

Divide, Conquer, and Distill: Unified Sample Selection and LLM Distillation for Noisy Long-Tailed Learning

Bo Yuan¹, Yulin Chen², Jie Cao³, Mei Gao^{1*}

¹Test Center, National University of Defense Technology, Xi'an, China

²Independent Researcher, China

³Zhejiang University, Hangzhou, China

byuan.186@qq.com, sylvia_cyl@qq.com, caojie@zju.edu.cn, gao_mei@nudt.edu.cn

Abstract

There is a rising interest in further exploring learning with noisy labels on textual data. Existing solutions often assume the ground truth classes of the training data are balanced, overlooking the long-tailed distribution prevalently encountered in real-world data, which can significantly harm the effect of denoising strategies. Meanwhile, misidentifying clean samples further compromises the model's learning of long-tailed data. To tackle this issue, we propose a novel knowledge distillation framework for learning with noisy labels under long-tailed distribution (LNL-LT). Specifically, we first introduce a divide-and-conquer sample selection strategy to perform head-tail class division and better separate clean samples from noisy samples within these classes, especially for the tail classes. Then, we employ LLM knowledge distillation to generate pseudo-labels and new samples for tail classes, enhancing pseudo-label quality while mitigating the effects of sample scarcity caused by the long-tailed distribution. Experimental results on both synthetic and real-world datasets demonstrate that our method surpasses current state-of-the-art methods.

1 Introduction

The core of deep learning success lies in high-quality annotated datasets. However, datasets in real-world applications often exhibit two issues: (1) *class-imbalanced*, where a long-tailed class distribution exhibits. In this case, the model tends to learn head classes and ignore tail classes (Wang et al., 2024c). (2) *noisy labels*, where a part of samples is wrongly labeled due to mistakes of annotators, which impair the generalization performance of models (Qiao et al., 2022).

While many prior studies have tackled issues of long-tailed data and noisy labels individually, they struggle to achieve strong performance when both

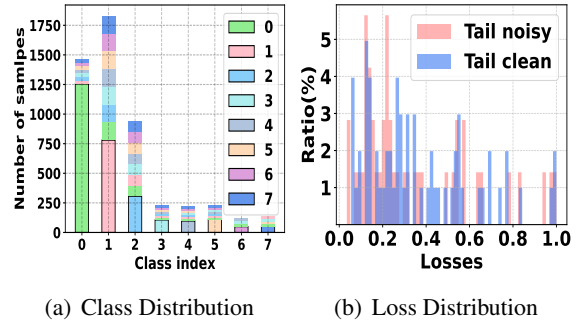


Figure 1: R8 dataset with 40%A: (a) The observed class distribution reflects distribution imbalance and label noise. Black boxes represent clean samples under this class, while no boxes represent noisy samples. (b) Loss distributions of clean and noisy samples in the tail class.

challenges coexist. Concretely, the methods for learning with noisy labels (Wang et al., 2023; Kim et al., 2024) always distinguish clean samples from noisy ones under the assumption that distributions are balanced and may fail when applied to long-tailed data. Moreover, the methods for learning with long-tailed data (Huang et al., 2022; Zhang et al., 2023) often assume that labels in a dataset are clean, whereas applying these methods directly to a long-tailed data with noisy labels will adversely affect model training by accumulating label error.

To this end, several works have explored Learning with Noisy Labels on Long-Tailed data (LNL-LT). Among them, the most popular methods (Wei et al., 2022; Zhang et al., 2023; Li et al., 2024b) have considered the distance between input features and class centroids to select clean samples. However, a recent study (Baik et al., 2024) proposes that these methods do not actively seek to enhance class feature representation quality, which limits their effectiveness for tail classes, where obtaining high-quality feature representations is more critical than for head classes.

Recently, LLMs have exhibited strong advance-

*Corresponding Author

ments in generating synthetic data or pseudo labels tailored to specific tasks (Li et al., 2023; Wang et al., 2024b). Inspired by this, we aim to leverage LLMs to generate extended data for tail classes, helping the model learn richer feature representations of tail classes. Additionally, we also intend to leverage LLMs to generate pseudo-labels for noisy samples in the tail classes to ensure that valuable feature information in the original non-synthetic training data is not wasted. Nevertheless, *effectively leveraging LLMs for enhancing feature representations of tail classes is not trivial*: (1) *Tail invisibility*. It is more difficult to directly recognize tail classes because the observed and true distributions are likely inconsistent due to noisy labels. As shown in Fig.1(a), class 2 is essentially a tail class but is occupied by noisy data, making it no longer a tail class by observation. (2) *Tail indivisibility*. Even if the tail class is identified, it is more difficult to distinguish between clean and noisy samples in the tail class because the samples in the tail class tend to be ignored during model training, which makes their values of the separation metric highly similar (e.g., training losses, see Fig.1(b)). So, our solution comes from thinking: *To leverage LLMs effectively, how can we distinguish between head and tail classes and further accurately select clean samples and noisy samples from tail classes?*

Unfortunately, previous selection methods (Shu et al., 2019; Qiao et al., 2022; Wei et al., 2022) before the era of LLMs focused solely on selecting clean and noisy samples in a unified way, neglecting the recognition and refined utilization of long-tail classes. In this paper, we revamp the selection process from a more fundamental perspective and propose a Divide-and-Conquer sample Selection strategy (DCS) to revolutionize the previous methods. Specifically, we first use unsupervised contrastive learning to generate robust representations for all training samples. Since this process is unaffected by corrupted labels, the resulting representations naturally maintain high robustness. Based on these representations, we apply a clustering algorithm to categorize the dataset into head and tail classes. Subsequently, distinct separation metrics are applied separately to head and tail classes to select clean and noisy samples. Moreover, *DCS can also provide tail class information for LLMs to enhance feature representation of tail classes*. To integrate DCS and LLMs as a whole for LNL-LT, we propose a novel knowledge distillation framework where the small language model (SLM) operates as

the student model performing DCS during training to obtain the information on tail classes for actively querying LLMs, and the LLM acts as the teacher model generating synthetic data and pseudo-labels to expand tail classes and correct their noisy samples. Benefiting from LLMs, the SLM can perform robust training on the more balanced and less noisy subset. We call this framework **Loki** (Learning with noisy labels on long-tailed data via DCS and LLM Knowledge distillation based on tail data), which effectively solves the LNL-LT problem on textual data. Overall, our main contributions are:

- We propose an innovative divide-and-conquer sample selection strategy, which can distinguish between long and tail classes and select clean and noisy samples within them.
- To the best of our knowledge, we are pioneers in leveraging LLM knowledge distillation to address the LNL-LT problem by enhancing class feature representations of tail data.
- We conduct extensive experiments on both simulated and real-world datasets. The results demonstrate our method **Loki**’s superiority over existing state-of-the-art methods.

2 Related Work

Learning with Noisy Labels (LNL). Real-world datasets inevitably contain noisy labels. To mitigate the influence of noisy labels in text classification, Qiao et al. (2022) uses the Gaussian Mixture Model to separate samples and leverages semisupervised learning, Zhuang et al. (2023) utilizes the dynamics of features with generative models to learn the transition matrix for noisy labels, and Wang et al. (2023) tackles noisy labels in classification tasks with supplemental guidance from LLMs (e.g., ChatGPT). Recently, Kim et al. (2024) proposes a routing-based parameter-efficient fine-tuning method for LNL, and Yuan et al. (2024) introduces a co-prediction network consisting of BERT and BiLSTM to split the noisy dataset. These existing LNL methods all assume the classes of the training data are balanced. However, many text datasets are imbalanced (Li et al., 2024a), and the mechanism of sample selection is sensitive to data imbalance, which may harm the model performance. To bridge this gap, we explore the LNL-LT problem in classification tasks.

Learning with Long-Tailed data. Previous methods specific to the long-tailed distribution problem are mainly divided into two categories:

re-sampling (Song et al., 2022) and re-weighting (Menon et al., 2021). Recently, Li et al. (2024a) leverages the supervised contrastive learning and prototypes of each class to solve imbalanced text classification tasks, Wang et al. (2024a) transfers the statistics from relevant head classes to infer the distribution of tail classes. However, these methods mentioned above rely on real class sample sizes to recognize head class and tail class, which are unavailable when label noise exists.

3 Methodology

3.1 Preliminaries

Problem Setup. Given a training dataset $\mathcal{D}=\{(x_i, \tilde{y}_i)\}_{i=1}^n$ with n samples and K classes, where $\tilde{y} \in [1, K]$ is the observed label of the sample x , and $\tilde{y}$ is possibly an incorrect label. The predictions obtained from the model f are denoted as $\mathbf{p}_i = f_{\theta}(x_i) = [p_i^1, \dots, p_i^K]$, where θ is the model’s parameters and p_i^d is the d -th dimension of vector $\mathbf{p}_i$. Meanwhile, the training data $\mathcal{D}$ also follows a long-tailed distribution with an imbalance ratio of ρ , where $\rho = \max_k(n_k)/\min_k(n_k)$, n_k is the number of training data belonging to k -th class. This paper aims to robustly train a model by using this imbalanced and noisily labeled training dataset (LNL-LT). To solve this task, we introduce a novel framework Loki, which explores the opportunity to enhance tail class feature representation via LLM knowledge distillation in the LLM era.

Observation. In the task of LNL-LT, the training data is predominantly composed of head class samples. During the early training stages, the model primarily learns patterns associated with these head classes. As shown in Fig.2 (a), relying solely on cross-entropy loss—a common approach in previous sample selection methods—to identify noisy samples struggles to effectively differentiate between mislabeled samples and those from tail classes. This difficulty arises due to the significant overlap in loss values for these two types of samples. In contrast, for samples with noisy labels and those belonging to head classes, the cross-entropy loss still seems to have the potential to distinguish them (refer to Fig.2 (b)).

Algorithm Overview. In the following, we discuss the proposed framework Loki step by step. Generally, Loki consists of three stages: (1) Dividing the training dataset into different subsets by a divide-and-conquer sample selection strategy, which can obtain clean samples and noisy samples

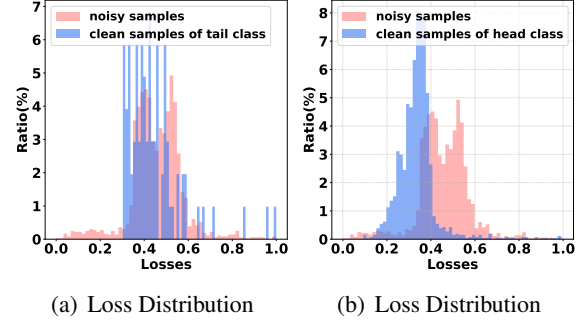


Figure 2: The loss values are calculated on the imbalanced dataset R8 under 40% asymmetric noise.

in tail classes. (2) Improving the quality of tail data by leveraging LLMs to generate new samples and pseudo-labels, which can be performed using clean samples of tail classes to construct proper LLM prompts. (3) Training a robust model by learning from different subsets. The procedure of Loki is illustrated in Fig.3. We provide more technical details as follows. The algorithm flow of Loki is provided in Algorithm 1.

3.2 Divide-and-Conquer Sample Selection

In the complex situation of noisy labels with long-tail distribution, previous sample selection methods focused only on separating clean samples from noisy ones. They mainly do not focus on tail class information. On the other hand, in our preliminary observation (see Fig.2), we found that the cross-entropy loss, commonly used in previous sample selection methods, fails to simultaneously distinguish clean samples from noisy samples in both head and tail classes. The above two aspects motivate us to propose a Divide-and-Conquer sample Selection strategy (DCS) to partition the dataset through two steps: (1) head-tail class division, then separately (2) selection of clean and noisy samples.

3.2.1 Dividing: Head-tail Class Division

Since the observed class distribution may be inconsistent with the true class distribution (see Fig.1(a)), it is infeasible to directly recognize head classes and tail classes based on the class number size when label noise exists. Given this, let us revisit the core issue in this step: *Can we use the text information to divide head and tail classes without considering observed labels?* This inspires us to resort to text clustering, which can divide head classes and tail classes based on the number of samples in each cluster. A typical method for text

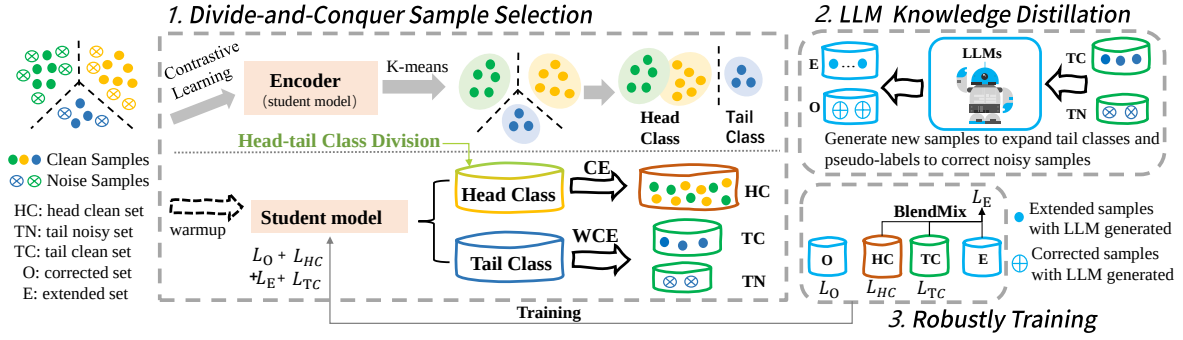


Figure 3: The overview of the Loki framework.

clustering is to apply clustering algorithms based on pre-trained embeddings. However, the quality of these embeddings directly impacts the final clustering results, potentially affecting their accuracy and robustness. To improve the robustness of text embeddings, we exploit unsupervised contrastive learning. Intuitively, as the text embeddings in unsupervised contrastive learning do not access the labels of training data, the achieved embeddings will not be influenced by incorrect labels. Besides, prior work (Liu et al., 2022) shows that contrastive learning can improve the network’s tolerance to long-tailed data. Thus, text embeddings achieved with contrastive learning help text clustering to better tackle noisy labels in long-tailed cases.

Specifically, we apply unsupervised contrastive learning via SimCSE (Gao et al., 2021) to get the text embeddings of samples, and then apply standard clustering algorithms, such as K-means (Huang and He, 2024), to obtain the final cluster results. Note that the number of clusters is equal to the number of classes, which eliminates the need to select clustering hyperparameters. Clusters with a larger number of samples are more likely to be head classes. This enables the partitioning of head and tail classes based on the number of samples ($|\mathcal{D}_{cluster_i}|$) within each cluster ($\mathcal{D}_{cluster_i}$). That is, we rank the sample number of all clusters and select clusters with top- S smallest values as tail classes, while the rest are considered head classes:

$$\mathcal{K}_{tail} = \text{argTopS}[|\mathcal{D}_{cluster_1}|, \dots, |\mathcal{D}_{cluster_K}|], \quad (1)$$

$$\mathcal{K}_{head} = \{1, \dots, K\} - \mathcal{K}_{tail}, \quad (2)$$

where $\mathcal{K}_{head}$ and $\mathcal{K}_{tail}$ are the cluster indices corresponding to head classes and tail classes, respectively. Thus, we can obtain samples of the head class $\mathcal{D}_{head} = \{x_i | x_i \in \bigcup_{k \in \mathcal{K}_{head}} \mathcal{D}_{cluster_k}\}$ and tail class $\mathcal{D}_{tail} = \{x_i | x_i \in \bigcup_{k \in \mathcal{K}_{tail}} \mathcal{D}_{cluster_k}\}$.

3.2.2 Conquering: Selection of Clean and Noisy Samples

In the previous step, we reduce the separation granularity to alleviate the interference from the head class to the tail class. Next, we adopt different separation metrics to separately select clean and noisy samples from head and tail classes.

Estimating clean probability for samples in head classes. To derive the clean probability for each sample, existing methods (Qiao et al., 2022; Kim et al., 2024) for learning with noisy labels always leverage the small-loss phenomena: *clean samples tend to have a smaller cross-entropy loss than noisy samples in the early training stage*. These methods generally assume that the class distribution of the training data is balanced and may not be directly applicable to LNL-LT. However, we have preliminarily divided the long-tailed noisy data into head and tail classes in the previous step, the sample sizes of all categories in the head class group are relatively balanced. This enables the selection of clean samples in the head classes based on loss deviation. Moreover, our preliminary observations (see Fig.2 (b)) also confirm that small-loss phenomena are still helpful in selection. Taking advantage of such phenomena, we adopt the Gaussian Mixture Model (GMM) (Li et al., 2020) to estimate the probability of samples in head classes being clean by the per-sample cross-entropy loss.

Specifically, for a sample x in head classes, we compute the clean probability $\zeta(x)$ using the posterior probability $p(g|l_{ce}(x))$, where $l_{ce}(x)$ is the cross-entropy (CE) loss, and g is the Gaussian component with a smaller mean value (smaller loss). Then, we set a threshold ξ for $\zeta(x)$ to distinguish whether x belongs to clean or noisy.

Estimating clean probability for samples in tail classes. Different from head classes, the noisy and clean samples in a tail class k tend to obtain

similar prediction confidence in the observed class k , because their predictions are both towards head classes. In this scenario, using CE as a separation metric fails to distinguish them effectively. This limitation is explained by Theorem 1, which provides the upper limit for the absolute difference in CE values between two samples.

Theorem 1 (Upper limit of CE difference). *Let x_i and x_j be two samples from a tail class k , with p_i^k and p_j^k representing the k -th dimension of their prediction confidences $\mathbf{P}_i$ and $\mathbf{P}_j$, respectively. The upper limit of the absolute difference between their CE values is given by:*

$$|CE(x_i) - CE(x_j)| \leq |p_i^k - p_j^k| / p_i^k. \quad (3)$$

Theorem 1 demonstrates that the difference in CE values is solely determined by the prediction confidence of the observed class, denoted as p_i^k and p_j^k . The proof of **Theorem 1** is given in Appendix A.1. Furthermore, as $|p_i^k - p_j^k|$ decreases for a fixed value p_i^k , the difference $|CE(x_i) - CE(x_j)|$ also diminishes. This implies that when two samples belong to the same class k with close values p_i^k , their CE values are also close (see Fig.1(b)), making it challenging to distinguish between them.

To overcome the dilemmas of overly concentrated loss distribution in tail-class samples, we propose a new separation metric, weighted cross-entropy (WCE), to distinguish two samples even if their values p_i^k are close. WCE incorporates an additional weighting factor into the CE to better distinguish samples by analyzing their prediction confidence across classes other than the observed class (i.e., p_i^m , $m \neq k$). Specifically, we first consider the maximum prediction confidence $\max(\mathbf{p}_i)$, as it may indicate the ground-truth class for a noisy sample y_i . Next, we compute the ratio of the maximum prediction confidence to the confidence of the observed class $\max(\mathbf{p}_i)/p_i^k$. This weighting factor $W(x) = \max(\mathbf{p}_i)/p_i^k$ exceeds one only when the class with the maximum prediction confidence differs from the observed class. The larger gap between $\max(\mathbf{p}_i)$ and p_i^k results in a higher weight. Finally, WCE can be calculated by: $WCE(x_i) = W(x_i) * CE(x_i)$.

Remark. *In contrast to CE, which relies solely on the prediction confidence of the observed class, incorporating the maximum prediction confidence in WCE provides a more effective means for tail classes to separate clean samples from noisy ones.*

After setting the new separation metric WCE for tail classes, we also use GMM to estimate the

probability of clean samples in tail classes. For a sample x in tail classes, $l_{wce}(x)$ is its WCE loss and $v(x) = p(g|l_{wce}(x))$ represents its clean probability. Then, we define a threshold φ for $v(x)$ to determine whether x belongs to clean or noisy.

Selection. Once the clean probability is estimated, we combine ξ and φ to perform selection. For head class $\mathcal{D}_{head}$, we apply ξ to distinguish the clean samples from noisy samples. Specifically, for a sample in $\mathcal{D}_{head}$, if its clean probability is higher than ξ , we put it into the head clean set HC : $HC = \{(x_i, \tilde{y}_i) | \zeta(x_i) \geq \xi\} \cap \mathcal{D}_{head}$. Similarly, for a sample in $\mathcal{D}_{tail}$, if its clean probability exceeds φ , it is placed in the tail clean set TC : $TC = \{(x_i, \tilde{y}_i) | v(x_i) \geq \varphi\} \cap \mathcal{D}_{tail}$. To capture more features of tail classes, we also select noisy samples from tail classes to form the tail noisy set TN : $TN = \{(x_i, \tilde{y}_i) | v(x_i) < \varphi\} \cap \mathcal{D}_{tail}$, which can be used for LLM correction and later learning.

3.3 Knowledge Distillation Based on Tail Data

In this part, given the strong performance of LLMs in handling NLP tasks, our proposed Loki uses LLMs as the teacher model to perform knowledge distillation on the filtered tail data to generate new samples and pseudo-labels, which can enhance the feature representation of tail classes to facilitate student model (i.e., SLMs) learning better. Specifically, we subdivided the generation process into two sub-steps: (1) **New Samples Generation.** To obtain more features of tail classes, we select a few examples from the TC to construct the prompt (see Appendix F.1), and query LLM to create new training samples for tail classes. For these generated new samples, we put them in the extended set E . (2) **Pseudo-labels Generation.** To ensure that the original non-synthetic textual features of the tail classes are not wasted, LLM is also prompted to regenerate pseudo-labels $\tilde{y}_i^{LLM}$ for noisy samples in tail classes (see Appendix F.1). Then, we place these corrected samples in the corrected set O : $O = \{(x_i, \tilde{y}_i^{LLM}) | (x_i, \tilde{y}_i) \in TN\}$.

In this knowledge distillation (KD) process, Loki allows the student model to directly communicate its needs to the teacher model, helping the teacher model identify targeted optimization directions. For instance, the query on the tail class will signal the teacher model to generate more relevant data samples for that class; this makes the KD process tail data aware and allows the teacher model to directly optimize the performance of the student model on the tail class.

Dataset	R8						AGNews ($\rho:10$)						AGNews ($\rho:50$)					
Noise	20%S	40%S	20%A	40%A	20%I	40%I	20%S	40%S	20%A	40%A	20%I	40%I	20%S	40%S	20%A	40%A	20%I	40%I
GT																		
LLM-base																		
BERT-base	91.19	81.59	94.39	80.16	88.77	71.65	85.97	69.81	83.79	65.08	88.78	73.37	78.51	69.11	77.15	61.11	86.25	65.15
SelfMix	<u>97.57</u>	94.62	96.40	92.17	82.07	77.32	87.56	86.83	87.55	86.56	86.09	77.78	80.80	78.87	80.07	78.18	85.43	69.41
DyGen	92.93	91.02	94.77	88.52	92.69	87.89	87.26	86.19	87.07	85.26	87.64	77.42	84.20	79.71	83.95	78.53	86.31	69.58
LAFT	92.34	85.22	96.18	81.94	91.26	77.73	88.18	87.27	88.45	87.39	87.95	80.77	84.65	80.18	80.99	77.54	87.75	75.44
NoiseAL	97.16	<u>95.01</u>	<u>96.51</u>	<u>94.29</u>	<u>93.08</u>	<u>90.12</u>	<u>90.10</u>	<u>88.06</u>	<u>90.52</u>	<u>88.18</u>	89.15	<u>85.71</u>	<u>85.39</u>	<u>81.35</u>	<u>86.66</u>	<u>80.03</u>	<u>87.95</u>	<u>80.06</u>
LADC	92.75	83.24	96.34	81.47	89.44	73.56	86.64	71.42	84.59	67.01	89.45	74.16	80.12	69.79	78.44	62.45	87.58	66.05
LWS	92.05	82.28	95.60	81.81	89.65	73.48	87.46	71.09	85.42	66.62	90.33	75.32	79.71	69.88	79.12	62.96	87.57	65.76
BSCE	92.91	82.68	94.91	81.35	90.72	72.41	87.70	71.52	85.17	65.98	90.49	74.32	79.47	70.53	78.08	62.05	87.20	65.83
SharpReCL	92.63	83.15	95.73	82.00	89.39	73.36	86.93	70.39	85.08	66.26	89.88	74.81	79.46	69.85	78.83	62.23	87.49	65.80
Ours	98.31	97.26	98.27	97.58	97.18	95.02	91.76	91.12	91.71	90.99	91.60	90.42	87.55	87.10	88.14	87.47	88.83	87.85

Table 1: Performance (average accuracy %) of Loki and other baselines on synthetic datasets with varying label noise and imbalance ratios. All experiments are repeated 5 times. The best and second results are highlighted in **Bold** and underlined, respectively. GT refers to the base model trained on ground truth data (0% noise), and LLM-base refers to directly using LLMs (ChatGPT).

3.4 Robustly Training

Learning from head clean set HC . For the HC , we utilize the cross-entropy loss $\mathcal{L}_{CE}$ to learn from it: $\mathcal{L}_{HC} = -\frac{1}{n} \sum_{i=1}^{|HC|} \mathcal{L}_{CE}(f(x_i), \tilde{y}_i)$, where $|HC|$ denotes the size of HC .

Learning from tail clean set TC . For the TC , we also utilize $\mathcal{L}_{CE}$ to learn from TC : $\mathcal{L}_{TC} = -\frac{1}{n} \sum_{i=1}^{|TC|} \mathcal{L}_{CE}(f(x_i), \tilde{y}_i)$, where $|TC|$ denotes the size of TC , n is the size of the entire dataset.

Learning from corrected set O . After generating the pseudo-labels by LLMs for samples in TN to construct the corrected set O , the number of noise samples in O is significantly less than that in TN . However, even the powerful LLM, cannot assure that all samples in O have the correct labels. To this end, we introduce symmetric cross entropy (SCE) (Wang et al., 2019), which can simultaneously address both the under learning and overfitting problems of CE in the presence of noisy labels. This robust property lets us optimize the model given a dataset with a lower noise ratio. Specifically, we utilize SCE loss for sample $(x_i, \tilde{y}_i^{LLM})$ in O : $\mathcal{L}_O = -\frac{1}{n} \sum_{i=1}^{|O|} \mathcal{L}_{SCE}(f(x_i), \tilde{y}_i^{LLM})$, where $|O|$ denotes the size of O .

Learning from extended set E . With generated tailed samples, the final problem is how to learn these generated samples in E efficiently. Due to the distribution gap between the generated data and the original tail data (Chen et al., 2024), we propose a BlendMix method to handle this challenge. To be specific, BlendMix first concatenates HC and TC to obtain the clean data C containing original samples. Then it balance-samples (Buda et al., 2018) a sample x_c from C

and chooses a sample x_e from E . Meanwhile, it mixes the sample x_c and x_e and their corresponding labels, denoted as $x' = \lambda_m x_e + (1 - \lambda_m) x_c$, $y'_i = \lambda_m y_e + (1 - \lambda_m) y_c$, $\lambda_m = \max(\lambda, 1 - \lambda)$, where λ in Beta (0,1) distribution. Next, we compute the loss: $\mathcal{L}_E = -\frac{1}{n} \sum_{i=1}^{n_e} \mathcal{L}_{CE}(f(x'_i), y'_i)$, where $n_e = \min(|C|, |E|)$, $|C| = |HC| + |TC|$, $|E|$ and $|C|$ is the size of E and C , respectively.

Finally, our overall training objective $\mathcal{L}$ can be calculated by: $\mathcal{L} = \mathcal{L}_{HC} + \mathcal{L}_{TC} + \mathcal{L}_O + \mathcal{L}_E$.

4 Experiments

4.1 Baselines and Dataset Configurations

Baselines. For comprehensive evaluations, we employ comparison methods as follows: (1) Methods designed for LNL include **SelfMix** (Qiao et al., 2022), **LAFT** (Wang et al., 2023), **DyGen** (Zhuang et al., 2023), **NoiseAL** (Yuan et al., 2024); (2) Methods designed for learning long-tail data include **SharpReCL** (Yi et al., 2022), **LADC** (Wang et al., 2024a), **LWS** (Kang et al., 2020), **BSCE** (Ren et al., 2020). See Appendix B for details.

Real-world datasets. we conduct experiments on imbalanced datasets with **real-world noise**: **ChemProt** (Krallinger et al., 2017), **SemEval** (Zhou et al., 2020), **TREC** (Awasthi et al., 2020).

Synthetic datasets. To construct the class-imbalanced versions of **AGNews** (Gulli, 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 ($n_0 \geq \dots \geq n_k$) and $\mu \in (0, 1]$. Generally, the imbalance ratio ρ is defined as $\frac{n_0}{n_k}$ to measure the imbalance degree, which is chosen from $\{10, 50\}$

Method	ChemProt	TREC	SEMEVAL
Noise Ratio	22.88%	38.56%	16.00%
BERT-base	64.84 $\pm$ 0.28	67.33 $\pm$ 0.83	71.44 $\pm$ 0.10
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
NoiseAL [†]	70.98 $\pm$ 0.46	78.84 $\pm$ 1.82	81.47 $\pm$ 0.62
LADC	68.19 $\pm$ 0.35	70.04 $\pm$ 0.42	74.21 $\pm$ 0.40
LWS	67.86 $\pm$ 0.45	70.43 $\pm$ 0.36	74.16 $\pm$ 0.41
BSCE	67.52 $\pm$ 0.33	70.30 $\pm$ 0.43	74.71 $\pm$ 0.25
SharpReCL	68.79 $\pm$ 0.23	70.49 $\pm$ 0.51	75.05 $\pm$ 0.36
Ours	72.23$\pm$0.17	79.07$\pm$0.50	82.91$\pm$0.09

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

in this paper. Then, we also select some known imbalanced dataset: **Trec** (Li and Roth, 2002), **R8** (Lewis, 1992). After obtaining the long-tailed distribution, three types of synthetic label noise are injected into datasets following the work (Zhuang et al., 2023): (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. (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). See Appendix C for details.

4.2 Implementation Details

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

For the student model, we choose the BERT. For the teacher model, we select the ChatGPT (e.g., GPT-3.5-Turbo-0613 API). Following previous work, 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 Loki and all baselines. The warmup epochs are set to 2, Loki first warms up student models for 2 epochs, then performs DCS and LLM knowledge distillation for the rest of the epochs. We report the average performance as well as standard deviations using 5 random seeds. We train our framework on Nvidia RTX 3090. In addition, we use Adam (Kingma and Ba, 2015) as an optimizer.

Moreover, Table 6 shows the detailed hyperparameter configuration. From these results, we find that the appropriate values of hyperparameters are not entirely consistent for different datasets. This is because the data distribution of datasets of differ-

ent types is complex and inconsistent. Hence, we should comprehensively consider various situations and carefully tune the value of hyperparameters according to actual datasets.

The pseudo-code of our proposed framework, Loki, is presented in Algorithm 1.

4.3 Main Results

Table 1 reports the performance of different methods for synthetic datasets (see Appendix D for more datasets’ results). Our key observations are as follows: (1) As shown in Table 1, our method consistently outperforms all baselines with varying noise types and ratios. (2) For real-world datasets, they often contain a more intricate mix of noises and are more difficult to separate than synthetic noise (Zhuang et al., 2023). However, as shown in Table 2, our method still performs well, which again confirms its effectiveness. (3) We also provide the base model’s performance on the ground truth data (GT) and the performance of the LLM-base (ChatGPT with zero-shot). In some cases, Loki’s results even surpass GT. This is because Loki can generate additional correct samples for tail classes.

4.4 Ablation Studies and Discussions

In the following experiments, we analyze each component of the proposed Loki on R8 to verify its effectiveness (refer to Table 3).

Modules	R8			
Noise Setup	20%S	40%S	20%A	40%A
Loki	98.31	97.26	98.27	97.58
Loki w/o DCS	95.10	90.19	94.06	89.68
Loki w/o LKD	96.30	94.49	96.13	95.44
Loki w/o $\mathcal{L}_{HC}$	80.95	70.98	80.99	70.50
Loki w/o $\mathcal{L}_{TC}$	97.01	95.70	96.22	95.72
Loki w/o $\mathcal{L}_O$	96.94	96.00	96.26	95.50
Loki w/o $\mathcal{L}_E$	96.52	94.97	96.36	95.44

Table 3: Ablation study on the R8 dataset.

Effectiveness of Divide-and-Conquer sample Selection strategy (DCS). DCS provides different separation metrics for head and tail classes, which can solve the problem of *Tail indivisibility* during training. Meanwhile, DCS also provides tail information for subsequent knowledge distillation. If we remove DCS and use the small-loss strategy on all classes for sample selection in a unified way without head-tail class division, Loki’s performance

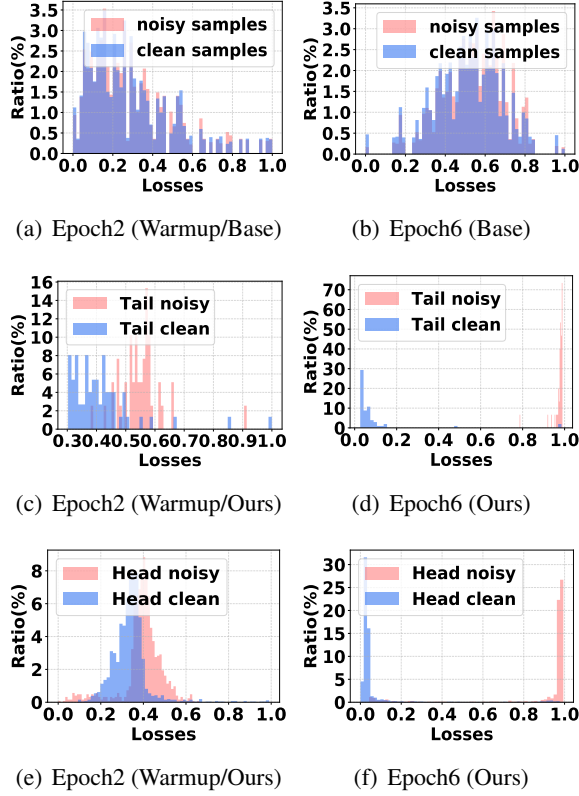


Figure 4: The loss distributions of base (a-b) and ours (c-f) on R8 under 40% asymmetric noise during training.

will decrease by a large margin, which proves DCS contributes significantly to performance improvement.

Effectiveness of LLM knowledge distillation based on tail data (LKD). Loki leverages LKD to generate new samples and pseudo-labels for tail classes, which can help student models learn more high-quality features of tail classes. The performance of Loki will decrease greatly when we remove the LKD. These results indicate that enhancing the tail class feature representation is crucial, particularly in scenarios with long-tail distribution.

Effectiveness of different learning objects. To validate the effectiveness of the different learning objects, we report the test accuracy by removing each object ($\mathcal{L}_{HC}$, $\mathcal{L}_{TC}$, $\mathcal{L}_O$, $\mathcal{L}_E$) respectively. As can be seen, failure to employ them leads to a decline in the results. From the overall results, the HC contributes the most to performance improvement.

The ability to select clean samples and noisy samples. To validate the advantage of the proposed method Loki, we visualize clean and noisy samples' loss distributions of the base model and Loki on R8 under 40% asymmetric noise (refer to Fig. 4).

For the base model, it can be observed that the CE values of noise and clean samples highly overlap during training (refer to Fig. 4 (a-b)), which leads to failure sample separation. For Loki, the loss values of the head and tail classes are becoming increasingly polarized. As shown in Fig.4 (c-f), most clean samples of head classes and tail classes are concentrated in the interval of $[0.8, 1]$, which is easily clustered by GMM. In this case, Loki can better isolate noisy and clean samples from head classes and tail classes, which can help student models better perform training on these selected clean samples and avoid overfitting noisy samples. Furthermore, we present a more direct demonstration of Loki's ability to sample selection. Specifically, we report the number of clean samples in each subset during training. From Table 4, we observe the following: (1) The number of clean samples in HC gradually increases; (2) Clean samples in TC are completely identified by Loki, validating the effectiveness of our proposed splitting metric, WCE; (3) After using LLMs to regenerate pseudo-labels for samples in TN , the number of clean samples in O significantly increases, which verifies the effectiveness of querying LLMs for label denoising.

5 Conclusion

This paper handles a realistic while challenging text classification problem: learning with noisy labels on long-tailed data. We propose a novel knowledge distillation framework named Loki that leverages the teacher LLM to generate new samples and pseudo-labels of tail classes, which can help the student model learn more high-quality tail class features for learning with noisy labels on long-tailed textual data. During the distillation process, the major challenge is that it is hard to recognize tail classes and distinguish clean samples from noisy samples on tail classes. Accordingly, we introduce a Divide-and-Conquer sample Selection strategy that performs head-tail class division and uses distinct separation metrics for head and tail classes to address the problem of sample selection in tail classes. Extensive experiments demonstrate our methods can help improve the robustness against noisy labels and long-tailed data simultaneously.

Limitations

Our proposed Loki is a knowledge distillation framework that aims to handle label noise on class-imbalanced datasets for multi-class text classifica-

Dataset	Subsets	epoch3	epoch4	epoch5	epoch6
Trec	head clean set (HC)	4074/4115 (99.00%)	4091/4132 (99.00%)	4126/4168 (99.00%)	4181/4222 (99.03%)
	tail clean set (TC)	48/50 (96.00%)	54/55 (98.18%)	58/58 (1.0%)	62/62 (1.0%)
	tail noisy set (TN)	30/202 (14.85%)	26/215 (12.09%)	25/213 (11.74%)	25/213 (11.74%)
	$\mapsto$ corrected set (O)	$\mapsto$ 176/202 (87.12%)	$\mapsto$ 188/215 (87.44%)	$\mapsto$ 201/213 (94.37%)	$\mapsto$ 208/213 (97.65%)

Table 4: The data statistical distribution (the number of samples with right labels / the number of all samples) of different subsets on the Trec dataset under 20%A. The value (%) represents the ratio of correct labels in the subset.

tion. Despite its effectiveness, there is still much potential for improvement: (1) **Computational cost during training:** Although Loki results in a more efficient student model for inference, the training process itself can be computationally expensive due to the iterative generation of samples from the LLM. This may limit the scalability of the approach for vast datasets or frequent model updates. (2) **Exploration of Different Task:** Our efforts have been focused on the text classification task, aligning with previous studies (Qiao et al., 2022; Wang et al., 2023; Zhuang et al., 2023; Yuan et al., 2024). Therefore, Loki’s applicability to different tasks remains under-explored in this work. We believe that Loki is expected to work well within other tasks, especially in generation tasks. We leave the exploration of this direction as promising future work.

Ethics Statement

We adhere to the ACL Code of Ethics. All datasets and language models used in this study are publicly available. The code and related materials will be appropriately released to ensure transparency and reproducibility of our work.

Funding

Scientific Research Project of National University of Defense Technology under Grant 25-ZZCX-KXKY-11.

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.
- Jae Soon Baik, In Young Yoon, Kun Hoon Kim, and Jun Won Choi. 2024. [Distribution-aware robust learning from long-tailed data with noisy labels](#). *CoRR*, abs/2407.16802.
- Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. 2018. [A systematic study of the class imbalance problem in convolutional neural networks](#). *Neural Networks*, 106:249–259.
- Jie Chen, Yupeng Zhang, Bingning Wang, Xin Zhao, Ji-Rong Wen, and Weipeng Chen. 2024. [Unveiling the flaws: Exploring imperfections in synthetic data and mitigation strategies for large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 14855–14865. Association for Computational Linguistics.
- 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.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. [Simcse: Simple contrastive learning of sentence embeddings](#). 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 6894–6910. Association for Computational Linguistics.
- 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.
- Chen Huang and Guoxiu He. 2024. [Text clustering as classification with llms](#). *CoRR*, abs/2410.00927.
- 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.
- Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. 2020. [Decoupling representation and classifier for long-tailed recognition](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Yeanchan Kim, Junho Kim, and SangKeun Lee. 2024. [Towards robust and generalized parameter-efficient fine-tuning for noisy label learning](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 5922–5936. Association for Computational Linguistics.
- 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*.
- Martin Krallinger, Obdulia Rabal, Saber A Akhondi, Martín Pérez Pérez, Jesús Santamaría, Gael Pérez Rodríguez, Georgios Tsatsaronis, Ander Intxaurrenondo, 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 net-news](#). 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.
- Mengyu Li, Yonghao Liu, Fausto Giunchiglia, Xiaoyue Feng, and Renchu Guan. 2024a. [Simple-sampling and hard-mixup with prototypes to re-balance contrastive learning for text classification](#). *CoRR*, abs/2405.11524.
- 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*.
- Zhuo Li, He Zhao, Zhen Li, Tongliang Liu, Dandan Guo, and Xiang Wan. 2024b. [Extracting clean and balanced subset for noisy long-tailed classification](#). *CoRR*, abs/2404.06795.
- Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. [Synthetic data generation with large language models for text classification: Potential and limitations](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 10443–10461. Association for Computational Linguistics.
- Hong Liu, Jeff Z. HaoChen, Adrien Gaidon, and Tengyu Ma. 2022. [Self-supervised learning is more robust to dataset imbalance](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. 2021. [Long-tail learning via logit adjustment](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- 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.
- Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. 2020. [Balanced meta-softmax for long-tailed visual recognition](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- 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.
- Linxin Song, Jieyu Zhang, Tianxiang Yang, and Masayuki Goto. 2022. [Adaptive ranking-based sample selection for weakly supervised class-imbalanced text classification](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 1641–1655. Association for Computational Linguistics.

- 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.
- Chaozheng Wang, Shuzheng Gao, Pengyun Wang, Cuiyun Gao, Wenjie Pei, Lujia Pan, and Zenglin Xu. 2024a. [Label-aware distribution calibration for long-tailed classification](#). *IEEE Trans. Neural Networks Learn. Syst.*, 35(5):6963–6975.
- Ke Wang, Jiahui Zhu, Minjie Ren, Zeming Liu, Shiwei Li, Zongye Zhang, Chenkai Zhang, Xiaoyu Wu, Qiqi Zhan, Qingjie Liu, and Yunhong Wang. 2024b. [A survey on data synthesis and augmentation for large language models](#). *CoRR*, abs/2410.12896.
- Leyao Wang, Yu Wang, Bo Ni, Yuying Zhao, and Tyler Derr. 2024c. [Large language model-based augmentation for imbalanced node classification on text-attributed graphs](#). *CoRR*, abs/2410.16882.
- 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.
- Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. 2019. [Symmetric cross entropy for robust learning with noisy labels](#). In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 322–330. IEEE.
- Tong Wei, Jiang-Xin Shi, Yufeng Li, and Min-Ling Zhang. 2022. [Prototypical classifier for robust class-imbalanced learning](#). In *Advances in Knowledge Discovery and Data Mining - 26th Pacific-Asia Conference, PAKDD 2022, Chengdu, China, May 16-19, 2022, Proceedings, Part II*, volume 13281 of *Lecture Notes in Computer Science*, pages 44–57. Springer.
- Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. 2018. [Feature generating networks for zero-shot learning](#). In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 5542–5551. Computer Vision Foundation / IEEE Computer Society.
- Shuo Yang, Lu Liu, and Min Xu. 2021. [Free lunch for few-shot learning: Distribution calibration](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Xuanyu Yi, Kaihua Tang, Xian-Sheng Hua, Joo-Hwee Lim, and Hanwang Zhang. 2022. [Identifying hard noise in long-tailed sample distribution](#). In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVI*, volume 13686 of *Lecture Notes in Computer Science*, pages 739–756. Springer.
- Bo Yuan, Yulin Chen, Yin Zhang, and Wei Jiang. 2024. [Hide and seek in noise labels: Noise-robust collaborative active learning with llms-powered assistance](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 10977–11011. Association for Computational Linguistics.
- Manyi Zhang, Yuxin Ren, Zihao Wang, and Chun Yuan. 2022. [Tackling instance-dependent label noise with dynamic distribution calibration](#). In *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022*, pages 4635–4644. ACM.
- Manyi Zhang, Xuyang Zhao, Jun Yao, Chun Yuan, and Weiran Huang. 2023. [When noisy labels meet long tail dilemmas: A representation calibration method](#). In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 15844–15854. IEEE.
- 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 Theoretical Analysis

A.1 Proof for upper limit of CE difference

Proof. Let x_i and x_j be two samples from a tail class k , with p_i^k and p_j^k representing the k -th dimension of their prediction confidences $\mathbf{P}_i = [p_i^1, \dots, p_i^K]$ and $\mathbf{P}_j = [p_j^1, \dots, p_j^K]$, respectively. Their commonly observed class label $\tilde{\mathbf{y}} = [\tilde{y}_i^1, \dots, \tilde{y}_i^K]$ is in the one-hot form where only the value of the k -th dimension is 1 ($\tilde{y}_i^k = 1$), and the values on other dimensions are all 0 ($\tilde{y}_i^m = 0$ for all $m \neq k$). Cross entropy (CE) for a sample x_i is defined as:

$$\begin{aligned} CE(x_i) &= - \sum_{d=1}^{K-1} \tilde{y}_i^d \log p_i^d \\ &= - \sum_{d \neq k} \tilde{y}_i^d \log p_i^d - \tilde{y}_i^k \log p_i^k \\ &= -\log p_i^k \end{aligned} \quad (4)$$

Then

$$|CE(x_i) - CE(x_j)| = |-\log p_i^k - (-\log p_j^k)| \quad (5)$$

Now, for $u \in (0, 1)$, let

$$f(u) = -\log u \quad (6)$$

The first-order derivative and second-order derivative of $f(u)$ can be obtained by:

$$f'(u) = -1/u \quad (7)$$

$$f''(u) = 1/u^2 \quad (8)$$

It can be observed that $f'(u) < 0$ and $f''(u) > 0$ that makes $f'(u)$ monotonically increase and $|f'(u)|$ monotonically decrease. By the Lagrange mean value theorem, if function $f(u)$ is continuous and differentiable on the interval $(0, 1)$, then there is at least one point ε between two real numbers $a, b \in (0, 1)$:

$$|f(a) - f(b)| = |f'(\varepsilon)| * |a - b|. \quad (9)$$

As $\varepsilon > a$ and $|f'(u)|$ monotonically decrease, we have:

$$\begin{aligned} |f'(\varepsilon)| * |a - b| &\leq |f'(a)| * |a - b| \\ &= |-1/a| * |a - b| \\ &= |a - b|/a \end{aligned} \quad (10)$$

By replacing $f(u)$ with $CE(x)$, a with p_i^k and b with p_j^k , we have:

$$|CE(x_i) - CE(x_j)| \leq |p_i^k - p_j^k|/p_i^k \quad (11)$$

A.2 Proof for benefits of enhancing tail data

We give a theoretical analysis to show the benefit of enhancing tail data. We begin by formally presenting some model assumptions of the tuples $\{(f(x_i), \tilde{y}_i, y_i)\}_i^n$, where $f(x_i)$ is the feature representation of x_i , y_i is the true label, and $\tilde{y}_i$ is the possibly incorrect label. Note that y_i is unobserved. For theoretical simplicity, we assume that $n_k = n_{tail}$ for each $k \in \mathcal{D}_{tail}$ and $n_k = n_{head}$ for each $k \in \mathcal{D}_{head}$.

Assumption A.2. (1) In this paper, we assume that the feature representations of training data in each class conform to a multivariate Gaussian distribution. The assumption is mild and has been verified by existing works (Xian et al., 2018; Yang et al., 2021; Zhang et al., 2022). That is, for K classes, we define K multivariate Gaussian distributions. The distribution belonging to the k -th class is denoted by $\mathcal{N}(f(x)|\mu_k, \Sigma_k)$ with $\mu_k \in \mathbb{R}^m$ and $\Sigma_k \in \mathbb{R}^{m \times m}$, where m denotes the dimension of the feature representations. For simplicity, we assume that all classes have the same covariance matrix Σ . Then, the $f(x) \in \mathbb{R}^m$ and the corresponding true label y satisfies $P(y = k) = n_k/n$ and $f(x)|y = k \sim \mathcal{N}(\mu_k, \Sigma)$. For the Gaussian model, the Bayes optimal classifier on top of $f(x_i)$ is well-known to be Fisher's linear discriminant, which is defined as

$$h_*(f(x)) = \arg \max_{k \in [K]} \left\{ \log \left(\frac{n_k}{n} \right) + \mu_k^\top \Sigma^{-1} f(x) - \frac{1}{2} \mu_k^\top \Sigma^{-1} \mu_k \right\}. \quad (12)$$

In practice, the true mean μ is unknown and needs to be estimated from data, whose estimation error directly affects the corresponding classification error. Therefore, the following Theorem A.2 gives the upper limit of the difference between the vanilla empirical mean $\tilde{\mu}$ and the true mean μ .

(2) There is a constant $\rho' > 0$ such that given the true label $y_i = k$, the possibly incorrect label $\tilde{y}_i$ satisfies $P(\tilde{y}_i = j|y = k) = \rho' \times n_j/n$, for $j \neq k$. Actually, the ρ' is the noise rate, which measures the degree of label noise. The label mislabeling probability is assumed to be proportional to the sample size, based on the intuition that labels are more likely to be misclassified into classes with a larger sample size.

Theorem A.2. Under Assumption A.2, there exists a constant $\mathbb{C}$ such that:

$$\mathbb{E} \|\tilde{\mu}_k - \mu_k\|^2 \leq \mathbb{C} \times [\rho'^2 + m/n_{tail}] \quad (13)$$

The term ρl is caused by the label noise in the training data and is inevitable. The last terms are variance terms caused by the finite sample issue. Since the sample sizes of tail classes are relatively small, the variance term is dominated by m/n_{tail} and could be very large. *That is to say, for long-tailed noisy tasks, increasing the sample sizes of tail classes helps to obtain better feature representations.*

Proof. In the following, we provide a proof of Theorem A.2.

For $P(\tilde{y}_i = j | y = k)$, which can be calculated by:

$$P(\tilde{y}_i = j | y = k) = \frac{P(\tilde{y}_i = j)P(y = k | \tilde{y}_i = j)}{P(y = k)}. \quad (14)$$

Further, we can also infer that $P(\tilde{y}_i = j | y = k) = P(y = j | \tilde{y}_i = k)$. Therefore,

$$\begin{aligned} \mathbb{E}\tilde{\mu}_k &= \mathbb{E}[f(x_i) | \tilde{y} = k] = \sum_{j=1}^K P(y = j | \tilde{y} = k) \mu_j \\ &= (1 - \rho l + \frac{n_k}{n} \rho l) \mu_k + \sum_{j \neq k} \frac{n_j \rho l}{n} \mu_j. \end{aligned} \quad (15)$$

$$\begin{aligned} \mathbb{E}\|\tilde{\mu}_k - \mu_k\|^2 &= \mathbb{E}\|\tilde{\mu}_k - \mathbb{E}\tilde{\mu}_k\|^2 + \|\mathbb{E}\tilde{\mu}_k - \mu_k\|^2 \\ &\leq \mathbb{E}\|\sum_{j \neq k} \frac{n_j \rho l}{n} (\mu_j - \mu_k)\|^2 + \frac{m \sigma^2}{n} \\ &\leq \mathbb{C} \left(\rho l^2 + \frac{m}{n} \right). \end{aligned} \quad (16)$$

This finishes the proof.

B Baseline 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;
- **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 inaccuracy 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.
- **NoiseAL** (Yuan et al., 2024) focuses on solving the problem of text classification with noisy labels. It introduces active learning to combine the non-pretrained model, PLMs, and ChatGPT for learning from noisy labels.
- **SharpReCL** (Yi et al., 2022) is the first to focus on imbalanced text classification tasks with supervised contrastive learning (SCL), which employs simple sampling and hard-mixup techniques to generate a balanced sample set for SCL to mitigate the class imbalance problem.
- **LADC** (Wang et al., 2024a) has exploited the transferable knowledge from head classes to tail classes for calibrating the distribution of tail classes.
- **LWS** (Kang et al., 2020), decouples the learning procedure into representation learning and classification, then systematically explores how different balancing strategies affect them for long-tailed recognition.
- **BSCE** (Ren et al., 2020) applies a complementary meta sampler to estimate the optimal class sample rate and further improve long-tailed learning.
- **ChatGPT**. In addition to utilizing ChatGPT’s zero-shot capabilities, we also evaluated its

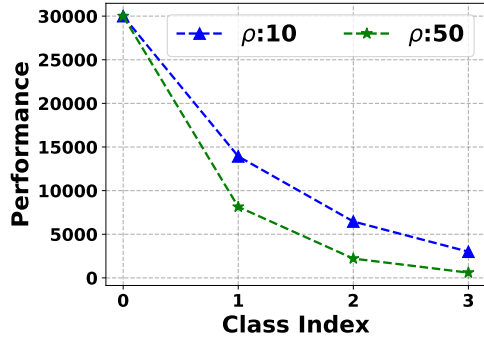


Figure 5: The class distribution of AGNews under different imbalance factors ρ .

few-shot potential (random sampling) in our experiments. For both ChatGPT and Loki, we uniformly configured the setup to a 5-shot setting in our experiments to ensure a fair comparison.

C Dataset Details

In this section, we introduce the details of the datasets used in our experiments. Specifically, we first select **20ng** and **AGNews** to construct the **synthetic imbalanced datasets** with different imbalance factors ρ (*i.e.*, 10 and 50). Figure 5 presents the sample distribution of synthetic AGNews under different imbalance factors. Among them, 20ng and AGNews are news topic classification datasets. Then, we select two **known 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 4 datasets, we follow the existing work (Zhuang et al., 2023) to manually inject three different types of noise (**Symmetric noise**, **Asymmetric noise** and **Instance-dependent noise**) into them for evaluation in our experiments. The noise rate is chosen in $\{0.2, 0.4\}$.

Real-world datasets. To further verify the effectiveness of our method in practical scenarios, we conduct experiments on known 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 previous 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
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.

For these baselines, we perform their public code or corresponding algorithms reported in the paper to implement them.

D More detailed Results

In Table 7, we also report the detailed performance (average accuracy) on Trec, 20ng with $\rho=10$, and 20ng with $\rho=50$.

E More Analysis

E.1 Computational Cost Analysis

We compute the relative training cost of each method compared to the baseline on Trec, as presented in the Table 8. By comparison, it is evident that the training cost of the proposed method is indeed higher but falls within an acceptable range. Despite this higher cost, the proposed method offers remarkable performance gains in return. Although our method may require more time to train the model, the trained model does not incur additional costs during the inference process.

E.2 Hyperparameters Selection

Sensitivity to clustering parameters. In our methods, S is a main hyperparameter to divide the tail classes. We select S from $[0, 1, \dots, K]$, where K is the number of categories in the corresponding dataset. To see how S affects the final performance of Loki. The experimental results of adjusting the value of parameter S are shown in Figure 6. The results show that performance first improves and then degrades as S increases, with the best performance achieved at $S=1$. This trend is consistent with the underlying data distribution: tail classes constitute a minority, and treating all classes as tail classes leads to suboptimal performance. This suggests that a moderate head-tail division is important.

Hyperparameter	Trec	20ng ($\rho:10 / \rho:50$)	AGNews ($\rho:10 / \rho:50$)	R8	TREC	ChemProt	SemEval
S	1	3 / 5	1 / 2	2	1	2	3
φ	0.1	0.2 / 0.3	0.1 / 0.2	0.2	0.2	0.3	0.1
ξ	0.1	0.3 / 0.2	0.1 / 0.2	0.2	0.1	0.1	0.1
Batch Size	32	32	32	32	32	32	32
learning rate	1e-5	1e-5	1e-5	1e-5	1e-5	1e-5	1e-5

Table 6: Main hyperparameter settings of our models.

Dataset	Trec				20ng ($\rho:10$)						20ng ($\rho:50$)					
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
GT	97.40				83.07						70.12					
LLM-base	61.60				69.33						69.33					
BERT-base	94.64	87.45	93.60	85.72	77.24	71.87	71.43	58.37	74.96	67.50	68.84	59.24	64.10	45.77	64.63	56.86
SelfMix	95.20	89.80	95.16	89.00	71.74	67.59	69.55	61.37	64.45	57.30	50.96	46.46	53.19	45.80	53.16	48.67
DyGen	95.88	89.00	94.96	88.56	77.51	74.32	<u>76.28</u>	<u>71.55</u>	77.06	73.66	63.82	<u>61.54</u>	62.86	57.59	64.34	60.90
LAFT	95.00	88.68	94.28	87.52	77.17	72.86	70.62	59.06	72.34	64.16	67.28	59.41	59.77	43.66	63.25	55.61
NoiseAL	<u>97.16</u>	<u>96.40</u>	<u>96.80</u>	<u>95.80</u>	78.76	<u>75.29</u>	74.15	70.01	<u>78.72</u>	<u>74.81</u>	65.00	52.56	65.15	<u>59.24</u>	65.08	56.32
LADC	95.37	88.44	94.21	87.39	78.62	73.58	72.10	59.18	76.53	68.44	<u>70.40</u>	59.82	64.98	45.99	65.43	58.29
LWS	96.20	88.28	93.91	87.54	<u>79.02</u>	72.14	72.08	58.54	75.44	68.74	69.58	60.47	64.68	47.73	65.39	58.11
BSCE	96.31	89.05	94.81	87.34	78.06	73.41	71.49	60.13	75.07	68.17	68.86	60.74	<u>65.99</u>	46.97	<u>65.74</u>	57.95
SharpReCL	96.47	89.28	94.73	86.66	78.68	72.21	72.30	59.08	75.73	68.69	68.97	60.28	<u>64.73</u>	47.20	65.26	<u>58.67</u>
Ours	97.33	96.96	97.36	97.02	82.85	79.75	82.63	77.83	82.66	79.31	69.66	65.90	69.41	65.01	69.63	65.11

Table 7: Performance (average accuracy %) of Loki and other baselines on synthetic datasets with varying label noise and imbalance ratios. All experiments are repeated 5 times. The best results and second-best results are highlighted in **Bold** and underlined, respectively. GT refers to the base model trained on ground truth data (0% noise), and LLM-base refers to directly using LLMs (ChatGPT).

Method	SelfMix	DyGen	LAFT	NoiseAL	LADC	LWS	BSCE	SharpReCL	Ours
Relative Cost	$\times 3.70$	$\times 3.26$	$\times 4.22$	$\times 3.88$	$\times 2.55$	$\times 3.10$	$\times 2.85$	$\times 2.82$	$\times 3.95$

Table 8: Relative training cost compared to the baseline.

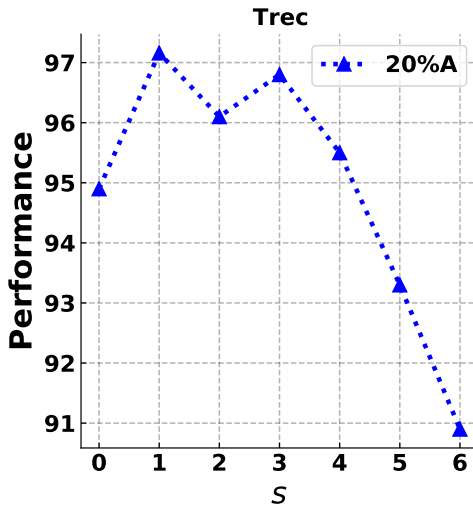


Figure 6: Performance of our Loki on the Trec dataset under 20% asymmetric label noise when S is ranging from 0 to 1.

Sensitivity to the number of generated tail-class samples. We further conduct additional ab-

Generated Tail samples	Accuracy (%)
8	92.82
16	94.10
24	96.35
32	97.36
40	97.33
48	97.35
56	97.36
64	97.36

Table 9: Sensitivity to the number of generated tail-class samples, based on the experiments on the Trec dataset under 20%A.

lation experiments by varying the number of generated tail-class samples. As shown in Figure 9, the results show that increasing the number of generated samples initially improves performance by alleviating data imbalance, while the gains gradually saturate beyond a certain point. Importantly, the performance remains stable under moderate variations, indicating that our method is not overly sensitive to the exact number of generated samples.

F Prompt Details

F.1 Prompts Structure

This section reports the prompts used for tail classes in this work:

(1) Prompts for teacher models to generate new samples. The teacher prompt template is shown in Figure 7.

(2) Prompts for teacher models to generate pseudo labels. The teacher prompt template is shown in Figure 8 and consists of the following three components:

a. **System prompt**, which describes the specific task.

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

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):

b. **Examples**, which consist of clean samples under each tail class demonstrating the task. The format is as follows: "tail class: clean samples of tail class".

c. **Input**, which is the test text from the tail noisy set TN to classify.

In our experiments, we adopt 3-shot examples in all prompts to perform in-context learning for fair comparisons.

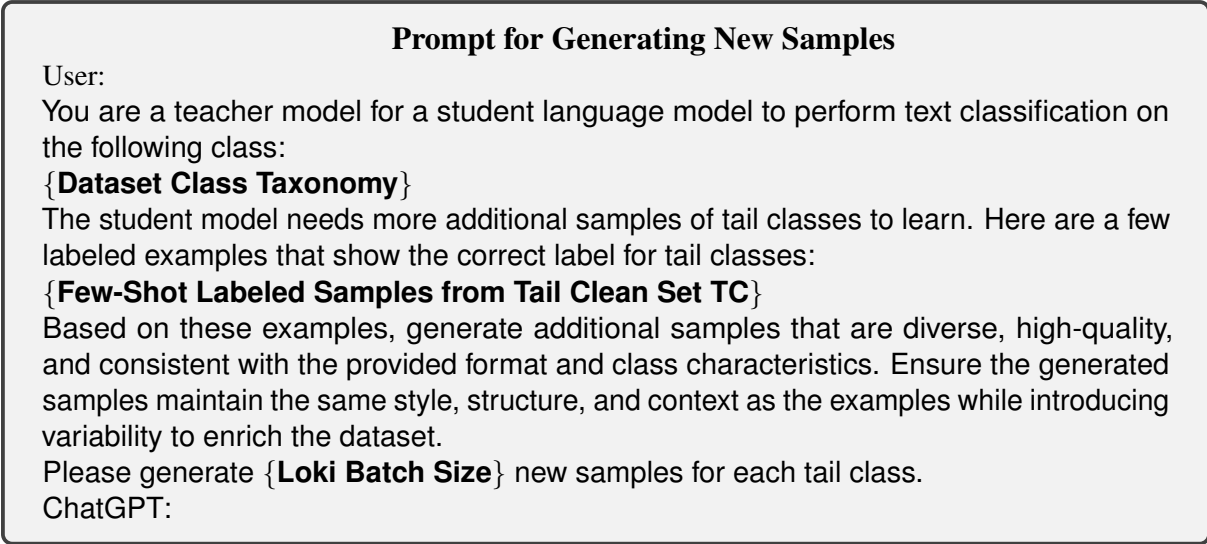


Figure 7: The prompt instruction to generate new samples for tail classes.

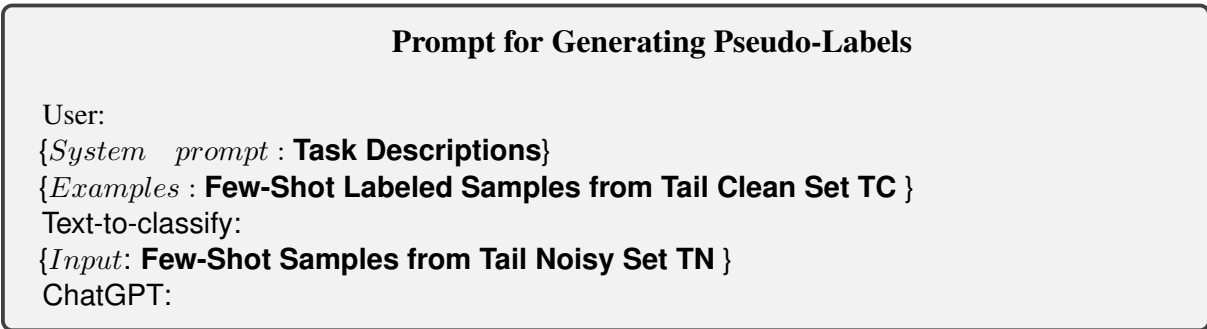


Figure 8: The prompt instruction to generate pseudo-labels for tail classes.

Algorithm 1 The training process of the proposed framework Loki

Require: $\mathcal{D}=\{(x_i, \tilde{y}_i)\}_{i=1}^N$, $\tilde{y}_i \in [1, \mathcal{K}]$, student model (SLM), teacher model (LLM), warmup epochs T_w , max epochs T , the pre-training epochs T_u for unsupervised contrastive learning, prompt

- 1: **For** $t=1, \dots, T_u$ **do**
- 2: Pre-train the encoder of SLM via unsupervised SimCSE (Gao et al., 2021)
- 3: **end for**;
- 4: Extract deep representations (embeddings) of samples
- 5: Apply the clustering algorithm (K-means) to embeddings of all samples
- 6: Set epoch = 1; Initialize the SLM f_θ trained on $\mathcal{D}$
- 7: // Step 1: Warm up
- 8: **while** epoch $\leq T_w$ **do**
- 9: Warmup SLM f_θ ;
- 10: epoch = epoch + 1;
- 11: **end while**
- 12: **while** epoch $\leq T$ **do**
- 13: // Step 2: Divide-and-Conquer Sample Selection
- 14: Rank all clusters $[|\mathcal{D}_{cluster_1}|, \dots, |\mathcal{D}_{cluster_K}|]$ and obtain head class $\mathcal{D}_{head}$ and tail class $\mathcal{D}_{tail}$ by Eq.(1) and Eq.(2);
- 15: Calculate the CE for all samples in $\mathcal{D}_{head}$;
- 16: Apply GMM with CE and ξ to divide the $\mathcal{D}_{head}$ into HC and HN ;
- 17: Calculate the WCE for all samples in $\mathcal{D}_{tail}$
- 18: Apply GMM with WCE and φ to divide the $\mathcal{D}_{tail}$ into TC and TN ;
- 19: // Step 3: Knowledge Distillation Based on Tail Data
- 20: Obtain corrected sets O by leveraging LLM to generate pseudo-labels: $O \leftarrow \text{LLM}(TN, \text{prompt})$
- 21: Obtain extended sets E by leveraging LLM to generate new samples: $E \leftarrow \text{LLM}(TC, \text{prompt})$
- 22: // Step 4: Robustly training
- 23: Update the SLM f_θ by robustly training with different loss functions based on the different subsets (HC, TC, E, O)
- 24: epoch = epoch + 1
- 25: **end while**
