta(x), y) + \varepsilon \sum_{j \neq y} \mathcal{L}_{robust}(f_\theta(x), j)] \\
&= \mathbb{E}_{\boldsymbol{x},y}[(1-2\varepsilon)\mathcal{L}_{robust}(f_\theta(x), y)] + \varepsilon C \\
&= (1-2\varepsilon)\mathcal{L}_{robust}(f_\theta, \mathcal{D}_{clean}) + \varepsilon C
\end{aligned} \tag{11}
$$

where C is a constant due to the property 7. Suppose θ^* is the optimal parameter of f over the clean dataset $\mathcal{D}_{clean}$, then for any θ:

$$
\begin{aligned}
&\mathcal{L}_{robust}(f_{\theta^*}, \mathcal{D}_{noisy}) - \mathcal{L}_{robust}(f_\theta, \mathcal{D}_{noisy}) \\
&= (1-2\varepsilon) \\
&\quad (\mathcal{L}_{robust}(f_\theta{}^*, \mathcal{D}_{clean}) - \mathcal{L}_{robust}(f_\theta, \mathcal{D}_{clean})) \\
&\leq 0.
\end{aligned} \tag{12}
$$

Thus, when $\varepsilon < \frac{1}{2}$, θ^* is also the optimal parameter of f over the noisy dataset $\mathcal{D}_{noisy}$.

Instance-dependent Noise. For a loss function $\mathcal{L}_{robust}$ satisfying property 7 and $0 \leq \mathcal{L}_{robust}(f_\theta(x), i) \leq \frac{C}{\mathcal{K}-1}, \forall i \in [\mathcal{K}]$. Suppose θ^* is the optimal parameter of f over the clean dataset $\mathcal{D}_{clean}$ and $\mathcal{L}_{robust}(f_{\theta^*}, \mathcal{D}_{clean}) = 0$. Then $\mathcal{L}_{robust}$ is noise-tolerant under instance-dependent noise label if noise rate $\varepsilon_j < 1 - \varepsilon_{ij}, \forall j \neq i, \forall i, j \in [\mathcal{K}]$, ε_{ij} represents the probability of class i mislabeled into class j. For instance-dependent noise, we have:

$$
\begin{aligned}
&\mathcal{L}_{robust}(f_\theta, \mathcal{D}_{noisy}) \\
&= \mathbb{E}_{x,\hat{y}}[\mathcal{L}_{robust}(f_\theta(x), \hat{y})] \\
&= \mathbb{E}_x \mathbb{E}_{y|x} \mathbb{E}_{\hat{y}|y,x}[\mathcal{L}_{robust}(f_\theta(x), \hat{y})] \\
&= \mathbb{E}_x \mathbb{E}_{y|x}[(1-\varepsilon_y)\mathcal{L}_{robust}(f_\theta(x), y) + \\
&\quad \sum_{j \neq y} \varepsilon_{yj} \mathcal{L}_{robust}(f_\theta(x), j)] \\
&= \mathbb{E}_x \mathbb{E}_{y|x}[(1-\varepsilon_y)(C - \sum_{j \neq y} \mathcal{L}_{robust}(f_\theta(x), j)]) \\
&\quad + \sum_{j \neq y} \varepsilon_{yj} \mathcal{L}_{robust}(f_\theta(x), j)] \\
&= \mathbb{E}_{x,y}[(1-\varepsilon_y)(C - \sum_{j \neq y} \mathcal{L}_{robust}(f_\theta(x), j)]) \\
&\quad + \sum_{j \neq y} \varepsilon_{yj} \mathcal{L}_{robust}(f_\theta(x), j)] \\
&= \mathbb{E}_{x,y}(C(1-\varepsilon_y) - (1-\varepsilon_y) \sum_{j \neq y} \mathcal{L}_{robust}(f_\theta(x), j) \\
&\quad + \sum_{j \neq y} \varepsilon_{yj} \mathcal{L}_{robust}(f_\theta(x), j)) \\
&= C\mathbb{E}_{x,y}(1-\varepsilon_y) \\
&\quad - \mathbb{E}_{x,y} \sum_{j \neq y} (1 - \varepsilon_y - \varepsilon_{yj}) \mathcal{L}_{robust}(f_\theta(x), j)
\end{aligned} \tag{13}
$$

where C is a constant due to the property 7. Suppose $\theta^\dagger$ is the optimal parameter of f over the noisy dataset $\mathcal{D}_{noisy}$ and $\theta^\dagger = \arg\min_\theta \mathcal{L}_{robust}(f_\theta, \mathcal{D}_{noisy})$, then:

$$
\begin{aligned}
&\mathcal{L}_{robust}(f_{\theta^\dagger}, \mathcal{D}_{noisy}) - \mathcal{L}_{robust}(f_{\theta^*}, \mathcal{D}_{noisy}) \\
&= \mathbb{E}_{x,y} \sum_{j \neq y} (1 - \varepsilon_y - \varepsilon_{yj})(\mathcal{L}_{robust}(f_{\theta^*}(x), j) \\
&\quad - \mathcal{L}_{robust}(f_{\theta^\dagger}(x), j)) \\
&\leq 0.
\end{aligned} \tag{14}
$$

Since we are given $\mathcal{L}_{robust}(f_{\theta^*}, \mathcal{D}_{clean}) = 0$, we have $\mathcal{L}_{robust}(f_{\theta^*}(x), y) = 0$. Considering the conditions stated before, we can get that $\mathcal{L}_{robust}(f_{\theta^*}(x), i) = \frac{C}{\mathcal{K}-1}, \forall i \neq y$. If we assume that $1 - \varepsilon_y - \varepsilon_{yj} > 0$, in order the Eq. 14 to hold, we must have $\mathcal{L}_{robust}(f_{\theta^\dagger}(x), i) = \frac{C}{\mathcal{K}-1}, \forall i \neq y$, which implies $\mathcal{L}_{robust}(f_{\theta^\dagger}(x), y) = 0$ due to the symmetric property of $\mathcal{L}_{robust}$. Thus, when $\varepsilon_j < 1 - \varepsilon_{ij}$, $\theta^\dagger$ is also the optimal parameter of f over the clean dataset $\mathcal{D}_{clean}$.

I Proof For Reversed Cross-entropy Loss

Theorem. *The reversed cross-entropy loss function satisfies formula 7 and has the noisy-tolerant property 8.*

Proof. For the input x and its label y, the predicted probability of x for each label $k \in [1, \mathcal{K}]$ can be represented as $p(k;x) = \frac{e^{f_\theta(k;x)}}{\sum_{k=1}^{\mathcal{K}} e^{f_\theta(k;x)}}$. $q(k|x)$ is the ground-truth distribution over labels, and $\sum_{k=1}^{\mathcal{K}} q(k|x) = 1$. If the ground-truth label is y, then $q(y|x) = 1$ and $q(k|x) = 0$ for all $k \neq y$. Based on it, we can obtain the reversed cross-entropy loss function $\mathcal{L}_{rce}$:

$$\begin{aligned}
\mathcal{L}_{rce}(f_\theta(x), y) &= -\sum_{k=1}^{\mathcal{K}} p(k;x) \log q(k|\boldsymbol{x}) \\
&= -p(y;x) \log q(y|\boldsymbol{x}) - \sum_{k \neq y}^{\mathcal{K}} p(k;x) \log q(k|\boldsymbol{x}) \\
&= -\sum_{k \neq y}^{\mathcal{K}} p(k;x) \log q(k|\boldsymbol{x}) \\
&= -\sum_{k \neq y}^{\mathcal{K}} p(k;x) \log(0). \qquad (15)
\end{aligned}$$

We approximate the $log(0)$ as a constant A, then $\sum_{k=1}^{\mathcal{K}} \mathcal{L}_{rce}(f_\theta(x), y) = -(\mathcal{K}-1)A$, which satisfies formula 7 and C=$-(\mathcal{K}-1)A$.

J Prompts Details

J.1 Prompts Structure

Our LLMs prompt consists of the following three components:

(1) ***Task description***, which describes the task. For different classification tasks, *e.g.*, question classification, sentiment classification, topic classification, *etc*, the descriptions are different.

Trec: You are a text classifier and your task is to classify a given sentence with the following classes (along with some examples):

MR: You are a sentiment classifier and your task is to classify a given text with the following classes (along with some examples). Your answer can be either positive or negative.

R8: You are a text classifier and your task is to classify a given text with the following classes (along with some examples):

TREC: You are a text classifier and your task is to classify a given sentence with the following categories (along with some examples). The true category must be one of these categories.

Agnews: You are a text classifier and your task is to classify a given sentence with the following classes (along with some examples). Your answer must be exactly one of ['World', 'Sports', 'Business', 'Science/Technology'].

Chemprot: You are a text classifier and your task is to classify a given sentence with the following classes (along with some examples). Your answer must be exactly one of ['Part of','Regulator','Upregulator','Downregulator','Agonist','Antagonist','Modulator', 'Cofactor','Substrate/Product','NOT'].

Semeval: You are a text classifier and your task is to classify a given sentence with the following classes (along with some examples). Your answer must be exactly one of ['Cause-Effect','Component-Whole', 'Content-Container','Entity-Destination', 'Entity-Origin','Instrument-Agency','Member-Collection','Message-Topic','Product-Producer'].

20ng: You are a text classifier and your task is to classify a given text with the following classes (along with some examples):

(2) ***Demonstration***, which consists of descriptions corresponding to all classes within the dataset, along with clean samples under each class (class+***class descriptions***+clean samples).

class descriptions. Although LLM can produce reasonably good labels for some tasks, there may be some tasks, such as those that require significant domain expertise, where LLM would perform poorly and cause our proposed approach may not work well. To tackle these questions and potential limitations, we drew inspiration from the recently popular RAG, introducing label descriptions for each class from external sources (the label description is either the definitions specified in the corresponding paper or from Wikipedia), enabling the large model to better comprehend the knowledge contained within the dataset. We provide the class

description of all datasets on Table 25, Table 26, Table 27, Table 28, Table 29, Table 30, Table 31, Table 32, Table 33.

For the demonstration sampling, we select the sample most similar to the test input. Specifically, given a sample $x_i \in N$, we can calculate the cosine similarity of the text feature between x_i and other samples x_j of each class ($x_j \in C_k, C_k = \{(x_i, y_i) \in C | y_i = k\}$). Then, we sample top-$w$ similar samples to form w-shot examples for each class k. Note that text features of datasets are computed and stored beforehand, allowing efficient sampling.

Finally, for each class k, we combine the description and w examples in the prompt.

(3) *Input*, which is the test text to classify.

J.2 Prompts Cases

We present the case (Table 34) of prompts designed for the LLMs.

Dataset	Trec	MR	20ng	AGNews	R8
Noisy Ratio (%)	38.4	21.31	30.67	17.08	14.34
FreeAL	93.50	**93.29**	80.63	92.69	98.02
Ours	**96.40**	92.20	**83.62**	**93.95**	**98.11**

Table 13: The comparison results with FreeAL.

Dataset	**Trec**			
Method(↓) / **Noise**(→)	20%**S**	40%**S**	20%**A**	40%**A**
BERT	94.64±$_{0.81}$	87.45±$_{0.74}$	93.60±$_{1.30}$	85.72±$_{0.97}$
Co-Teaching	95.08±$_{0.57}$	89.30±$_{1.50}$	94.88±$_{0.53}$	87.16±$_{0.36}$
CL	95.64±$_{0.15}$	89.72±$_{0.81}$	95.52±$_{0.24}$	86.24±$_{4.94}$
CR	95.15±$_{0.29}$	89.74±$_{0.17}$	94.87±$_{0.59}$	87.77±$_{0.33}$
NPC	95.10±$_{0.21}$	88.58±$_{0.22}$	94.48±$_{1.04}$	87.25±$_{0.86}$
SelfMix	95.20±$_{0.89}$	89.80±$_{1.15}$	95.16±$_{1.23}$	89.00±$_{0.86}$
DyGen	95.88±$_{0.32}$	89.00±$_{0.82}$	94.96±$_{0.57}$	88.56±$_{1.16}$
Supervised GT (0% Noise)	97.20±$_{0.20}$			
ChatGPT (Zero-shot)	61.60			
ChatGPT (10-shot)	72.00			
Ours	**97.08**±$_{0.30}$	**96.56**±$_{0.17}$	**97.16**±$_{0.09}$	**95.32**±$_{0.27}$

Table 14: The detailed results (accuracy with standard deviation %) on Trec datasets. **Bold** means the best score.

Dataset	**AGNews**					
Imbalance Factor	**1**					
Method(↓) / **Noise**(→)	20%**S**	40%**S**	20%**A**	40%**A**	20%**I**	40%**I**
BERT	90.68±$_{0.15}$	84.43±$_{1.36}$	90.27±$_{0.65}$	84.30±$_{1.90}$	88.24±$_{0.49}$	85.72±$_{0.97}$
Co-Teaching	92.03±$_{0.12}$	88.41±$_{0.26}$	92.12±$_{0.09}$	89.38±$_{1.11}$	89.53±$_{0.86}$	88.72±$_{0.14}$
CL	92.17±$_{0.11}$	88.45±$_{0.10}$	92.30±$_{0.11}$	89.13±$_{1.27}$	89.94±$_{0.15}$	87.03±$_{0.31}$
CR	92.11±$_{0.42}$	88.52±$_{0.88}$	91.52±$_{0.22}$	89.60±$_{0.54}$	89.12±$_{0.95}$	88.06±$_{1.01}$
NPC	91.15±$_{0.29}$	87.74±$_{0.07}$	90.87±$_{0.58}$	88.77±$_{0.33}$	88.33±$_{0.47}$	87.07±$_{0.49}$
SelfMix	91.37±$_{0.59}$	89.28±$_{0.90}$	91.21±$_{1.26}$	87.80±$_{0.40}$	88.32±$_{0.34}$	87.45±$_{0.74}$
DyGen	91.61±$_{0.20}$	89.88±$_{0.31}$	91.59±$_{0.25}$	86.62±$_{0.78}$	89.15±$_{0.24}$	87.72±$_{4.95}$
Supervised GT (0% Noise)	94.05±$_{0.14}$					
ChatGPT (Zero-shot)	82.92					
ChatGPT (10-shot)	84.23					
Ours	**93.98**±$_{0.10}$	**93.25**±$_{0.09}$	**93.55**±$_{0.11}$	**93.27**±$_{0.10}$	**93.60**±$_{0.05}$	**92.75**±$_{0.08}$

Table 15: The detailed results (accuracy with standard deviation %) on AGNews datasets. **Bold** means the best score.

Dataset	**AGNews**					
Imbalance Factor	**10**					
Method(↓) / **Noise**(→)	20%**S**	40%**S**	20%**A**	40%**A**	20%**I**	40%**I**
BERT	85.97±$_{0.69}$	69.81±$_{2.58}$	83.79±$_{0.57}$	65.08±$_{6.24}$	88.78±$_{0.14}$	73.37±$_{1.18}$
Co-Teaching	89.51±$_{0.77}$	86.55±$_{0.83}$	89.34±$_{0.64}$	84.01±$_{1.51}$	88.99±$_{0.26}$	82.89±$_{1.02}$
CL	90.58±$_{0.55}$	88.04±$_{0.60}$	89.51±$_{0.23}$	83.52±$_{0.92}$	88.76±$_{0.21}$	78.15±$_{1.04}$
CR	89.98±$_{0.43}$	88.07±$_{0.58}$	89.96±$_{0.53}$	79.06±$_{1.59}$	88.40±$_{0.24}$	77.81±$_{0.51}$
NPC	89.49±$_{0.32}$	86.09±$_{0.44}$	89.68±$_{0.61}$	85.21±$_{0.81}$	89.33±$_{0.15}$	77.69±$_{0.41}$
SelfMix	87.56±$_{0.50}$	86.83±$_{0.96}$	87.55 ±$_{1.77}$	86.56±$_{1.65}$	86.09±$_{0.78}$	77.78±$_{4.56}$
DyGen	87.26±$_{0.49}$	86.19±$_{1.12}$	87.07±$_{0.63}$	85.26±$_{0.22}$	87.64±$_{0.34}$	77.42±$_{1.07}$
Supervised GT (0% Noise)	91.87±$_{0.17}$					
ChatGPT (Zero-shot)	82.92					
ChatGPT (10-shot)	84.23					
Ours	**91.44**±$_{0.35}$	**90.94**±$_{0.10}$	**91.52**±$_{0.14}$	**90.84**±$_{0.11}$	**91.51**±$_{006}$	**90.33**±$_{0.10}$

Table 16: The detailed results (accuracy with standard deviation %) on AGNews datasets with IF 10. **Bold** means the best score.

Dataset	**AGNews**					
Imbalance Factor	**50**					
Method(↓) / **Noise**(→)	20%**S**	40%**S**	20%**A**	40%**A**	20%**I**	40%**I**
BERT	78.51±$_{0.83}$	69.11±$_{1.38}$	77.15±$_{1.42}$	61.11±$_{3.20}$	86.25±$_{0.59}$	65.15±$_{2.63}$
Co-Teaching	84.04±$_{1.48}$	80.38±$_{1.27}$	84.64±$_{1.76}$	78.22±$_{2.59}$	87.21±$_{0.93}$	65.86±$_{1.35}$
CL	85.51±$_{1.76}$	82.99±$_{1.09}$	83.24±$_{0.96}$	74.58±$_{3.22}$	87.53±$_{0.64}$	69.72±$_{2.44}$
CR	83.71±$_{0.12}$	82.12±$_{2.01}$	82.93±$_{1.58}$	73.75±$_{2.44}$	86.98±$_{0.74}$	66.74±$_{1.32}$
NPC	82.96±$_{1.11}$	80.18±$_{0.12}$	83.11±$_{1.61}$	78.91±$_{1.47}$	87.89±$_{0.47}$	69.14±$_{2.57}$
SelfMix	80.80±$_{0.62}$	78.87±$_{1.85}$	80.07±$_{0.06}$	78.18±$_{0.74}$	85.43±$_{0.88}$	69.41±$_{5.61}$
DyGen	84.20±$_{1.12}$	79.71±$_{2.34}$	83.95±$_{0.69}$	78.53±$_{0.01}$	86.31±$_{0.70}$	69.58±$_{5.28}$
Supervised GT (0% Noise)	88.96±$_{0.27}$					
ChatGPT (Zero-shot)	82.92					
ChatGPT (10-shot)	84.23					
Ours	**87.46**±$_{0.17}$	**87.05**±$_{0.20}$	**87.80**±$_{0.68}$	**87.24**±$_{0.46}$	**88.79**±$_{0.21}$	**87.80**±$_{0.15}$

Table 17: The detailed results (accuracy with standard deviation %) on AGNews datasets with IF 50. **Bold** means the best score.

Dataset	**R8**					
Method(↓) / **Noise**(→)	20%**S**	40%**S**	20%**A**	40%**A**	20%**I**	40%**I**
BERT	91.19±$_{3.02}$	81.59±$_{1.04}$	94.39±$_{0.87}$	80.16±$_{2.17}$	88.77±$_{0.82}$	71.65±$_{1.22}$
Co-Teaching	97.25±$_{0.51}$	93.82±$_{1.34}$	97.20±$_{0.52}$	94.63±$_{2.26}$	90.86±$_{1.34}$	80.00±$_{1.55}$
CL	95.85±$_{1.82}$	82.22±$_{6.73}$	96.91±$_{0.44}$	82.95±$_{3.49}$	91.49±$_{0.30}$	71.88±$_{0.56}$
CR	97.52±$_{0.25}$	84.12±$_{5.86}$	97.01±$_{0.31}$	91.65±$_{1.44}$	89.64±$_{1.17}$	71.59±$_{1.50}$
NPC	97.85±$_{0.24}$	93.19±$_{1.16}$	97.09±$_{0.15}$	93.06±$_{0.61}$	90.83±$_{0.93}$	72.86±$_{1.74}$
SelfMix	97.57±$_{0.28}$	94.62±$_{1.66}$	96.40±$_{2.00}$	92.17±$_{2.00}$	82.07±$_{2.92}$	77.32±$_{1.43}$
DyGen	92.93±$_{0.66}$	91.02±$_{0.81}$	94.77±$_{0.69}$	88.52±$_{2.08}$	92.69±$_{1.15}$	87.89±$_{1.57}$
Supervised GT (0% Noise)	98.39±$_{0.16}$					
ChatGPT (Zero-shot)	85.66					
ChatGPT (10-shot)	88.78					
Ours	**98.29**±$_{0.04}$	**97.65**±$_{0.08}$	**98.05**±$_{0.09}$	**97.68**±$_{0.09}$	**96.37**±$_{0.37}$	**93.28**±$_{0.18}$

Table 18: The detailed results (accuracy with standard deviation %) on R8 datasets. **Bold** means the best score.

Dataset	**MR**					
Imbalance Factor	**1**					
Method(↓) / **Noise**(→)	20%**S**	40%**S**	20%**A**	40%**A**	20%**I**	40%**I**
BERT	78.72±$_{1.35}$	64.74±$_{1.58}$	79.13±$_{1.22}$	63.97±$_{2.01}$	83.41±$_{2.14}$	76.73±$_{0.85}$
Co-Teaching	85.11±$_{0.24}$	81.95±$_{0.86}$	85.17±$_{0.52}$	81.59±$_{0.44}$	85.93±$_{0.78}$	78.61±$_{1.55}$
CL	84.90±$_{0.59}$	78.25±$_{0.94}$	85.64±$_{0.48}$	77.29±$_{1.82}$	85.52±$_{0.35}$	78.89±$_{0.69}$
CR	84.61±$_{0.63}$	73.73±$_{2.05}$	84.32±$_{0.81}$	73.20±$_{1.78}$	84.12±$_{1.62}$	79.15±$_{0.63}$
NPC	84.00±$_{0.82}$	72.91±$_{1.11}$	84.51±$_{0.46}$	71.29±$_{0.47}$	84.93±$_{0.91}$	78.79±$_{0.92}$
SelfMix	83.76±$_{0.70}$	77.69±$_{0.93}$	84.36±$_{1.11}$	78.36±$_{1.44}$	83.54±$_{4.51}$	76.05±$_{1.78}$
DyGen	84.69±$_{1.30}$	75.27±$_{3.72}$	84.44±$_{0.40}$	73.37±$_{1.74}$	83.89±$_{1.03}$	78.76±$_{3.33}$
Supervised GT (0% Noise)	87.33±$_{0.45}$					
ChatGPT (Zero-shot)	78.69					
ChatGPT (10-shot)	80.25					
Ours	**86.54**±$_{0.20}$	**82.74**±$_{0.09}$	**87.10**±$_{0.15}$	**82.94**±$_{0.22}$	**91.35**±$_{0.31}$	**90.68**±$_{0.36}$

Table 19: The detailed results (accuracy with standard deviation %) on MR datasets. **Bold** means the best score.

Dataset	**MR**					
Imbalance Factor	**10**					
Method(↓) / **Noise**(→)	20%**S**	40%**S**	20%**A**	40%**A**	20%**I**	40%**I**
BERT	74.20±$_{2.45}$	61.50±$_{5.88}$	77.00±$_{2.77}$	59.48±$_{4.37}$	82.09±$_{0.54}$	69.01±$_{0.87}$
Co-Teaching	76.21±$_{1.73}$	64.00±$_{3.61}$	78.38±$_{1.76}$	60.01±$_{7.00}$	73.23±$_{2.19}$	67.18±$_{2.31}$
CL	80.70±$_{2.08}$	70.63±$_{2.41}$	79.96±$_{2.74}$	66.07±$_{1.88}$	74.35±$_{1.96}$	66.55±$_{0.57}$
CR	79.27±$_{4.06}$	67.51±$_{4.79}$	77.05±$_{3.83}$	64.05±$_{2.90}$	70.63±$_{1.94}$	66.09±$_{0.71}$
NPC	75.81±$_{2.40}$	63.92±$_{3.20}$	75.53±$_{1.86}$	62.19±$_{2.71}$	71.40±$_{2.14}$	65.84±$_{0.40}$
SelfMix	76.68±$_{1.30}$	72.32±$_{1.49}$	74.79±$_{1.34}$	71.69±$_{1.28}$	75.77±$_{0.81}$	69.13±$_{1.96}$
DyGen	77.42±$_{1.05}$	74.63±$_{1.18}$	77.03±$_{1.21}$	75.01±$_{1.03}$	76.57±$_{1.75}$	70.63±$_{0.51}$
Supervised GT (0% Noise)	85.89±$_{1.00}$					
ChatGPT (Zero-shot)	78.69					
ChatGPT (10-shot)	80.25					
Ours	**84.47**±$_{0.35}$	**81.89**±$_{0.56}$	**84.57**±$_{0.36}$	**82.87**±$_{0.62}$	**85.01**±$_{0.93}$	**83.31**±$_{0.26}$

Table 20: The detailed results (accuracy with standard deviation %) on MR datasets. **Bold** means the best score.

Dataset	**MR**					
Imbalance Factor	**50**					
Method(↓) / **Noise**(→)	20%**S**	40%**S**	20%**A**	40%**A**	20%**I**	40%**I**
BERT	67.08±$_{1.07}$	57.20±$_{1.99}$	59.91±$_{1.88}$	55.71±$_{1.39}$	61.51±$_{1.22}$	60.07±$_{1.07}$
Co-Teaching	65.45±$_{2.79}$	60.73±$_{1.06}$	60.91±$_{1.14}$	64.00±$_{1.84}$	66.42±$_{3.77}$	64.92±$_{2.37}$
CL	66.01±$_{2.01}$	57.98±$_{1.43}$	54.55±$_{1.78}$	59.14±$_{2.37}$	69.21±$_{2.21}$	62.51±$_{0.72}$
CR	67.70±$_{2.04}$	59.01±$_{2.18}$	55.35±$_{2.08}$	58.68±$_{1.27}$	68.00±$_{1.43}$	59.16±$_{0.93}$
NPC	61.34±$_{1.61}$	57.36±$_{1.04}$	55.46±$_{2.19}$	57.24±$_{1.35}$	68.93±$_{2.86}$	60.07±$_{1.06}$
SelfMix	72.97±$_{1.28}$	67.51±$_{0.95}$	71.21±$_{0.29}$	68.88±$_{1.59}$	70.04±$_{1.55}$	58.73±$_{0.48}$
DyGen	73.41±$_{0.29}$	71.45±$_{0.47}$	73.25±$_{0.92}$	71.32±$_{0.28}$	72.53±$_{1.16}$	70.59±$_{0.26}$
Supervised GT (0% Noise)	80.19±$_{1.80}$					
ChatGPT (Zero-shot)	78.69					
ChatGPT (10-shot)	80.25					
Ours	**80.01**±$_{0.28}$	**78.95**±$_{0.19}$	**80.08**±$_{1.28}$	**78.62**±$_{0.67}$	**79.98**±$_{1.59}$	**77.61**±$_{0.39}$

Table 21: The detailed results (accuracy with standard deviation %) on MR datasets. **Bold** means the best score.

Dataset	**20ng**					
Imbalance Factor	**1**					
Method(↓) / **Noise**(→)	20%**S**	40%**S**	20%**A**	40%**A**	20%**I**	40%**I**
BERT	78.79±$_{2.51}$	66.55±$_{2.02}$	75.28±$_{2.10}$	60.15±$_{3.51}$	76.07±$_{0.96}$	66.32±$_{1.24}$
Co-Teaching	76.76±$_{1.43}$	68.42±$_{1.72}$	77.04±$_{0.85}$	58.95±$_{4.53}$	77.64±$_{0.67}$	66.43±$_{1.67}$
CL	80.48±$_{0.85}$	77.04±$_{0.90}$	80.50±$_{0.75}$	66.98±$_{3.93}$	79.35±$_{0.81}$	72.41±$_{1.44}$
CR	80.60±$_{0.42}$	70.61±$_{0.23}$	81.00±$_{0.11}$	68.33±$_{0.38}$	81.83±$_{0.42}$	71.74±$_{0.47}$
NPC	79.88±$_{0.29}$	68.74±$_{0.13}$	79.08±$_{0.46}$	66.80±$_{0.24}$	80.89±$_{0.39}$	70.45±$_{0.95}$
SelfMix	80.46±$_{1.28}$	72.50±$_{2.32}$	80.15±$_{1.98}$	72.50±$_{2.32}$	78.36±$_{0.41}$	74.40±$_{1.24}$
DyGen†	83.82±$_{0.04}$	79.56±$_{0.93}$	83.63±$_{0.23}$	81.98±$_{0.80}$	84.07±$_{0.17}$	81.54±$_{0.44}$
LAFT†	82.04±$_{0.11}$	76.93±$_{0.63}$	83.70	81.97	83.61	80.49
ChatGPT (Zero-shot)	69.33					
ChatGPT (10-shot)	70.03					
Supervised GT (0% Noise)	85.02±$_{0.41}$					
Ours	**84.01**±$_{0.02}$	**82.69**±$_{0.05}$	**84.69**±$_{0.04}$	**82.53**±$_{0.10}$	**84.30**±$_{0.09}$	**82.67**±$_{0.08}$

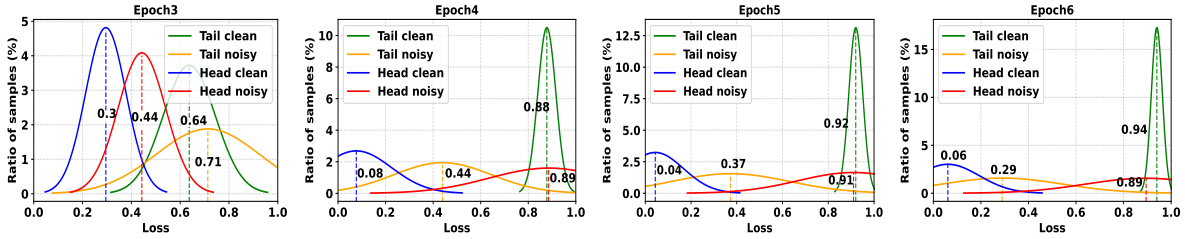
Table 22: The detailed results (accuracy %) on 20ng datasets. DyGen and LAFT also perform experiments on the 20ng dataset, so we directly report the results † of their versions. Since LAFT doesn't public their codes and report accuracy with standard deviation only under 20% Symmetric and 40% Symmetric, we can only report their incomplete results in our paper. **Bold** means the best score.

Dataset	**20ng**					
Imbalance Factor	**10**					
Method(↓) / **Noise**(→)	20%**S**	40%**S**	20%**A**	40%**A**	20%**I**	40%**I**
BERT	77.24±$_{0.49}$	71.87±$_{0.67}$	71.43±$_{1.18}$	58.37±$_{1.73}$	74.96±$_{0.30}$	67.50±$_{0.94}$
Co-Teaching	77.17±$_{0.57}$	72.86±$_{1.01}$	70.62±$_{0.69}$	49.06±$_{1.84}$	72.34±$_{0.87}$	61.46±$_{1.70}$
CL	72.57±$_{1.08}$	66.80±$_{1.98}$	72.16±$_{1.23}$	57.57±$_{2.02}$	71.39±$_{0.93}$	64.32±$_{0.38}$
CR	72.32±$_{0.72}$	69.80±$_{0.59}$	69.31±$_{1.80}$	51.71±$_{1.25}$	70.69±$_{0.58}$	63.07±$_{0.43}$
NPC	75.98±$_{0.35}$	71.63±$_{1.06}$	73.04±$_{1.16}$	59.83±$_{0.54}$	72.66±$_{0.86}$	65.74±$_{0.60}$
SelfMix	71.74±$_{1.96}$	67.59±$_{1.44}$	69.55±$_{2.35}$	61.37±$_{5.20}$	64.45±$_{1.28}$	57.30±$_{1.98}$
DyGen	77.51±$_{0.33}$	74.32±$_{0.37}$	76.28±$_{0.46}$	71.55±$_{0.80}$	77.06±$_{0.43}$	73.66±$_{0.82}$
Supervised GT (0% Noise)	80.17±$_{0.57}$					
ChatGPT (Zero-shot)	69.33					
ChatGPT (10-shot)	70.03					
Ours	**78.67**±$_{0.11}$	**75.33**±$_{0.25}$	**79.37**±$_{0.14}$	**75.50**±$_{0.10}$	**78.27**±$_{0.06}$	**74.58**±$_{0.08}$

Table 23: The detailed results (accuracy with standard deviation %) on 20ng datasets with IF 10. **Bold** means the best score.

Dataset	**20ng**					
Imbalance Factor	**50**					
Method(↓) / **Noise**(→)	20%**S**	40%**S**	20%**A**	40%**A**	20%**I**	40%**I**
BERT	68.84±$_{1.28}$	59.24±$_{2.25}$	64.10±$_{1.97}$	45.77±$_{3.09}$	64.63±$_{1.12}$	56.86±$_{1.43}$
Co-Teaching	67.18±$_{0.95}$	61.34±$_{2.08}$	59.67±$_{0.89}$	43.06±$_{2.22}$	63.17±$_{1.73}$	55.88±$_{1.53}$
CL	61.90±$_{1.03}$	57.44±$_{1.12}$	62.02±$_{1.15}$	53.72±$_{1.82}$	63.96±$_{0.57}$	60.48±$_{1.04}$
CR	62.98±$_{0.70}$	57.22±$_{1.95}$	59.51±$_{0.51}$	44.29±$_{3.51}$	63.28±$_{1.19}$	57.60±$_{2.29}$
NPC	66.13±$_{0.64}$	60.34±$_{0.95}$	61.93±$_{1.28}$	52.14±$_{1.57}$	64.32±$_{1.67}$	58.94±$_{0.67}$
SelfMix	50.96±$_{5.06}$	46.46±$_{2.34}$	53.19±$_{3.46}$	45.80±$_{3.83}$	53.16±$_{3.68}$	48.67±$_{3.64}$
DyGen	63.82±$_{0.75}$	61.54±$_{1.25}$	62.86±$_{0.83}$	57.59±$_{0.63}$	64.34±$_{0.95}$	60.90±$_{2.60}$
Supervised GT (0% Noise)	68.07±$_{0.61}$					
ChatGPT (Zero-shot)	69.33					
ChatGPT (10-shot)	70.03					
Ours	**72.01**±$_{0.12}$	**70.65**±$_{0.15}$	**73.04**±$_{0.44}$	**71.50**±$_{0.30}$	**72.88**±$_{0.22}$	**70.27**±$_{0.21}$

Table 24: The detailed results (accuracy with standard deviation %) on 20ng datasets with IF 50. **Bold** means the best score.

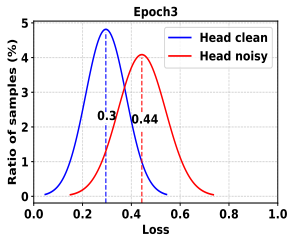


(a) Different Classes

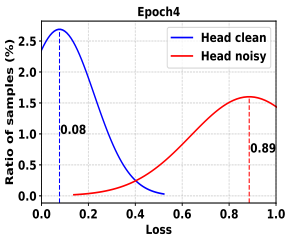
(b) Different Classes

(c) Different Classes

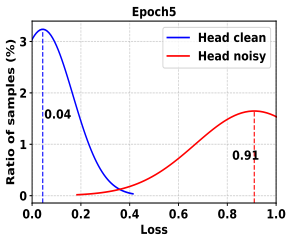
(d) Different Classes



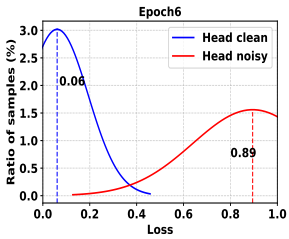
(e) Head Class



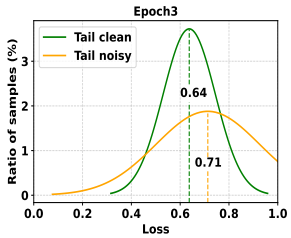
(f) Head Class



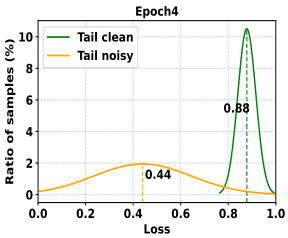
(g) Head Class



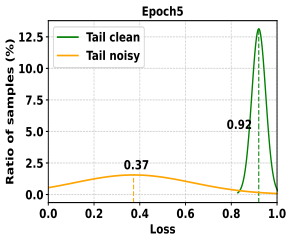
(h) Head Class



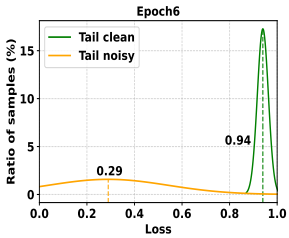
(i) Tail Class



(j) Tail Class

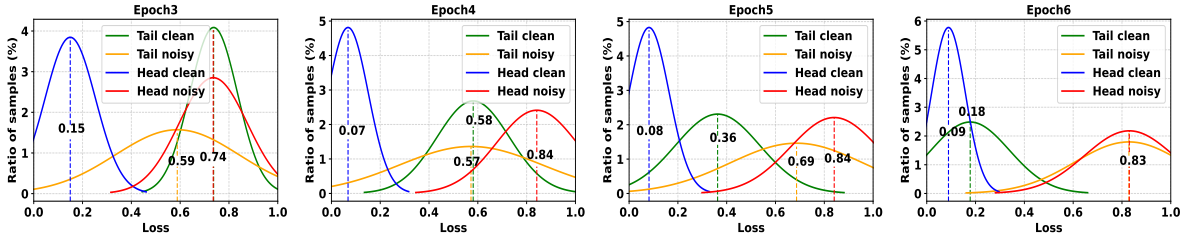


(k) Tail Class

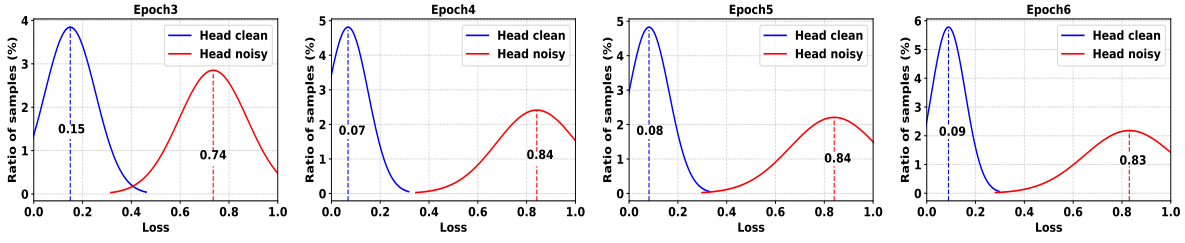


(l) Tail Class

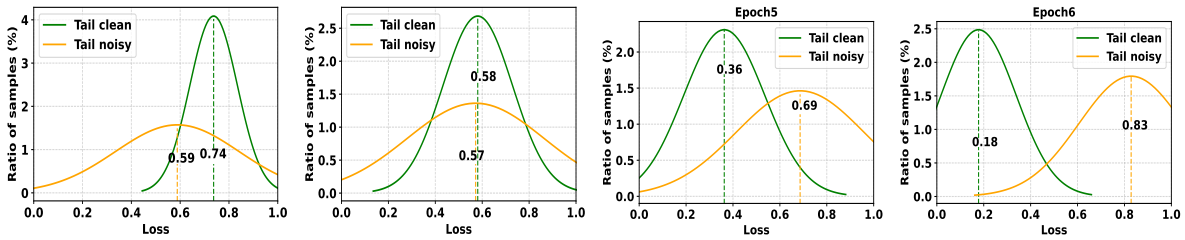
Figure 12: The loss distributions of SelfMix on R8 under 40% asymmetric label noise (a-d), 40% asymmetric label noise on tail class (e-h), 40% asymmetric label noise on head class (i-l). The solid line represents the loss distributions, and the dashed line points out the mean value of loss distributions. We observe that: (1) the loss distribution of clean samples and noisy samples, whether in the head class or tail class, dynamically evolves during the training process; (2) both tail class samples and noisy samples exhibit large losses; (3) the losses of some clean samples belonging to the tail are even larger than the losses of some noisy ones from the head class; (4) For the head class, the clean samples tend to have a smaller loss value and the noisy samples tend to have a bigger loss value; (5) For the tail class, the noisy samples tend to have a smaller loss value and the clean samples tend to have a bigger loss value. Since existing small-loss-based sample selection methods always set a global fixed loss value to separate noisy data, they tend to fail when distinguishing clean and noisy samples on imbalanced datasets.



(a) Different Classes (b) Different Classes (c) Different Classes (d) Different Classes

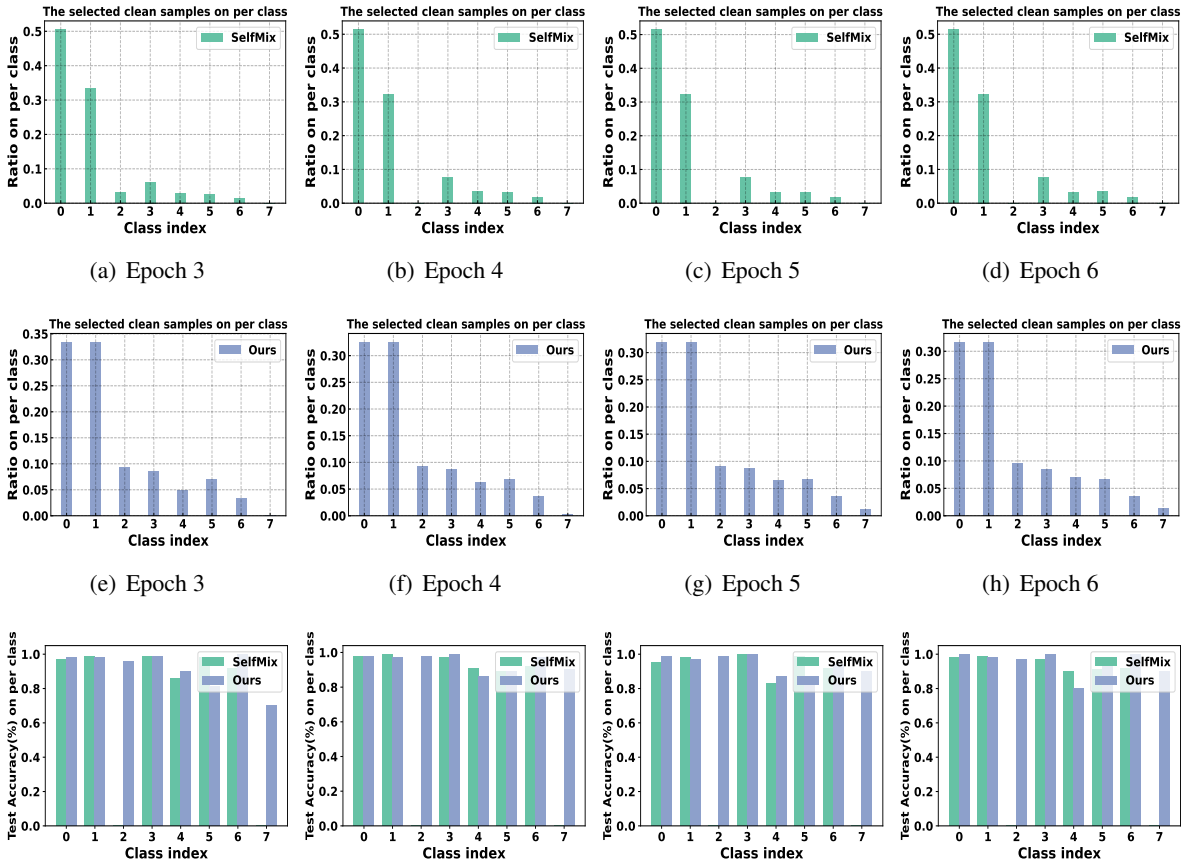


(e) Head Class (f) Head Class (g) Head Class (h) Head Class



(i) Tail Class (j) Tail Class (k) Tail Class (l) Tail Class

Figure 13: The loss distributions of our methods on R8 under 40% asymmetric label noise (a-d), 40% asymmetric label noise on tail class (e-h), 40% asymmetric label noise on head class (i-l). The solid line represents the loss distributions, and the dashed line points out the mean value of loss distributions. For the tail class, the loss values of the clean samples decrease from high to low during the training process. At the same time, the loss values of the noisy samples increase from low to high during training. Ultimately, the loss distribution of the clean and noisy samples becomes consistent across both head and tail classes. This phenomenon proves that our method can effectively learn from the clean samples of the tail classes while avoiding overfitting noisy samples.



(a) Epoch 3 (b) Epoch 4 (c) Epoch 5 (d) Epoch 6

(e) Epoch 3 (f) Epoch 4 (g) Epoch 5 (h) Epoch 6

(i) Epoch 3 (j) Epoch 4 (k) Epoch 5 (l) Epoch 6

Figure 14: The class distribution of clean samples selected by SelfMix on the imbalanced dataset R8 under 40% asymmetric noise (a-d) in different training stages. The class distribution of clean samples selected by our method on the imbalanced dataset R8 under 40% asymmetric noise (c-h) in different training stages. Due to self-training, SelfMix gradually ignored the samples in Class 2 and consistently failed to select clean samples from Class 7 to learn, which maintains or exacerbates the imbalanced ratio of selected clean samples. However, our method progressively selects clean samples from Class 7 without neglecting any class, resulting in an debiased sample selection. This debiased sample selection prompts our method better learn from each class and obtain greater performance in these categories (i-l).

Datasets	Label	Description
Chemprot	Part of	CEM that are structurally related to a GPRO, $e.g.$, specific amino acid residues of a protein.
	Regulator	CEM that clearly regulates a GPRO, but for which there is no further information on whether the regulation is direct or indirect.
	Upregulator	CEM that increments a GPRO signal, without any insight on the mechanism. Despite this class is named "Upregulator", it comprises positive regulation (increments): direct protein activation by binding or indirect up-regulation of GPRO expression/protein levels or post translational modification).
	Downregulator	CEM that decreases a GPRO signal, without any insight on the mechanism. Despite this class is named "Downregulator", it comprises negative regulation (decrements): direct protein inhibition by binding or indirect downregulation of GPRO expression/protein levels or post translational modification).
	Agonist	CEM that binds to a receptor and alters the receptor state resulting in a biological response. Conventional agonists increase receptor activity, whereas inverse agonists reduce it. If no information is provided on whether the CEM activates or reduces GPRO activity, this general class should be assigned.
	Antagonist	CEM that reduces the action of another CEM, generally an agonist. Many antagonists act at the same receptor macromolecule as the agonist.
	Modulator	CEM that acts as allosteric modulator, compound that increases or decreases the action of an (primary or orthosteric) agonist or antagonist by combining with a distinct (allosteric or allotropic) site on the receptor macromolecule. If no information is available on whether the CEM activates or reduces GPRO activity, this general subclass should be assigned.
	Cofactor	CEM that is required for a protein's biological activity to happen.
	Substrate/Product	CEM that is both, substrate and product of enzymatic reaction. Specifically, "Substrate" indicates the CEM upon which a GPRO (typically protein) acts, and it should be understood as the substrate of a reaction carried out by a protein ("reactant") or as transporter substrate. "Product" indicates that CEM is a product of enzymatic reaction or a transporter.
	Not	This class should be used to define the NEGATIVE occurrence of a chemical-protein interaction, without providing any further information on the specific negative CHEMPROT class or class.

Table 25: Label Descriptions used in main experiments, where "CEM" represents the Chemical Entities Mention, and "GPRO" represents the Gene and Protein Related Object (Krallinger et al., 2017).

Datasets	Label	Description
Trec	description	any type of communication that aims to make vivid a place, object, person, group, or other physical entity.
	entity	something that exists as itself. It does not need to be of material existence.
	human	the most common and widespread species of primate, and the last surviving species of the genus Homo.
	numeric	a mathematical object used to count, measure, and label.
	location	it is likely to have a well-defined name but a boundary that is not well defined varies by context
	abbreviation	a shortened form of a word or phrase, by any method.
MR	negative	is a personality variable that involves the experience of negative emotions and poor self-concept.
	positive	is a human characteristic that describes how much people experience positive affects (sensations, emotions, sentiments); and as a consequence how they interact with others and with their surroundings.

Table 26: Label Descriptions used in main experiments, which are drawn from Wikipedia.

Datasets	Label	Description
SEMEVAL	Cause-Effect	An event or object yields an effect. Example: those cancers were caused by radiation exposures.
	Component-Whole	An object is a component of a larger whole. Example: my apartment has a large kitchen.
	Content-Container	An object is physically stored in a delineated area of space. Example: a bottle full of honey was weighed.
	Entity-Destination	An entity is moving towards a destination. Eg. the boy went to bed.
	Entity-Origin	An entity is coming or is derived from an origin (e.g., position or material). Example: letters from foreign countries.
	Instrument-Agency	An agent uses an instrument. Example: phone operator.
	Member-Collection	A member forms a nonfunctional part of a collection. Example: there are many trees in the forest.
	Message-Topic	An act of communication, written or spoken, is about a topic. Example: the lecture was about semantics
	Product-Producer	A producer causes a product to exist. Example: a factory manufactures suits.

Table 27: Label Descriptions used in main experiments, which are drawn from the Semeval dataset (Hendrickx et al., 2010)

Datasets	Label	Description
20ng	alt.atheism	A theism is, in the broadest sense, an absence of belief in the existence of deities. Less broadly, atheism is a rejection of the belief that any deities exist.
	comp.graphics	Computer graphics is the discipline of generating images with the aid of computers. Today, computer graphics is a core technology in digital photography, film, video games, cell phone and computer displays, and many specialized applications. A great deal of specialized hardware and software has been developed, with the displays of most devices being driven by computer graphics hardware. It is a vast and recently developed area of computer science. The phrase was coined in 1960 by computer graphics researchers Verne Hudson and William Fetter of Boeing. It is often abbreviated as CG, or typically in the context of film as CGI.
	comp.sys.ibm.pc.hardware	A personal computer (PC) is a multi-purpose computer whose size, capabilities, and price make it feasible for individual use. Personal computers are intended to be operated directly by an end user, rather than by a computer expert or technician. Unlike large costly minicomputer and mainframes, time-sharing by many people at the same time is not used with personal computers.
	comp.sys.mac.hardware	The Macintosh (branded simply as Mac since 1998) is a family of personal computers designed, manufactured and sold by Apple Inc. since January 1984.
	comp.windows.x	Windows XP is a personal computer operating system produced by Microsoft as part of the Windows NT family of operating systems. It was released to manufacturing on August 24, 2001, and broadly released for retail sale on October 25, 2001.
	misc.forsale	Online shopping is a form of electronic commerce which allows consumers to directly buy goods or services from a seller over the Internet using a web browser. Consumers find a product of interest by visiting the website of the retailer directly or by searching among alternative vendors using a shopping search engine, which displays the same products availability and pricing at different eretailers. As of 2016, customers can shop online using a range of different computers and devices, including desktop computers, laptops, tablet computers and smartphones.
	rec.autos	A car (or automobile) is a wheeled motor vehicle used for transportation. Most definitions of cars say that they run primarily on roads, seat one to eight people, have four tires, and mainly transport people rather than goods.

Table 28: Label Descriptions used in main experiments, which are drawn from Wikipedia for the 20ng dataset (Chai et al., 2020).

Datasets	Label	Description
20ng	rec.motorcycles	A motorcycle, often called a bike, motorbike, or cycle, is a two- or three-wheeled motor vehicle. Motorcycle design varies greatly to suit a range of different purposes: long distance travel, commuting, cruising, sport including racing, and off-road riding. Motorcycling is riding a motorcycle and related social activity such as joining a motorcycle club and attending motorcycle rallies.
	rec.sport.baseball	Baseball is a bat-and-ball game played between two opposing teams who take turns batting and fielding. The game proceeds when a player on the fielding team, called the pitcher, throws a ball which a player on the batting team tries to hit with a bat. The objective of the offensive team (batting team) is to hit the ball into the field of play, allowing its players to run the bases, having them advance counter-clockwise around four bases to score what are called ¨runs¨. The objective of the defensive team (fielding team) is to prevent batters from becoming runners, and to prevent runners advance around the bases. A run is scored when a runner legally advances around the bases in order and touches home plate (the place where the player started as a batter). The team that scores the most runs by the end of the game is the winner.
	rec.sport.hockey	Hockey is a sport in which two teams play against each other by trying to manoeuvre a ball or a puck into the opponents goal using a hockey stick. There are many types of hockey such as bandy, field hockey, ice hockey and rink hockey.
	sci.crypt	In cryptography, encryption is the process of encoding a message or information in such a way that only authorized parties can access it and those who are not authorized cannot. Encryption does not itself prevent interference, but denies the intelligible content to a would-be interceptor. In an encryption scheme, the intended information or message, referred to as plaintext, is encrypted using an encryption algorithm cipher generating ciphertext that can be read only if decrypted. For technical reasons, an encryption scheme usually uses a pseudo-random encryption key generated by an algorithm. It is in principle possible to decrypt the message without possessing the key, but, for a welldesigned encryption scheme, considerable computational resources and skills are required. An authorized recipient can easily decrypt the message with the key provided by the originator to recipients but not to unauthorized users.
	sci.electronics	Electronics comprises the physics, engineering, technology and applications that deal with the emission, flow and control of electrons in vacuum and matter.

Table 29: Label Descriptions used in main experiments, which are drawn from Wikipedia for the 20ng dataset (Chai et al., 2020).

Datasets	Label	Description
20ng	sci.med	Medicine is the science and practice of establishing the diagnosis, prognosis, treatment, and prevention of disease. Medicine encompasses a variety of health care practices evolved to maintain and restore health by the prevention and treatment of illness. Contemporary medicine applies biomedical sciences, biomedical research, genetics, and medical technology to diagnose, treat, and prevent injury and disease, typically through pharmaceuticals or surgery, but also through therapies as diverse as psychotherapy, external splints and traction, medical devices, biologics, and ionizing radiation, amongst others.
	sci.space	Outer space, or simply space, is the expanse that exists beyond the Earth and between celestial bodies. Outer space is not completely empty it is a hard vacuum containing a low density of particles, predominantly a plasma of hydrogen and helium, as well as electromagnetic radiation, magnetic fields, neutrinos, dust, and cosmic rays.
	soc.religion.christian	Christians are people who follow or adhere to Christianity, a monotheistic Abrahamic religion based on the life and teachings of Jesus Christ. The words Christ and Christian derive from the Koine Greek title Christ, a translation of the Biblical Hebrew term mashiach.
	talk.politics.guns	A gun is a ranged weapon typically designed to pneumatically discharge solid projectiles but can also be liquid (as in water guns/cannons and projected water disruptors) or even charged particles (as in a plasma gun) and may be free-flying (as with bullets and artillery shells) or tethered (as with Taser guns, spearguns and harpoon guns).
	talk.politics.mideast	The Middle East is a transcontinental region which includes Western Asia (although generally excluding the Caucasus), and all of Turkey (including its European part) and Egypt (which is mostly in North Africa). The term has come into wider usage as a replacement of the term Near East (as opposed to the Far East) beginning in the early 20th century. The broader concept of the Greater Middle East (orMiddle East and North Africa) also adds the Maghreb, Sudan, Djibouti, Somalia, Afghanistan, Pakistan, and sometimes even Central Asia and Transcaucasia into the region. The term Middle East has led to some confusion over its changing definitions.
	talk.politics.misc	Politics is a set of activities associated with the governance of a country, state or an area. It involves making decisions that apply to groups of members.
	talk.religion.misc	Religion is a social-cultural system of designated behaviors and practices, morals, worldviews, texts, sanctified places, prophecies, ethics, or organizations, that relates humanity to supernatural, transcendental, or spiritual elements. However, there is no scholarly consensus over what precisely constitutes a religion.

Table 30: Label Descriptions used in main experiments, which are drawn from Wikipedia for the 20ng dataset (Chai et al., 2020).

Datasets	Label	Description
20ng	comp.os.ms-windows.misc	Microsoft Windows is a product line of proprietary graphical operating systems developed and marketed by Microsoft. It is grouped into families and sub-families that cater to particular sectors of the computing industry – Windows (unqualified) for a consumer or corporate workstation, Windows Server for a server and Windows IoT for an embedded system. Defunct families include Windows 9x, Windows Mobile, Windows Phone, and Windows Embedded Compact.

Table 31: Label Descriptions used in main experiments, which are drawn from Wikipedia for the 20news dataset (Chai et al., 2020).

Datasets	Label	Description
AGNews	World	It's a news article about international affairs, geopolitics, global events, or any topic that has a worldwide or international scope. Examples may include news on international diplomacy, major global events like the United Nations General Assembly, international conflicts or wars, significant elections or political events in different countries, global environmental issues, and more.
	Sports	Articles related to various sporting events, news, and updates. the Sports category could encompass a wide range of topics such as game results, player transfers, injuries, interviews with athletes, coverage of international sporting events like the Olympics, football (soccer) world cup, tennis grand slams, and more.
	Business	The Business category typically cover topics related to commerce, economics, and finance on a local, national, or international scale. It may include news about company mergers, financial reports, stock market updates, changes in economic policies, interviews with business leaders, innovation in business models, trends in various industry sectors, and so on.
	Sci/Tech	The Science/Technology category is designed to encompass articles related to science and technology. It might include news about scientific discoveries or research breakthroughs, technology product launches, technology company updates, coverage of scientific and technology conferences, interviews with scientists or tech leaders, articles on new theories or models in various scientific disciplines, advancements in medical technology, and many more.

Table 32: Label Descriptions used in main experiments, which are drawn from Wikipedia for the AGNews dataset (Luo et al., 2024).

Datasets	Label	Description
R8	earn	Income and money related topics
	acq	Corporate acquisitions related topics
	crude	Crude oil related topics
	trade	Domestic and foreign trade related topics
	money-fx	Money foreign exchange related topics
	interest	Interest rate related topics
	ship	Global shipping and transport commerce related topics
	grain	The grain's trade, price, security and related topics

Table 33: Label Descriptions used in main experiments, which are obtained from (Cai et al., 2014).

Trec

Task description:

You are a text classifier and your task is to classifiy a given sentence with the following classes (along with some examples):

Demonstration:

Class descriptions:

1.description, which is any type of communication that aims to make vivid a place, object, person, group, or other physical entity.

-what does the name shawn mean

2.entity, which is something that exists as itself. It does not need to be of material existence.

-what s the common name for acetylsalicylic acid

3.human, which are the most common and widespread species of primate, and the last surviving species of the genus Homo.

-what is the viking prince s first name

4.numeric, which is a mathematical object used to count, measure, and label.

-what is columbia tristar s phone number

5.location, which is likely to have a well-defined name but a boundary that is not well defined varies by context.

-what s the most common street name in America

6.abbreviation, which is a shortened form of a word or phrase, by any method.

-what does bud stand for

Based on the descriptions of the categories provided, along with the respective examples under each category, please classify the test sentence into one of the previously described classes. Don't explain other things. Your answer must be one of description, entity, human, numeric, location, abbreviation.

Inputs:

Query: Consider the following test sentence:

Query Text: what is smokey the bear s middle name

Table 34: The prompt instruction (1-shot) for ChatGPT on **Trec**.