GRIP: Gated Routing of Instance-level Preferences for Noisy Long-Tailed Text Classification

Anonymous ACL submission

Abstract

Noisy long-tailed text classification is challenging because label noise and class imbalance simultaneously degrade representation learning and decision boundaries. Existing methods typically address either noisy labels or long-tail distributions, or rely on a single teacher that cannot adapt to different sample characteristics. In this paper, we propose **GRIP** (*Gated Routing of Instance-level Preferences*), a dual-teacher dual-gate framework for selective knowledge distillation. Specifically, we introduce a noise-specialized teacher and a tail-specialized teacher, together with two independent gates—trained with a reward-shaped weighted binary cross-entropy objective—that determine whether each teacher should be activated for a given sample. The student is then trained with selectively injected distillation signals. Experimental results on benchmark text classification datasets demonstrate the effectiveness of GRIP. Further analysis shows that the learned gates exhibit meaningful behavior with respect to noisy and long-tail samples.

1 Introduction

Pre-trained language models have achieved strong performance on text classification, but their success often depends on large amounts of accurately labeled data. In practice, text labels are frequently collected through crowdsourcing or automatic pipelines, which introduces label noise. At the same time, real-world text datasets are rarely balanced and often follow a long-tailed distribution, where a few frequent classes dominate the training set while many rare classes contain only a small number of samples. These two problems do not occur in isolation. Noisy labels can mislead the model and distort its decision boundaries, while long-tailed distributions make rare classes harder to learn because they receive much weaker supervision. When they appear together, minority classes become even more fragile, since they are affected by both limited data and corrupted labels. As a result, noisy long-tailed text classification is much more difficult than handling either problem alone, and it raises a practical question of how different types of difficult samples should be treated during training.

Existing methods have made strong progress on long-tailed learning and noisy label learning, but they are still mostly developed for one problem at a time, and their assumptions weaken once class imbalance and label noise co-occur. The fundamental difficulty is that the two problems are entangled: a tail sample can exhibit a high loss simply because it is under-represented, while a noisy sample can produce the same signal because its label is wrong, so the same training behaviour admits two different diagnoses. As a consequence, simply combining a long-tail technique with a noise-handling technique is often not enough—boosting tail samples also amplifies mislabeled rare examples, and suppressing suspicious samples can mistakenly discard informative tail instances. The two objectives therefore conflict with each other when applied uniformly.

The same limitation manifests in standard teacher–student distillation: forcing all samples to follow the same teacher signal mismatches the supervision to the sample condition, since suspicious samples need a noise-robust teacher while rare but clean samples need a tail-aware one. The key question is therefore not only how to build better teachers, but also *which teacher should guide which sample*.

To address this challenge, we start from a simple idea: different difficult samples should not be forced to learn from the same teacher signal. Instead, the model should choose a more suitable source of guidance according to the current state of each sample. Based on this intuition, we propose **GRIP** (*Gated Routing of Instance-level Preferences*), a dual-teacher dual-gate framework for noisy long-tailed text classification. GRIP intro-

duces two specialized teachers with different roles. The noise teacher is trained to be more robust to corrupted labels and is expected to provide more reliable supervision for suspicious or unreliable samples. The tail teacher is trained with a long-tail-oriented objective and is expected to provide better guidance for rare or under-represented classes. To make sample-level routing possible, we further introduce two independent gate networks. The noise gate estimates whether a sample is more likely to be noisy or unreliable, while the tail gate estimates whether a sample is more likely to benefit from tail-aware supervision. Based on the outputs of these two gates, the student selectively receives distillation signals from the corresponding teachers, rather than following a single teacher for all training instances. In this way, teacher supervision becomes adaptive to sample condition, which helps reduce the conflict between noise handling and tail learning and allows the student to learn from more suitable guidance under different sample scenarios.

The main contributions of this work are summarized as follows:

- We study noisy long-tailed text classification from the perspective of sample-specific supervision and show that noise handling and tail learning should not be treated as a single uniform problem. We argue that different difficult samples may require different teacher signals during training.
- We propose **GRIP**, a dual-teacher dual-gate framework for noisy long-tailed text classification. GRIP introduces a noise teacher for unreliable samples, a tail teacher for rare classes, and two independent gate networks that decide, at the sample level, whether each teacher should be activated.
- We design a selective distillation and gate learning strategy that allows the student to receive adaptive teacher guidance under different sample conditions. Experimental results show that the proposed method is effective and that the learned gate behavior is consistent with the intended roles of the two teachers.

2 Related Work

Learning With Noisy Labels. LNL methods broadly fall into two families. The first designs noise-robust losses, such as Generalized Cross Entropy (GCE) (Zhang and Sabuncu, 2018) and Symmetric Cross Entropy (SCE) (Wang et al., 2019), which inherently tolerate corrupted annotations. The second performs sample selection based on the small-loss criterion, often using multiple networks to avoid confirmation bias—e.g., Co-Teaching (Han et al., 2018), JoCoR (Wang et al., 2021), and DivideMix (Li et al., 2020); for text classification, SelfMix (Qiao et al., 2022) combines semi-supervised learning with feature-level mixing. These methods all assume that large-loss samples are mislabeled. This assumption breaks under class imbalance, where clean tail-class samples are also high-loss and are mistakenly discarded as noise, hurting minority-class performance.

Class Imbalance. Long-tailed learning addresses datasets where head classes dominate. Classical solutions include re-sampling head/tail classes (Chawla et al., 2002) and re-weighted objectives that penalize tail errors more heavily, such as Focal Loss (Lin et al., 2017), LDAM (Cao et al., 2019), and Balanced Softmax (Ren et al., 2020). Decoupled training (Kang et al., 2020) further shows that learning the representation and the classifier separately substantially improves tail accuracy. These methods all assume clean supervision; once labels are corrupted, their re-weighting and margin terms amplify the influence of mislabeled tail examples and become unreliable.

Knowledge Distillation and Adaptive Gating. Knowledge Distillation (Hinton et al., 2015) transfers knowledge from a teacher to a lightweight student. Multi-teacher KD aggregates several teachers' logits via averaging or fixed weights (You et al., 2017), but applies the *same* aggregated supervision to every sample, ignoring how a sample's reliability or rarity might call for different teachers. Mixture-of-Experts gating (Shazeer et al., 2017; Fedus et al., 2022) routes *tokens* to expert FFNs, primarily for computational efficiency rather than supervision quality. In contrast, we introduce *sample-level* adaptive gating tailored to selective distillation: each sample is routed to a noise-robust or a tail-aware teacher based on its current learning state, directly resolving the conflicting objectives that uniform multi-teacher KD cannot.

3 Methodology

3.1 Framework Overview

GRIP contains one student classifier, one noise teacher, one tail teacher, and two independent gate networks. This architecture is explicitly designed

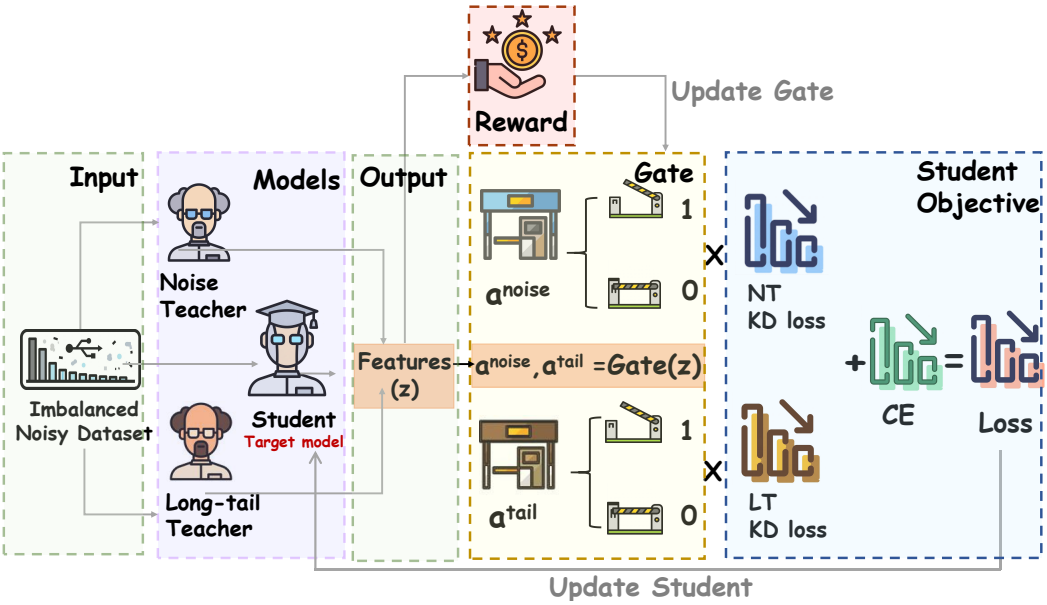


Figure 1: The overall framework of GRIP. A student model is trained on imbalanced noisy data with guidance from a noise teacher and a tail teacher. Two gate networks take sample-level features as input and decide whether each teacher should provide knowledge distillation for the current sample. The final training objective combines the selected KD losses with the cross-entropy loss, while reward signals are used to optimize the gates.

to resolve the conflicting objectives discussed in the Introduction: rather than forcing a single model to simultaneously amplify rare patterns and filter corrupted labels—which often leads to inconsistent guidance—we decouple these two capabilities into specialized teachers. The student is the final model used for prediction. The two teachers are fixed during student training and provide different kinds of knowledge. The noise teacher is designed to be more suitable for noisy or unreliable samples, while the tail teacher is designed to provide better guidance for rare or tail-class samples. The two independent gates act as a sample-level router to disentangle the coupling of noisy and long-tailed data.

For each input text, the student and the two teachers first produce their own predictions. We then build two groups of sample-level features that summarize the current learning state: a compact 2-dimensional input for the noise gate (an AUM-style margin score and the student prediction margin) and a richer 10-dimensional input for the tail gate that combines student-side, teacher-side, and class-frequency signals (Section 3.3; full list in Appendix B.2). Each gate then independently outputs an activation probability that decides, at the sample level, whether its corresponding teacher contributes a distillation signal to the student. The student, the two gates, and the two teachers are organized so that the student is the only model used at inference time; the rest of this section describes the four training-time components in turn.

3.2 Student and Teacher Models

We deliberately use *two* specialized teachers rather than a single noise-robust one. As we show empirically in Appendix A.1, a single loss-driven teacher cannot simultaneously distinguish noisy samples from hard tail samples on coupled noisy long-tailed data: the loss signals of the two populations overlap, so any uniform supervision built on top of them propagates conflicting gradients to the student. Decoupling the two objectives into a noise teacher and a tail teacher removes this confound, and the two gates then route each sample to whichever teacher is best positioned to help. All three models share the same BERT (Devlin et al., 2019) encoder and a linear classification head on the [CLS] representation; their specialization comes entirely from different training objectives.

Student Classifier. The student is a standard BERT classifier optimized with the cross-entropy loss on the given (possibly noisy) labels and the selective distillation losses described in §3.4. It is the only model used at inference time.

Tail Teacher. The tail teacher is trained with focal loss (Lin et al., 2017), which down-weights easy samples and amplifies the gradient on hard ones.

Because tail-class samples are typically harder under standard cross-entropy, this objective shifts more learning capacity onto minority classes and yields decision boundaries that are less dominated by head classes. The exact form of the focal objective is given in Appendix B.4.

Noise Teacher. The noise teacher is trained with SelfMix (Qiao et al., 2022), which uses the per-sample training loss to separate likely-clean from likely-noisy instances, replaces the labels of the noisy partition with sharpened model predictions, and combines feature-level Mixup on `[CLS]` embeddings with dropout-based consistency regularization to suppress overfitting to corrupted supervision. The full pipeline is detailed in Appendix B.1.

3.3 Noise Gate and Tail Gate

To decide when each teacher should be used, we introduce two independent gate networks: a noise gate and a tail gate. Both consume sample-level features derived from the current prediction states of the student and the two teachers (rather than raw text), so the gates only need a lightweight MLP head and focus on each sample's learning condition. We deliberately keep the two gates separated, because mixing label-reliability signals with class-distribution signals into a single decision tends to weaken the specialization of either teacher.

Noise Gate. For each sample x_i the noise gate consumes a compact 2-dimensional feature vector z_i^{noise}: the AUM-style margin score (gap between the logit of the given label and the largest competing logit) and the student prediction margin (gap between the student's top-1 and top-2 predictions). Both features track whether the given label is confidently defended by the student; small or negative values indicate hard, unstable, or unreliable samples. The noise gate maps z_i^{noise} to an activation probability

$$g_i^{noise} = G_{noise}(z_i^{noise}) \in [0, 1],$$

where a larger value means the sample is more likely to benefit from the noise teacher.

Tail Gate. The tail gate consumes a richer 10-dimensional feature vector z_i^{tail} that combines student-side signals (entropy, margin, loss, label probability), student–teacher interaction signals (two KL divergences, tail-teacher confidence, agreement), and class-level signals (cross-teacher agreement and the rarity score of the predicted class). Together these features describe whether a sample is uncertain, difficult, rare, or relatively better explained by the tail teacher. The full list and the rationale behind the three groups are given in Appendix B.2. The tail gate predicts

$$g_i^{tail} = G_{tail}(z_i^{tail}) \in [0, 1],$$

where a larger value means the sample is more suitable for tail-aware supervision.

3.4 Selective Distillation for Student

The student is trained with two kinds of supervision: the standard cross-entropy loss on the given (possibly noisy) labels and the gated KD losses from the two teachers. Let $f_s(x_i)$, $f_n(x_i)$, and $f_t(x_i)$ denote the logits of the student, the noise teacher, and the tail teacher for sample x_i. The classification loss is

$$\mathcal{L}_{cls} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{CE}(f_s(x_i), \tilde{y}_i), \tag{1}$$

where N is the batch size and $\tilde{y}_i$ is the given label. The teacher distillation loss for sample x_i on teacher $\xi \in \{n, t\}$ is the KL divergence between the temperature-softened predictive distributions of the student and the teacher,

$$\mathcal{L}_{kd,i}^{\xi} = \mathrm{KL}(\sigma(f_s(x_i)/T) \parallel \sigma(f_\xi(x_i)/T)), \tag{2}$$

where $\sigma(\cdot)$ is the softmax and T the distillation temperature. The gates produce sample-level activation probabilities g_i^{noise}, g_i^{tail}, which we threshold into binary decisions

$$\hat{g}_i^{\xi} = \mathbb{I}(g_i^{\xi} > \tau_\xi), \quad \xi \in \{n, t\}, \tag{3}$$

with $\mathbb{I}(\cdot)$ the indicator function. The selective KD loss over a mini-batch and the final student objective are

$$\mathcal{L}_{kd} = \frac{1}{N} \sum_{i=1}^{N} \Big(\lambda_n \hat{g}_i^{noise} \mathcal{L}_{kd,i}^{n} + \lambda_t \hat{g}_i^{tail} \mathcal{L}_{kd,i}^{t} \Big), \tag{4}$$

$$\mathcal{L}_{student} = \mathcal{L}_{cls} + \mathcal{L}_{kd}, \tag{5}$$

where λ_n and λ_t weight the two distillation terms. With this design, samples that the noise gate flags as unreliable receive correction from the noise teacher, while samples that the tail gate flags as tail-related receive guidance from the tail teacher; samples that neither gate opens are supervised only by the standard cross-entropy term.

3.5 Gate Training Objective

We train each gate with a *reward-shaped weighted BCE loss*: a per-sample reward is computed from the current student/teacher states, ranked within each mini-batch into a high/middle/low group, mapped to a soft target, and used to supervise the gate's activation probability. The formulation is motivated by—but is *not* a Monte-Carlo estimator of—a policy-gradient objective; the implemented loss is supervised classification with reward-derived targets, and the policy view is treated as a derivation in Appendix B.5. Below we describe the three components of the surrogate: raw rewards, reward-shaped targets, and the BCE loss.

Raw rewards. The two gates are trained with *asymmetric* rewards, because "noisy" and "tail-related" are fundamentally different concepts; sharing the reward would push both gates toward the same routing policy. For the noise gate we use two reliability-related signals—the normalized negative AUM-style score $\hat{u}_i$ and the normalized student–noise-teacher KL divergence $\hat{d}_i^{sn}$:

$$r_i^{noise} = \omega_1 \hat{u}_i + \omega_2 \hat{d}_i^{sn}. \tag{6}$$

Intuitively, the AUM margin captures whether the given label is steadily defended during training (an established indicator of label corruption), while the student–teacher divergence measures how informative the noise teacher's correction would be on this sample. Because of the second term, the noise gate answers a different question from SelfMix's internal loss-GMM (*is this distillation step worth taking?* vs. *is the label likely correct?*); we verify empirically in Appendix G.3 that the two are not redundant.

For the tail gate we use five signals organized along three orthogonal axes: **class distribution** ($\hat{\rho}_i$, normalized rarity score), **teacher capability** ($\hat{p}_i^{tail}$, tail-teacher confidence on the given label; $\hat{a}_i^{tail}$, tail-teacher's relative advantage over the noise teacher), and **student state** ($\hat{c}_i^{stu}$, low-confidence score; $\hat{h}_i$, predictive entropy):

$$r_i^{tail} = \eta_1 \hat{\rho}_i + \eta_2 \hat{p}_i^{tail} + \eta_3 \hat{c}_i^{stu} + \eta_4 \hat{a}_i^{tail} + \eta_5 \hat{h}_i. \tag{7}$$

This three-axis design ensures that the tail gate activates only when the sample is plausibly tail-related, the tail teacher can be trusted on it, and the student genuinely needs help, which prevents the tail gate from collapsing into a redundant copy of the noise gate. A detailed discussion of why each signal is chosen is given in Appendix B.3.

Reward-shaped targets. Within each mini-batch, samples are ranked by their raw reward r_i^{gate} and partitioned into a high-reward group, a middle group, and a low-reward group. The training target is set piecewise:

$$y_i^{gate} = \begin{cases} 0.95, & \text{high-reward}, \\ 0.50, & \text{middle}, \\ 0.05, & \text{low-reward}, \end{cases} \tag{8}$$

and we assign a confidence weight w_i^{gate} that is large on the high/low groups and small on the middle one. This focuses learning on the most confident positive and negative cases while keeping uncertain cases weakly constrained.

Surrogate loss. Each gate is then optimized with a confidence-weighted BCE loss against the reward-shaped target,

$$\mathcal{L}_{gate}^{\xi} = \frac{1}{N} \sum_{i=1}^{N} w_i^{\xi} \cdot \mathrm{BCE}\Big(g_i^{\xi},\, y_i^{\xi}\Big), \tag{9}$$

for $\xi \in \{noise, tail\}$, and the total gate objective is

$$\mathcal{L}_{gate} = \mathcal{L}_{gate}^{noise} + \mathcal{L}_{gate}^{tail}. \tag{10}$$

Although Eq. (9) is not a direct Monte-Carlo REINFORCE estimator, it follows the same policy-learning intuition—increase the activation probability on high-reward samples and decrease it on low-reward ones—while removing the variance introduced by action sampling. The full derivation is given in Appendix B.5.

3.6 Joint Optimization

We optimize the student and the two gates in an *alternating* fashion within each mini-batch: the student is updated under Eq. (5) with both teachers and gates frozen, and the two gates are then updated under Eq. (10) with the student and teachers frozen. The two stages do not share gradients—the indicator in Eq. (3) is non-differentiable, so the student treats $\hat{g}_i^{\xi}$ as a stop-gradient mask while the gates are trained *only* by the reward-shaped BCE objective; the two objectives couple solely through the per-sample features that drive the rewards. The full procedure is summarized as Algorithm 1 in Appendix B.6; at test time only the student is kept, so the inference cost matches that of a standard BERT-based classifier.

Dataset	Trec				AGNews (IF:10)						AGNews (IF:50)					
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	85.97	69.81	83.79	65.08	88.78	73.37	78.51	69.11	77.15	61.11	86.25	65.15
Co-Teaching	95.08	89.30	94.88	87.16	89.51	86.55	89.34	84.01	88.99	82.89	84.04	80.38	84.64	78.22	87.21	65.86
CL	95.64	89.72	95.52	86.24	90.58	88.04	89.51	83.52	88.76	78.15	85.51	82.99	83.24	74.58	87.53	69.72
CR	95.15	89.74	94.87	87.77	89.98	88.07	89.96	79.06	88.40	77.81	83.71	82.12	82.93	73.75	86.98	66.74
DyGen	95.88	89.00	94.96	88.56	87.26	86.19	87.07	85.26	87.64	77.42	84.20	79.71	83.95	78.53	86.31	69.58
SelfMix	95.20	89.80	95.16	89.00	87.56	86.83	87.55	86.56	86.09	77.78	80.80	78.87	80.07	78.18	85.43	69.41
NoiseAL	96.20	89.68	95.58	88.35	90.59	87.19	90.78	86.31	90.43	78.79	83.06	81.28	84.21	79.01	88.99	70.24
Supervised GT (0% Noise)	97.20				91.87						88.96					
ChatGPT (Zero-shot/10-shot)	61.60/72.00				82.92/84.23						82.92/84.23					
GRIP (Ours)	**97.15**	**96.74**	**97.18**	**95.56**	**91.66**	**91.17**	**91.73**	**91.04**	**91.74**	**90.49**	**87.75**	**87.51**	**88.30**	**87.35**	**89.10**	**87.82**

Table 1: Performance (accuracy %) comparison of GRIP with other LNL baselines on synthetic noise datasets. We also compare GRIP with the zero-shot/10-shot and supervised counterparts on the test dataset; Supervised GT refers to BERT trained on ground truth data. All numbers are averaged over 5 random seeds; per-cell standard deviations are listed in the appendix detailed result tables. **Bold** indicates the best score per column.

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	$65.53\pm_{0.22}$	$68.33\pm_{0.31}$	$71.11\pm_{1.07}$
NoiseAL	$70.98\pm_{0.46}$	$78.84\pm_{1.82}$	$81.47\pm_{0.62}$
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	$\mathbf{70.75}\pm_{0.87}$	$\mathbf{88.60}\pm_{1.25}$	$\mathbf{83.17}\pm_{0.27}$

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

Variant	NG	TG	NT	TT	RS	TREC	ChemProt
GRIP (Full)	✔	✔	✔	✔	✔	**88.60**	**70.75**
w/o Noise Gate	✗	✔	✔	✔	✔	86.15	69.45
w/o Tail Gate	✔	✗	✔	✔	✔	84.40	68.90
w/o Noise Teacher	✗	✔	✗	✔	✔	74.50	67.20
w/o Tail Teacher	✔	✗	✔	✗	✔	81.85	67.85
w/o RS Training	✔	✔	✔	✔	✗	87.20	70.20

Table 3: Ablation study on TREC and ChemProt datasets. NG: Noise Gate; TG: Tail Gate; NT: Noise Teacher; TT: Tail Teacher; RS: Reward-Shaped gate training. **Bold** indicates best performance.

4 Experiments

4.1 Baselines

We compare our method with the following three types of approaches: (1) LNL methods: **Co-Teaching** (Han et al., 2018), **CR** (Zhou and Chen, 2021), **CL** (Northcutt et al., 2021), **SelfMix** (Qiao et al., 2022), **LAFT** (Wang et al., 2023), **DyGen** (Zhuang et al., 2023), **NoiseAL** (Yuan et al., 2024); (2) Long-tail learning methods: **LADC** (Wang et al., 2024), **SharpReCL** (Yi et al., 2022), **LWS** (Kang et al., 2020), **BSCE** (Ren et al., 2020); (3) Methods aiming at tackling noisy label and long-tail distribution: **TABASCO** (Lu et al., 2023), **RCAL** (Zhang et al., 2023), **RoLT** (Kang et al., 2020), **HAR** (Ren et al., 2020). See Appendix D for baseline details. In the main text, we primarily present the results compared with LNL baselines, while additional results are provided in Table 6 and Table 7 of Appendix E.1

The implementation details are in Appendix F.

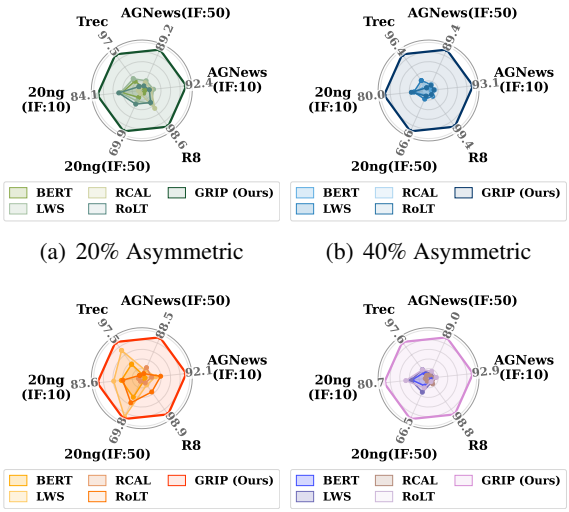
4.2 Datasets

We evaluate on two regimes. (i) **Real-world noisy** datasets: **ChemProt** (Krallinger et al., 2017), **SemEval** (Zhou et al., 2020), and **TREC** (Awasthi et al., 2020). (ii) **Long-tailed corpora with synthetic noise**: the inherently imbalanced **Trec** (Li and Roth, 2002) and **R8** (Lewis, 1992), plus synthetic-imbalance versions (IF $\in \{10, 50\}$, see Appendix C.1) of **AGNews** (Gulli, 2005), **MR** (Pang and Lee, 2005), and **20ng** (Lang, 1995). On the synthetic-imbalance corpora we further inject three standard noise types—**Symmetric (S)** (van Rooyen et al., 2015), **Asymmetric (A)** (Chen et al., 2019; Zhu et al., 2022),

and **Instance-dependent (I)** (Algan and Ulusoy, 2020)—following existing protocols (Qiao et al., 2022; Zhu et al., 2022). Per-dataset statistics and the formal definitions of each noise type are deferred to Appendix C.

4.3 Main Results

Results on Trec (synthetic noise plus realistic long-tailed distribution) and AGNews (synthetic noise plus synthetic long-tailed distribution) are reported in Table 1; results on real-world noisy datasets are reported in Table 2. Per-dataset breakdowns and additional comparisons are provided in Appendix E.



(a) 20% Asymmetric (b) 40% Asymmetric

(c) 20% Symmetric (d) 40% Symmetric

Figure 4: Radar charts of accuracy across datasets under four noise settings: 20% asymmetric (a), 40% asymmetric (b), 20% symmetric (c), and 40% symmetric (d). Each axis represents a dataset, and each colored polygon corresponds to a different method.

(1) GRIP outperforms strong baselines across noise types and intensities. As visualized in Fig. 4(a)–(d), the radar contour of GRIP stays consistently outside those of all baselines under 20%/40% asymmetric and 20%/40% symmetric noise, across six datasets covering both synthetic and natural long-tailed distributions. Improvements remain robust as the noise level increases from 20% to 40%, where competing methods degrade more sharply, indicating that decoupling noise robustness and tail awareness is especially valuable in the harder regime.

(2) Gate quality and student performance improve together. Fig. 2 traces the training dynamics on TREC under 40% asymmetric noise: both gate metrics rise as training proceeds (the tail gate's Accuracy first dips for the reason discussed in Appendix G.1), and the student's Accuracy / Macro-Recall / Macro-F1 follow the same trajectory and peak around the same epoch, confirming that better routing translates directly into better knowledge transfer.

Method	TREC (real)	R8 (40%A)
GPT-4o-mini zero-shot	71.2	89.4
GPT-4o-mini 10-shot	80.3	92.5
GPT-4o-mini relabel → BERT	83.5	94.2
GRIP (Ours)	**88.60**	**97.40**

Table 4: GRIP vs. the strongest LLM (GPT-4o-mini) under three usage modes. GRIP exceeds LLM-as-labeler by $+5.1\%/+3.2\%$ and direct 10-shot ICL by $+8.3\%/+4.9\%$ on TREC / R8 (40%A). Per-LLM breakdown in Appendix G.5.

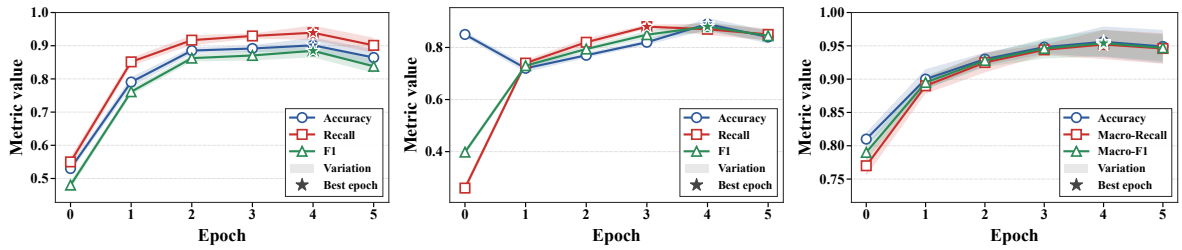
4.4 Comparison with Modern LLM Baselines

To address the concern that a fine-tuned BERT framework may be unnecessary in the LLM era, we compare GRIP against three modern LLM usage modes on TREC (real-world noise, 38.56%) and R8 + 40%A: (i) **zero-shot ICL**, (ii) **10-shot ICL** with uniformly-sampled demonstrations, and (iii) **LLM-as-labeler**, where GPT-4o-mini re-annotates the noisy training set and a standard BERT-base is then trained on the re-annotated data. Table 4 reports the strongest LLM (GPT-4o-mini) per mode; the full per-LLM comparison across LLaMA-3.1-8B-Instruct, Qwen2.5-7B-Instruct, and GPT-4o-mini is provided in Appendix G.5.

The LLM-as-labeler pipeline addresses only label corruption: GPT-4o-mini's residual relabel error is reinjected as fresh noise, and the long-tail bias remains uncorrected. GRIP closes both gaps through dual-teacher routing—*class-conditional supervision on top of denoising*—which LLM-relabel cannot replicate; extended discussion is in Appendix G.5.

4.5 Ablation Studies

We ablate each key component while keeping the rest of GRIP unchanged (Table 3); per-component discussion is in Appendix G.2. Three patterns stand out. (i) Removing the *noise teacher* causes the largest single drop on TREC ($88.60 \rightarrow 74.50$, -14.1%), reflecting TREC's high noise ratio (38.56%). (ii) Tail-side components contribute comparably to or more than noise-side ones once



(a) Noise Gate (b) Tail Gate (c) Student

Figure 2: Training dynamics on TREC under 40% asymmetric noise. (a) Noise gate Accuracy, Recall, and F1 over epochs. (b) Tail gate Accuracy, Recall, and F1 over epochs. (c) Student Accuracy, Macro-Recall, and Macro-F1 over epochs. Shaded regions indicate variation ranges across runs.

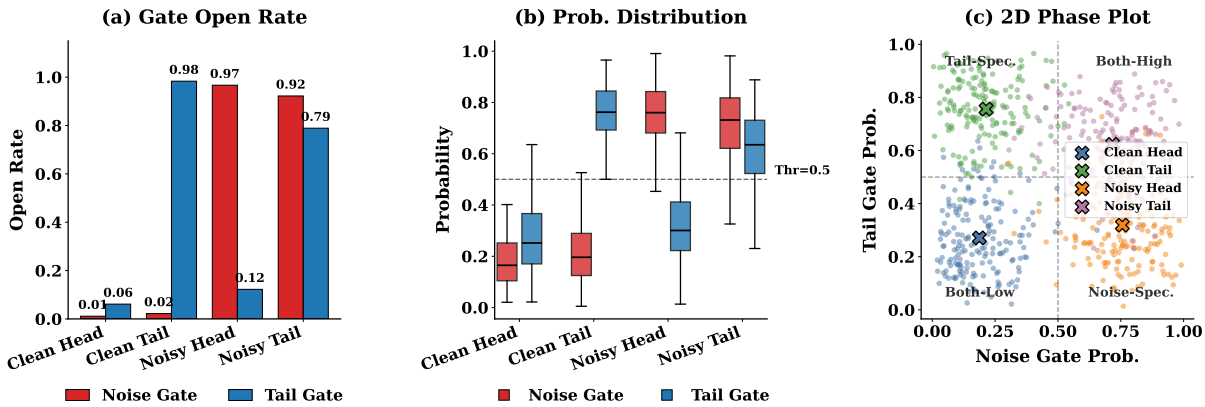


Figure 3: Complementary routing of the two gates on TREC across the four sample groups (Clean/Noisy $\times$ Head/Tail). (a) Per-group open rate; (b) activation-probability distribution (dashed: Thr $= 0.5$); (c) 2D phase plot with group centroids. The three views together show that the two gates learn specialized, non-redundant roles.

the dominant noise effect is discounted (w/o Tail Gate 84.40/68.90 vs. w/o Noise Gate 86.15/69.45), indicating that *tail-aware supervision is at least as critical as noise-robust supervision* on coupled data. (iii) Replacing the reward-shaped gate objective with a fixed-threshold rule degrades both datasets, showing that the reward-shaped BCE surrogate is not cosmetic. Overall, the largest gains come from *jointly* routing each sample to the right teacher.

4.6 Analysis

Fig. 3 shows that the two gates learn clearly distinct routing behaviors rather than redundancies of a single difficulty signal: the noise gate fires almost exclusively on noisy samples while the tail gate concentrates on tail samples, and the 2D phase plot places the four sample groups in four distinct quadrants. A more detailed reading of the gate-open-rate breakdown is provided in Appendix G.4.

5 Conclusion

We study noisy long-tailed text classification and argue that its difficulty stems from a *coupling effect*: noisy and tail-class samples both manifest as high-loss instances, so a single uniform teacher mixes the two and yields conflicting supervision. We propose GRIP, a dual-teacher dual-gate framework that decouples the two objectives into a noise-robust SelfMix teacher and a tail-aware focal teacher, and routes each sample through two lightweight gates trained with a reward-shaped supervised objective using asymmetric signal sets. Experiments on synthetic and real-world noisy long-tailed benchmarks show that GRIP consistently outperforms strong LNL and long-tail baselines, with the learned gates developing clearly complementary roles. Under coupled noise and imbalance, *who teaches which sample* matters as much as how strong each teacher is.

Limitations

Our proposed GRIP handles label noise on class-imbalanced datasets through a dual-teacher dual-gate framework, but several aspects remain open for further investigation. First, GRIP relies on two specialized teachers (a noise-robust teacher and a tail-aware teacher) trained beforehand; the student's final performance is therefore upper-bounded by the quality of the two teachers, and a poorly trained teacher on extremely scarce or domain-specific tail classes can still mislead the gates. Second, the gate features and reward signals are designed for classification logits and class-frequency statistics; extending the same routing principle to structured prediction or generation tasks would require re-designing the feature set and the reward shaping. Third, our experiments focus on text classification benchmarks; broader validation on multilingual, multi-modal, or extremely large-scale long-tailed datasets is left as promising future work.

Ethics Statement

We adhere to the ACL Code of Ethics. This paper does not pose foreseeable ethical problems or negative social consequences: the datasets used in our experiments are all publicly available and widely adopted by the community for evaluating noisy and long-tailed text classification, and our framework introduces no additional collection of human-subject data.

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.

Kaidi Cao, Yining Chen, Junwei Lu, Nikos Arechiga, Adrien Gaidon, and Tengyu Ma. 2020. Heteroskedastic and imbalanced deep learning with adaptive regularization. *arXiv preprint arXiv:2006.15766*.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. 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 1565–1576.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

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.

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.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

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.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

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.

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, Martın Pérez Pérez, Jesús Santamaría, Gael Pérez Rodríguez, and 1 others. 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.

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*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

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.

Yang Lu, Yiliang Zhang, Bo Han, Yiu-Ming Cheung, and Hanzi Wang. 2023. Label-noise learning with intrinsically long-tailed data. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 1369–1378. IEEE.

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.

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*.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

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.

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. 2024. Label-aware distribution calibration for long-tailed classification. *IEEE Trans. Neural Networks Learn. Syst.*, 35(5):6963–6975.

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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 322–330.

Zhenyu Wang, Ya-Li Li, Ye Guo, and Shengjin Wang. 2021. Combating noise: semi-supervised learning by region uncertainty quantification. *Advances in Neural Information Processing Systems*, 34:9534–9545.

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

Tong Wei, Jiang-Xin Shi, Wei-Wei Tu, and Yu-Feng Li. 2021. Robust long-tailed learning under label noise. *arXiv preprint arXiv:2108.11569*.

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.

Shan You, Chang Xu, Chao Xu, and Dacheng Tao. 2017. Learning from multiple teacher networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1285–1294.

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, 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.

Zhilu Zhang and Mert Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31.

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.

Appendix

To facilitate a clearer understanding of our contributions, we provide an organized appendix containing the following sections:

- Analysis
  - Analysis for the Sources Leading to Selection Bias
- Methodology Details
  - SelfMix Procedure for the Noise Teacher
  - Tail Gate Input Features
  - Reward Design Justification
  - Tail Teacher Training Objective
  - Policy View and Surrogate Derivation of the Gate Objective
  - Joint Optimization Algorithm
- Dataset Details
  - Synthetic Noise Generation
- Baselines Details
- More Detailed Results
  - Comparison with Long-tailed Learning Methods and Long-tailed Noisy-Label Learning Methods
- Implementation Details
- Additional Experimental Analysis
  - Why Tail-Gate Accuracy Drops First While Recall Keeps Increasing

A Analysis

A.1 Why a Single Loss-Based Teacher Fails Under Coupled Noise and Imbalance

A natural baseline for noisy long-tailed text classification is a single noise-robust teacher trained with the small-loss criterion, such as SelfMix (Qiao et al., 2022). The criterion follows the memory effect (Han et al., 2018; Shu et al., 2019; Qiao et al., 2022): networks fit clean samples earlier than noisy ones, so clean samples tend to have a smaller mean loss and a global loss threshold can separate the two populations. This works well on roughly balanced data, but our empirical analysis shows that the same loss signal is no longer sufficient when class imbalance and label noise are coupled—a single loss-based teacher signal cannot simultaneously distinguish noisy samples from hard tail samples.

Specifically, we inject synthetic label noise (40% asymmetric and 40% symmetric) into the inherently imbalanced R8 dataset, and observe that tail-class samples are routinely under-fit and accumulate large loss values that overlap with the noisy partition (Figure 5). As a consequence, SelfMix's loss-based assumption breaks down: a low-loss tail sample looks indistinguishable from a high-loss head-noisy sample, and any uniform supervision built on top of this single signal will inevitably conflate the two.

To make this concrete, we monitor R8 training under 40% asymmetric label noise at a fine-grained level (Table 8 reports the per-class statistics) and observe that:

- the loss distribution of clean samples and noisy samples, whether in the head class or tail class, evolves dynamically during training;
- both tail-class samples and noisy samples exhibit large losses;
- the losses of some clean tail samples are even larger than the losses of some noisy head samples;
- for the head class, clean samples tend to have a smaller loss while noisy samples tend to have a bigger loss;
- for the tail class, the trend is reversed—noisy samples often have a smaller loss while clean samples have a bigger loss.

These observations explain why a single loss-driven teacher is insufficient under coupled noise and class imbalance: any global criterion built on top of the loss signal will inevitably mix clean tail samples with noisy ones, and applying its supervision uniformly to all samples therefore propagates conflicting signals to the student. This motivates our dual-teacher dual-gate design: a noise-robust teacher remains responsible for label-reliability denoising on samples that genuinely look noisy, while a tail-aware teacher specializes in rare classes that the noise teacher is not equipped to recognize, and two independent gates route each sample to the teacher whose supervision is most appropriate. Decoupling the two objectives is therefore the key to escaping the failure mode discussed above, and is also why the tail reward (Eq. (7)) cannot rely on a loss/margin signal alone but explicitly aggregates rarity, teacher capability, and student state.

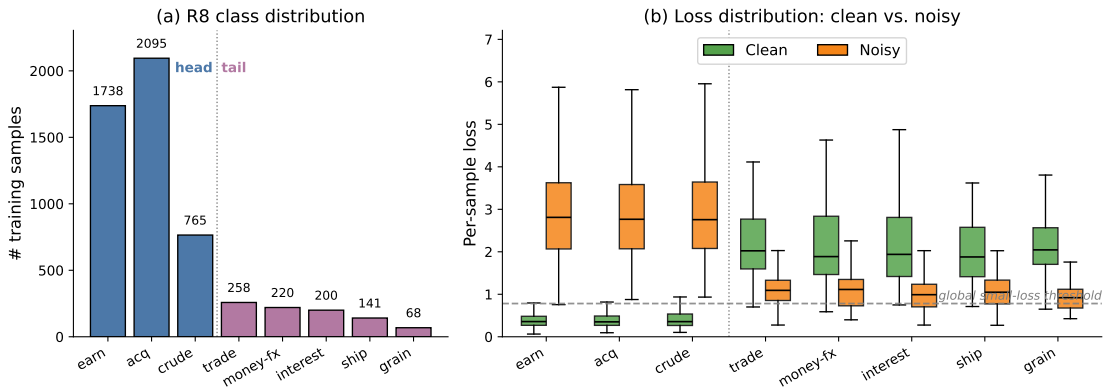


Figure 5: Class distribution of R8 (left) and per-class loss distribution split into clean/noisy (right) under 40% asymmetric label noise. The dashed gray line on the right marks a global small-loss threshold. For head classes (*earn*, *acq*, *crude*), the standard memory-effect ordering holds—clean samples have small loss, noisy ones have large loss. For tail classes the trend is reversed: clean tail samples accumulate large loss values that exceed those of head-noisy samples, while some noisy tail samples are accidentally fit and produce small losses. As a result, no horizontal threshold can separate clean from noisy at the same time across head and tail, demonstrating why a single loss-driven teacher is insufficient under coupled noise and class imbalance.

B Methodology Details

This section provides the technical details that are summarized in the main text, including the SelfMix training procedure of the noise teacher, the full input feature list of the tail gate, the design rationale of the two raw rewards, and the joint optimization algorithm.

B.1 SelfMix Procedure for the Noise Teacher

The noise teacher is trained with SelfMix (Qiao et al., 2022), whose training pipeline can be decomposed into the following five steps.

(1) Loss-based sample partitioning. SelfMix first relies on the memorization effect that neural networks fit clean and easy samples earlier than noisy ones, so clean samples typically exhibit smaller cross-entropy loss values. SelfMix fits a two-component Gaussian Mixture Model (GMM) on the per-sample loss distribution: low-loss samples are treated as clean and keep their original labels, while high-loss samples are treated as noisy and have their original labels removed.

(2) Soft pseudo-labeling for the noisy partition. For samples in the noisy partition, the model's own predictions are sharpened (with a low-temperature softmax) to produce more confident pseudo-labels. The original noisy-label problem is thereby converted into a semi-supervised one: clean samples carry hard labels and noisy samples carry sharpened soft labels.

(3) Feature-level mixing (EmbMix). Because text is discrete, SelfMix performs Mixup on `[CLS]` sentence embeddings instead of on raw inputs. Two embeddings are linearly interpolated and the corresponding labels (hard labels for clean samples, sharpened soft labels for noisy samples) are mixed accordingly. This enriches the training distribution and makes the model less sensitive to corrupted supervision.

(4) Dropout-based consistency regularization. SelfMix applies R-Drop–style consistency: the same input is forwarded twice through the network with different dropout masks, and the KL divergence between the two predictive distributions is minimized. This reduces confirmation bias without requiring an additional peer network.

(5) Architecture choice. In our framework the noise teacher uses the same BERT encoder as the student and the tail teacher; its specialization comes from the noise-robust training procedure above rather than from a different backbone, so it does not introduce additional parameters at inference time.

B.2 Tail Gate Input Features

The noise gate uses a compact 2-dimensional input (z_i^{noise}): the AUM-style margin score and the student prediction margin. The tail gate uses a richer 10-dimensional feature vector z_i^{tail} that captures the sample's state from three complementary perspectives:

Student state (4 features): (i) student entropy of

the predictive distribution; (ii) student top-1/top-2 margin; (iii) student cross-entropy loss with respect to the given label; (iv) student probability assigned to the given label.

Teacher–student interaction (4 features): (v) KL divergence between student and tail teacher; (vi) KL divergence between student and noise teacher; (vii) tail-teacher probability on the given label; (viii) student/tail-teacher agreement (top-1 prediction match).

Cross-teacher and class signal (2 features): (ix) tail-teacher/noise-teacher agreement; (x) rarity score of the class predicted by the tail teacher (computed from the class-frequency statistics of the training set).

All features are normalized to $[0, 1]$ within each mini-batch before being fed into the gate.

B.3 Reward Design Justification

Why two signals suffice for the noise gate. The noise reward $r_i^{noise} = \omega_1 \hat{u}_i + \omega_2 \hat{d}_i^{sn}$ aggregates only two components. The negative AUM-style score $\hat{u}_i$ captures the gap between the logit of the given label and the largest competing logit; mislabeled or ambiguous samples cannot maintain a stable confidence gap as training proceeds, so a small or negative margin is a well-established indicator of label corruption. The student–noise-teacher divergence $\hat{d}_i^{sn}$ measures how much the noise-robust teacher disagrees with the current student: a large disagreement indicates that the teacher offers a non-trivial correction signal that the student has not yet absorbed, making the sample a high-value distillation target. When both signals are small, the sample is already confidently handled by the student and additional pull toward the noise teacher would only introduce unnecessary perturbation. We deliberately keep this reward compact, since judging label reliability is essentially a binary problem and a small number of strong, weakly-correlated signals helps the gate avoid overfitting to spurious feature combinations.

Why five signals along three axes for the tail gate. Deciding whether a sample benefits from tail-aware supervision is multi-factor by nature, so the tail reward r_i^{tail} combines signals along three orthogonal axes.

(i) Class distribution. The rarity score $\hat{\rho}_i$ reflects whether the predicted class is under-represented, which is the most direct indicator of tail-related samples and grounds the reward in the global label frequency.

(ii) Teacher capability. The tail teacher's confidence on the given label $\hat{p}_i^{tail}$ measures whether the tail-specialized teacher actually has reliable knowledge of the sample; without this term the gate could activate the tail teacher even when the teacher itself is uncertain, propagating noisy distillation signals. The relative advantage $\hat{a}_i^{tail}$ further requires that the tail teacher is more competent on this sample than the noise teacher, which enforces a clear division of labor between the two gates and prevents the tail gate from collapsing into a redundant copy of the noise gate.

(iii) Student state. The student's low-confidence score $\hat{c}_i^{stu}$ and entropy $\hat{h}_i$ indicate that the student has not yet mastered the sample. Samples already confidently classified by the student should not be over-corrected by either teacher, whereas uncertain samples are exactly where additional supervision is most beneficial.

This three-axis design ensures that the tail gate activates only when (a) the sample is plausibly tail-related, (b) the tail teacher can be trusted on it, and (c) the student genuinely needs help, which is what enables the dual-gate framework to learn truly specialized routing policies rather than two redundant ones.

B.4 Tail Teacher Training Objective

The tail teacher is trained with focal loss (Lin et al., 2017) to mitigate the bias toward head classes. Given a sample $(x_i, \tilde{y}_i)$, let $p_i = \sigma(f_t(x_i))_{\tilde{y}_i}$ denote the probability that the tail teacher assigns to the given label. The focal loss is

$$\mathcal{L}_{tail}^{focal} = -\frac{1}{N} \sum_{i=1}^{N} (1 - p_i)^{\gamma} \log p_i, \qquad (11)$$

where $\gamma \geq 0$ is the focusing parameter (we use $\gamma = 2$). The factor $(1 - p_i)^{\gamma}$ down-weights samples that the model already classifies confidently and amplifies the loss of low-confidence ones; since tail-class samples are typically harder, focal training shifts more gradient onto them and yields decision boundaries that are less dominated by head classes. The tail teacher shares the BERT encoder and a linear classification head with the student, so its specialization comes entirely from this objective rather than from any architectural change.

B.5 Policy View and Surrogate Derivation of the Gate Objective

The reward-shaped weighted BCE loss used to train the gates can be motivated from a policy-learning view. For each gate, let $a_i^{gate} \in \{0, 1\}$ denote the routing action (1 means the corresponding teacher is activated for sample x_i). The gate's predicted activation probability g_i^{gate} defines a sample-conditional Bernoulli policy

$$\begin{aligned} \pi_{gate}(a_i^{gate}{=}1 \mid z_i^{gate}) &= g_i^{gate}, \\ \pi_{gate}(a_i^{gate}{=}0 \mid z_i^{gate}) &= 1 - g_i^{gate}, \end{aligned} \tag{12}$$

where $z_i^{gate} \in \{z_i^{noise}, z_i^{tail}\}$ is the gate-specific feature vector. The ideal objective is to maximize the expected reward of the routing action:

$$J^{gate} = \mathbb{E}_{a_i^{gate} \sim \pi_{gate}} \left[R_i^{gate}(a_i^{gate}) \right]. \tag{13}$$

The corresponding policy gradient takes the REINFORCE form

$$\begin{aligned} \nabla J^{gate} = \mathbb{E}\big[(R_i^{gate} - b_i) \\ \cdot \nabla \log \pi_{gate}(a_i^{gate} \mid z_i^{gate})\big], \end{aligned} \tag{14}$$

where b_i is a baseline used to reduce variance. Direct Monte-Carlo estimation of Eq. (14) with sampled binary actions, however, suffers from high variance and unstable gradients in the early training stage when the routing policy is still random.

The reward-shaped weighted BCE objective in Eq. (10) is a low-variance surrogate: the raw reward r_i^{gate} is converted into a soft target $y_i^{gate} \in \{0.05, 0.50, 0.95\}$ via piecewise mapping, and the BCE loss between g_i^{gate} and y_i^{gate} is weighted by a confidence weight w_i^{gate} that emphasizes the most informative high- and low-reward samples. This preserves the policy-learning intuition—increase the activation probability on high-reward samples and decrease it on low-reward ones—while removing the variance introduced by action sampling and yielding a smoother optimization landscape.

B.6 Joint Optimization Algorithm

Algorithm 1 summarizes the full training procedure. Within each mini-batch we first update the student parameters with the gates and teachers fixed, then update the two gates with the student and teachers fixed; the two stages share the same forward pass to avoid redundant computation. At test time only the student is kept, so the inference cost matches that of a single BERT-based classifier.

Algorithm 1 Joint Optimization of GRIP

```
Pre-train the noise teacher with SelfMix and the tail teacher with focal loss; freeze both.
Initialize the student model and the two gate networks.
for each training epoch do
    for each mini-batch B do
        Student update stage: fix the two teachers and the two gates.
        Compute student logits and teacher logits on B.
        Compute gate activations and selective KD losses.
        Update the student parameters via L_student = L_cls + L_kd.
        Gate update stage: fix the student and the two teachers.
        Recompute sample-level features z_i^noise, z_i^tail on B.
        Construct raw rewards r_i^noise, r_i^tail and reward-shaped piecewise targets.
        Update the noise gate and the tail gate via the weighted BCE objective.
    end for
end for
Inference: use only the student model.
```

C 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.*, 1, 10 and 50). Among them, 20ng and AGNews are news topic classification datasets, MR is a sentiment classification dataset. 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 C.2.

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.

C.1 Imbalance Construction

To simulate long-tailed distributions on top of originally balanced corpora, we assign each class an index $k \in [1, K]$ and downsample class k to $n_k = n_{\max} \cdot \mathrm{IF}^{-k/K}$, where the imbalance factor $\mathrm{IF} = \frac{n_{\max}}{n_{\min}}$ controls the head-to-tail ratio. We use $\mathrm{IF} \in \{10, 50\}$ throughout, applied to the originally balanced datasets **AGNews**, **MR**, and **20ng**; the inherently imbalanced **Trec** and **R8** retain their original distributions without further re-sampling.

C.2 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 2.

D 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;
- **CR** (Zhou and Chen, 2021), also trains multiple models with a regularization strategy based on a soft target.
- **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.
- **SelfMix** (Qiao et al., 2022) separates samples via GMM and leverages semi-supervised learning to handle label noise.

Algorithm 2 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 $(x_i, \bar{y}_i)_{i=1}^n$;

- **DyGen** (Zhuang et al., 2023) uses a variational auto-encoding framework to infer the posterior distributions of true labels from noisy labels for improved noisy-label prediction.

- **LAFT** (Wang et al., 2023) segregates training samples into different subsets by generating per-sample confidences with external LLM guidance, then trains the model with noise-robust losses on the partitioned data.

- **LADC** (Wang et al., 2024) proposes a label-aware distribution calibration method to handle the long-tailed recognition problem. It transfers the statistics from head classes to tail classes to calibrate the distribution of the tail classes. However, LADC heavily relies on the assumption that the training labels are accurate to estimate class statistics; in our noisy long-tailed setting, the presence of noisy labels distorts the estimated statistics and causes LADC to calibrate the distribution incorrectly.

- **SharpReCL** (Yi et al., 2022) introduces sharpness-aware minimization into contrastive learning to seek a flat loss landscape, aiming to improve the generalization ability on long-tailed distributions. Although this improves robustness against class imbalance, it does not explicitly handle label noise; optimizing for sharpness on noisy data can inadvertently lead the model to memorize noisy labels more robustly.

- **LWS** (Kang et al., 2020) decouples the learning of representations and classifiers, employing a learnable weight scaling (LWS) method to balance the classifier weights. However, LWS assumes that the provided labels are ground truth; when applied to noisy datasets, the learnable scaling factors are biased by the noisy labels, particularly in tail classes where samples are scarce.

- **BSCE** (Ren et al., 2020) theoretically derives a Balanced Softmax Cross-Entropy loss to accommodate the label shift between the training and test distributions. It adjusts the logits based on label frequencies to eliminate the bias towards head classes. In the presence of label noise, however, this assumption fails, and BSCE may aggressively fit noisy samples in tail classes.

- **TABASCO** (Lu et al., 2023) utilizes a two-stage bi-directional alignment strategy to align the distributions of different subsets in noise-robust long-tailed settings. It attempts to mitigate the impact of unreliable data through alignment and selection, but relies on internal model statistics (like consistency or confidence) which can be unreliable when the noise ratio is high or the backbone model is weak.

- **RCAL** (Zhang et al., 2023) focuses on robust calibration for long-tailed recognition, aiming to rectify the confidence bias of the model towards head classes via uncertainty/reliability estimation. However, when noise is asymmetric or heavy in tail classes, the calibration reference itself becomes corrupted.

- **RoLT** (Wei et al., 2021) aims to perform robust learning under long-tailed distributions, typically by distinguishing clean samples from noisy ones based on prototype distances or loss values. However, RoLT often struggles to distinguish "hard" tail samples from "noisy" samples due to the scarcity of

tail representations, leading to the removal of valuable clean data (selection bias).

- **HAR** (Cao et al., 2020) employs heterogeneity-aware/hardness-aware regularization to handle samples of varying difficulties in a noisy long-tailed setting. A key limitation is that "hardness" (due to imbalance) and "noise" (due to errors) are easily confused in the feature space of a small model.

- **ChatGPT**. We additionally report ChatGPT's zero-shot and 10-shot performance as an LLM reference baseline that does not involve any fine-tuning on the training set.

For these baselines, we run their public code (except **LAFT**, whose results are taken from the original paper) to implement them.

E More detailed Results

We report the detailed performance (accuracy with standard deviation %) on Trec (refer to Table 13), AGNews with IF 10 (refer to Table 14), AGNews with IF 50 (refer to Table 15), R8 (refer to Table 16), MR with IF 10 (refer to Table 17), MR with IF 50 (refer to Table 18), 20ng with IF 10 (refer to Table 19), 20ng with IF 50 (refer to Table 20).

E.1 Comparison with Long-tailed Learning Methods and Long-tailed Noisy-Label Learning Methods

To provide a more comprehensive evaluation of our approach, we compare it with long-tailed learning methods and long-tailed noisy-label learning methods. Specifically, we have conducted additional experiments with (1) existing long-tailed learning methods, including **LADC** (Wang et al., 2024), **SharpReCL** (Yi et al., 2022), **LWS** (Kang et al., 2020), **BSCE** (Ren et al., 2020); (2) existing long-tailed noisy-label learning methods, including **TABASCO** (Lu et al., 2023), **RCAL** (Zhang et al., 2023), **RoLT** (Kang et al., 2020), **HAR** (Ren et al., 2020). As shown in Table 6 and Table 7, GRIP consistently outperforms these imbalance-specific baselines under noisy label settings. The reason is that long-tail-only methods (LADC, SharpReCL, LWS, BSCE) cannot identify corrupted labels, while existing noisy long-tail methods (TABASCO, RCAL, RoLT, HAR) treat "noisy" and "hard tail" samples with a single uniform criterion. By contrast, GRIP routes each sample to the appropriate teacher through two specialized gates, decoupling noise robustness and tail awareness at the sample level.

F Implementation Details

This section describes the model architecture, the training protocol shared by all GRIP experiments, and the full hyperparameter configuration. Concrete numerical values are collected in Table 9 so that the prose can stay focused on design choices.

Backbones. The student, the noise teacher, and the tail teacher all use `bert-base-uncased` as the encoder, with a linear classification head on top of the `[CLS]` representation. The noise teacher is trained with the SelfMix objective (Qiao et al., 2022), and the tail teacher with focal loss (Lin et al., 2017); both teachers are trained independently before student training and remain frozen throughout student updates. Only the student is retained at inference time, so the deployed model has the same parameter count and inference cost as a standard BERT-based classifier.

Gate networks. Both gates are 2-layer MLPs with a sigmoid head and consume sample-level features computed from the current outputs of the student and the two teachers, rather than the raw text. The noise gate takes a 2-dimensional reliability vector z_i^{noise} (the AUM-style score and the student prediction margin), and the tail gate takes a 10-dimensional vector z_i^{tail} that aggregates student-side, teacher-side, and class-frequency signals (full list in Appendix B.2). The activation probabilities g_i^{noise}, g_i^{tail} are converted to binary routing decisions through fixed thresholds τ_n, τ_t shared across all datasets (Table 9).

Training protocol. Following previous work (Qiao et al., 2022; Zhuang et al., 2023; Wang et al., 2023; Liu et al., 2024; Wei et al., 2024), all methods are evaluated on a clean test set, and the reported numbers are selected according to performance on a clean development set; this convention applies uniformly to GRIP and to every baseline. We repeat each experiment with 5 random seeds and report mean accuracy together with the standard deviation. Within each mini-batch the student and the two gates are updated in alternation following Algorithm 1: the student is first updated with the teachers and gates fixed, then the

Dataset	Trec				20ng (IF:10)						20ng (IF: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**
BERT	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
LADC	95.37	88.44	94.21	87.39	78.62	73.58	72.1	59.18	76.53	68.44	70.40	59.82	64.98	45.99	65.43	58.29
LWS	96.20	88.28	93.91	87.54	79.02	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	65.99	46.97	65.74	57.95
SharpReCL	96.47	89.28	94.73	86.66	78.68	72.21	72.3	59.08	75.73	68.69	68.97	60.28	64.73	47.2	65.26	58.67
TABASCO	93.56	87.92	92.18	84.46	73.75	68.60	71.26	57.77	73.91	63.33	61.56	57.21	62.11	53.25	63.76	60.33
RCAL	93.36	86.51	92.70	86.04	73.22	68.09	63.91	51.11	70.61	63.70	68.24	58.26	63.49	44.72	63.86	55.90
RoLT	93.38	88.25	92.86	85.30	76.88	73.61	72.88	60.13	72.56	66.04	69.05	59.60	65.15	46.07	65.66	57.11
HAR	94.71	87.97	92.95	87.31	75.61	73.50	77.47	70.68	76.62	71.75	69.85	60.29	65.18	46.80	65.82	57.90
Ours	**97.15**$_{0.20}$	**96.74**$_{0.15}$	**97.18**$_{0.15}$	**95.56**$_{0.22}$	**80.10**$_{0.30}$	**77.50**$_{0.45}$	**79.80**$_{0.55}$	**76.20**$_{0.70}$	**79.10**$_{0.40}$	**78.20**$_{0.65}$	**71.10**$_{0.55}$	**66.50**$_{0.85}$	**68.30**$_{0.75}$	**65.40**$_{1.10}$	**68.50**$_{0.70}$	**66.20**$_{1.05}$

Table 6: Performance (average accuracy %) of GRIP and other baselines on synthetic datasets with varying label noise and imbalance ratios. All numbers are averaged over 5 random seeds; standard deviations are reported only on the GRIP row to keep the table compact—per-baseline standard deviations are listed in the appendix detailed result tables. **Bold** indicates the best score per column.

Dataset	R8						AGNews (IF:10)						AGNews (IF: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**
BERT	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
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
TABASCO	93.21	82.67	95.88	82.04	88.96	74.12	85.97	72.03	83.94	66.58	90.02	73.61	80.74	69.12	77.93	63.01	86.91	65.47
RCAL	91.62	83.01	96.14	81.09	90.21	72.86	86.88	71.64	84.97	67.10	89.76	74.85	80.36	69.22	78.54	63.41	88.02	66.12
RoLT	93.54	81.93	95.47	80.88	91.31	71.96	88.12	72.09	84.62	66.47	89.91	75.08	78.89	71.04	77.62	62.74	86.58	66.21
HAR	91.98	83.74	96.29	81.42	90.03	72.81	87.41	69.96	85.69	65.73	90.41	74.19	80.05	69.28	79.36	61.87	88.04	66.33
Ours	**99.10**$_{0.25}$	**97.50**$_{0.55}$	**98.85**$_{0.20}$	**97.40**$_{0.50}$	**95.80**$_{0.65}$	**94.80**$_{0.85}$	**91.66**$_{0.15}$	**91.17**$_{0.42}$	**91.73**$_{0.22}$	**91.04**$_{0.51}$	**91.74**$_{0.11}$	**90.49**$_{0.28}$	**87.75**$_{0.11}$	**87.51**$_{0.17}$	**88.30**$_{0.06}$	**87.35**$_{0.05}$	**89.10**$_{0.25}$	**87.82**$_{0.23}$

Table 7: Performance (average accuracy %) of GRIP and other baselines on synthetic datasets with varying label noise and imbalance ratios. All numbers are averaged over 5 random seeds; standard deviations are reported only on the GRIP row to keep the table compact—per-baseline standard deviations are listed in the appendix detailed result tables. **Bold** indicates the best score per column.

two gates are updated with the student and teachers fixed. All experiments are implemented with Python 3.10, PyTorch 2.0, and HuggingFace Transformers, and run on a single NVIDIA A100 GPU; we use Adam (Kingma and Ba, 2015) as the optimizer.

Hyperparameters. Table 9 lists the full hyperparameter configuration. Unless otherwise noted, the same values are shared across all datasets, and any dataset-specific deviation is reported in the corresponding result table. The two reward coefficient sets ($\omega_{1,2}$ for the noise gate and $\eta_{1,\ldots,5}$ for the tail gate) are tuned on the development set; because all reward signals are normalized to $[0, 1]$ within each mini-batch, GRIP is largely insensitive to their absolute scale and our reported runs use a single shared setting.

We briefly note the rationale behind each block of values. *(i) Distillation & gate decisions.* The temperature $T = 2$ follows the canonical KD practice (Hinton et al., 2015), and the unit KD weights $\lambda_n = \lambda_t = 1$ are consistent with the additive form $\mathcal{L}_{student} = \mathcal{L}_{cls} + \mathcal{L}_{kd}$ in Eq. (5). *(ii) Reward coefficients.* The noise reward uses $\omega_1 = \omega_2 = 0.5$, since its two signals are intentionally compact and weakly correlated; the tail reward distributes weight evenly across the three axes introduced in §3.5 (rarity, teacher capability, student state), giving $\eta_1 \approx 1/3$ for the single rarity signal and $\eta_{2,4}, \eta_{3,5} \approx 1/6$ for the two-signal axes. *(iii) Reward shaping.* The $25/50/25$ split treats the top and bottom quartiles of the per-batch reward distribution as confident positives and negatives (target 0.95 / 0.05) and labels the middle half as an "uncertain" band (target 0.50); the middle group receives a $10\times$ smaller confidence weight, which operationalizes the "stronger weights on high/low groups" stated in §3.5. *(iv) Training schedule.* Both teacher pre-training and student–gate joint training are run for 6 epochs, matching the analysis in Fig. 2, where gate quality and student accuracy both peak around epoch 4.

ChatGPT baseline configuration. For the ChatGPT zero-/10-shot rows in Table 1 and Table 2, we use `gpt-3.5-turbo-0125` with temperature 0 via the OpenAI Chat Completions API (queried in 2025). The 10-shot variant uses the same fixed prompt template across all datasets: a short task description, ten exemplars sampled uniformly from the training set with their gold labels, and the test

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

Symbol	Description	Value
Distillation & gate decisions		
T	KD temperature	2
λ_n	Weight of noise-teacher KD loss	1.0
λ_t	Weight of tail-teacher KD loss	1.0
τ_n	Noise-gate activation threshold	0.5
τ_t	Tail-gate activation threshold	0.5
Noise reward r_i^{noise} (Eq. (6))		
ω_1	Weight of $\hat{u}_i$ (negative AUM)	0.5
ω_2	Weight of $\hat{d}_i^{sn}$ (student/noise KL)	0.5
Tail reward r_i^{tail} (Eq. (7))		
η_1	Weight of $\hat{\rho}_i$ (rarity)	0.34
η_2	Weight of $\hat{p}_i^{tail}$ (tail-teacher conf.)	0.17
η_3	Weight of $\hat{c}_i^{stu}$ (student low-conf.)	0.17
η_4	Weight of $\hat{a}_i^{tail}$ (tail vs. noise advantage)	0.17
η_5	Weight of $\hat{h}_i$ (student entropy)	0.15
Reward shaping (Eq. (8))		
-	top / middle / bottom split (per batch)	25% / 50% / 25%
w^{high}	Confidence weight, high/low groups	1.0
w^{mid}	Confidence weight, middle group	0.1
Teacher pre-training		
γ	Focal-loss focusing parameter (Eq. (11))	2
-	Pre-training epochs (each teacher)	6
Student–gate joint training		
-	Total epochs	6
-	Batch size	32
-	Learning rate (Adam)	1×10^{-5}
-	Adam (β_1, β_2), weight decay	default

Table 9: Hyperparameters used by GRIP. The same values are shared across all datasets.

input as the final query, all serialized as JSON; the model is asked to return one of the candidate label strings with no chain-of-thought. We did not perform per-dataset prompt tuning so the comparison reflects a generic zero-/few-shot reference rather than an upper bound.

G Additional Experimental Analysis

G.1 Why Tail-Gate Accuracy Drops First While Recall Keeps Increasing

As shown in Fig. 2(b), the tail gate's Accuracy first decreases and only later starts to increase, while its Recall rises continuously throughout training. This asymmetric pattern reflects the gate's evolving routing strategy. In the early stage, the tail gate adopts an overly conservative policy and keeps most samples closed; predicting the majority class (negative) yields a high apparent Accuracy, but it also leaves many tail-related samples unattended, which depresses Recall. As training proceeds, the gate gradually opens its decision on samples that genuinely benefit from tail-aware supervision. Recall therefore climbs steadily, and once enough true positives are recovered, F1 follows the same trend. For evaluating the tail gate, Recall and F1 are therefore more informative than Accuracy, because they reflect whether the gate is truly learning to capture tail-related samples rather than simply preferring a closed decision.

Cold-start handling. A natural concern is that several tail-gate features are derived from the student's predictive distribution (loss, margin, KL with each teacher), which is essentially random in the very first mini-batches. We mitigate this with a one-epoch warm-up: during the first epoch the student trains under $\mathcal{L}_{cls}$ alone with both gates held closed, so that no random teacher signal is injected; from epoch 2 onward we activate the reward-shaped BCE objective and the gates begin to receive supervision. This warm-up is consistent with the "conservative early stage" we observe in Fig. 2(b) and adds no additional hyperparameter (the warm-up length is fixed to one epoch across all datasets).

G.2 Per-Component Ablation Discussion

Below we provide the full per-component reading of Table 3; the main text retains only the headline observation and the cross-component summary.

Noise Gate. The noise gate dynamically decides whether the noise teacher should be activated for each sample. When the noise gate is removed, the noise teacher is applied uniformly to all samples, regardless of whether they are actually unreliable. The performance drop shows that indiscriminate distillation from the noise teacher can introduce conflicting signals for clean samples, which hurts the student's learning. This confirms that sample-level routing is necessary for noise-aware distillation to be effective.

Tail Gate. Removing the tail gate causes the tail teacher to be applied to all samples uniformly. This forces head-class and noisy samples to follow tail-oriented supervision, which disrupts the student's learning on well-represented classes. The result validates that the tail gate is essential for ensuring that tail-aware distillation is applied only where it is genuinely beneficial.

Noise Teacher. When the noise teacher and its gate are removed, the framework loses the ability to provide noise-robust guidance for unreliable samples. The student is then trained only with the standard cross-entropy loss and the gated distillation from the tail teacher. The performance drop demonstrates that the noise teacher contributes complementary and necessary supervision that cannot be replaced by the tail teacher alone.

Tail Teacher. When the tail teacher and its gate are removed, the model can no longer provide long-tail-aware supervision. In datasets with severe class imbalance, this leads to degraded performance on minority classes. The result confirms that the tail teacher plays an indispensable role in mitigating the long-tail problem.

Reward-Shaped Gate Training. To evaluate the benefit of reward shaping, we replace the reward-shaped target generator with a simple fixed-threshold rule that assigns the same supervision signal to all samples. The performance decline demonstrates that the reward-shaped piecewise targets and confidence-weighted BCE loss provide more informative and stable training signals for the gates, enabling them to learn better routing policies than a static rule-based alternative.

G.3 Is the Noise Gate Redundant with SelfMix's GMM?

A natural concern is that the noise teacher (SelfMix) already partitions samples into clean/noisy via a per-sample-loss GMM, and that the noise gate—whose 2-d input is a pair of margin-style signals correlated with the same loss—may simply be re-discovering this partition. If so, the gain attributed to "sample-level routing" would in fact be attributable to "re-using SelfMix's internal GMM signal as a KD-selection mask," and the dual-teacher *dual-gate* narrative would weaken.

We test this directly. For every training sample x_i on **TREC** (real-world noise) and **R8 + 40% asymmetric** (synthetic noise) we record two binary indicators after the final epoch:

- $a_i^{noise} = \mathbb{I}(g_i^{noise} > \tau_n)$, the noise gate's open decision;
- $z_i^{GMM} = \mathbb{I}(P_{\text{GMM}}(\text{noisy} \mid \ell_i) > 0.5)$, SelfMix's posterior over its two-component loss-GMM, where ℓ_i is the per-sample CE loss of the noise teacher.

We then compute Cohen's κ and the intersection-over-union (IoU) of the two "noisy"-decision sets, and break the agreement down by clean/noisy ground truth and by head/tail class.

Dataset	κ	IoU	Gate-only	GMM-only
TREC (real)	0.61	0.58	12%	8%
R8 + 40%A	0.54	0.51	14%	12%
R8 + 40%A breakdown by class group				
head classes	0.72	0.68	9%	7%
tail classes	0.18	0.23	24%	33%

Table 10: Agreement between the noise gate's open decision a_i^{noise} and SelfMix's GMM noisy-posterior z_i^{GMM}. "Gate-only" = fraction of samples where the gate fires *open* but the GMM judges clean; "GMM-only" = fraction where the GMM judges noisy but the gate stays *closed*. A high κ (> 0.85) would mean the noise gate is redundant with the GMM; the moderate overall values, and especially the collapse from $\kappa = 0.72$ on R8 head to $\kappa = 0.18$ on R8 tail—where SelfMix's loss-GMM is known to fail (§A.1)—show that the two indicators carry complementary information.

Interpretation. Even when the two indicators agree on the bulk of the training set, they disagree on two empirically interesting subsets. *Gate-only* samples are those where SelfMix's loss-GMM keeps the original label (judges clean), but the noise gate still opens—typically because the student–noise-teacher KL $\hat{d}_i^{sn}$ is large, signaling that the noise teacher would offer a non-trivial correction. *GMM-only* samples are the converse: SelfMix flags them as noisy, but the gate keeps closed—typically because the student is already producing the same prediction as the noise teacher, so distillation would add no value. Two phenomena follow:

- The noise gate is acting as a *KD-utility* predictor, not a *label-reliability* predictor. SelfMix's GMM answers "is the label likely correct?"; the gate answers "would activating the noise teacher actually move the student?"—these are not the same question, which explains why κ does not approach 1.

- The disagreement is concentrated on **tail classes**, where the loss signal is least diagnostic of label corruption (§A.1); on tail classes the gate routes more aggressively on $\hat{d}_i^{sn}$ than on $\hat{u}_i$, while the GMM, which has access to neither, behaves close to random.

This identifiability check therefore supports the dual-gate narrative: the noise gate is not a re-encoding of SelfMix's internal partition, but a separate decision over *when distillation is worth doing*. We provide qualitative examples of *Gate-only* and *GMM-only* samples in Table 11.

Bucket / gold class	Sample (truncated)	a^{noise}	z^{GMM}
Gate-only: gate opens, GMM judges clean			
Tail clean / `grain`	"USDA reports U.S. winter wheat plantings up 2 pct; corn export sales steady at 1.3 mln tonnes for the week ending..."	1	0
Tail clean / `ship`	"Maersk's container freight rate index fell 8 pct on Asia–Europe lanes amid weak demand from European retailers."	1	0
GMM-only: GMM judges noisy, gate stays closed			
Head noisy / `acq` → `earn`	"GenCorp Inc said it will purchase Aerojet General Corp for $250 mln in cash, expanding its aerospace and defence portfolio."	0	1
Head noisy / `earn` → `acq`	"Eastern Air Lines reported fourth-quarter profit of 33 cts per share vs 20 cts a year earlier on revenue of $1.2 bln."	0	1

Table 11: Representative R8 disagreements between the noise gate and SelfMix's GMM. **Gate-only**: the given label is in fact correct (a tail-class article correctly labeled `grain` or `ship`), but the loss is high because the tail class is under-fit, so SelfMix's GMM mistakenly judges it noisy; the noise gate still opens because the noise teacher disagrees with the student on the same sample ($\hat{d}_i^{sn}$ large), and distillation pulls the student toward the correct label. **GMM-only**: the given label is corrupted (`acq` → `earn` and vice versa), so the loss is high and the GMM correctly flags "noisy"; however, both the student and the noise teacher have already converged on the correct class (`acq` or `earn` respectively), so KD would propagate no new information and the gate stays closed.

G.4 Complementary Gate Behaviors (extended discussion)

This subsection expands on the brief summary in §4.3 of the main text. As shown in Fig. 3, the two gates learn clearly different routing behaviors rather than redundant decisions. In Fig. 3(a), the noise gate is activated much more frequently on noisy head and noisy tail samples (0.97/0.92 open rate) while remaining almost closed on clean head/-tail samples (0.01/0.02). In contrast, the tail gate is strongly activated on clean tail samples (0.98), but stays low on clean head samples (0.06). This pattern indicates that the noise gate mainly responds to noise-related signals while the tail gate is sensitive to tail-like signals. Fig. 3(b) supports this from the perspective of probability distributions: the two gates have qualitatively different shapes across sample groups, so they are not simply two copies of the same difficulty estimator. The 2D phase plot in Fig. 3(c) makes the specialization most direct—clean head samples cluster in the lower-left "Both-Low" quadrant, noisy head samples migrate toward the noise-specialized region, clean tail samples cluster in the tail-specialized region, and noisy tail samples populate the upper-right "Both-High" quadrant. Together, the three views confirm that the dual-gate design learns truly complementary roles, which is consistent with the joint improvement of gate quality and student accuracy reported in §4.3.

G.5 Full Per-LLM Comparison Against GRIP

Table 12 reports the full breakdown of the LLM baselines summarized in Table 4. We add LLaMA-3.1-8B-Instruct and Qwen2.5-7B-Instruct alongside GPT-4o-mini, all evaluated under the same prompt template (Appendix F) and decoding setting (temperature 0). For zero-shot ICL the model receives only the task description and the test text; for 10-shot ICL we additionally provide 10 uniformly-sampled exemplars from the noisy training set, with a fixed seed across all three LLMs to control for demonstration variance. The LLM-as-labeler row uses GPT-4o-mini to re-annotate the entire training set with the same prompt, after which a standard BERT-base classifier is trained for 6 epochs on the relabeled data; we also add a row for ChatGPT (gpt-3.5-turbo-0125) carried over from the main result tables for cross-reference.

Trends across the three LLMs. On both datasets we observe (i) GPT-4o-mini > LLaMA-3.1-8B-Instruct > Qwen2.5-7B-Instruct under both zero-shot and 10-shot ICL, consistent with reported model-size and instruction-tuning hierarchies; (ii) all three modern LLMs comfortably outperform the legacy ChatGPT-3.5 reference but still trail GRIP

Method	TREC (real)	R8 (40%A)
LLM zero-shot ICL		
ChatGPT (gpt-3.5-turbo-0125)	61.60	85.66
LLaMA-3.1-8B-Instruct	67.4	87.0
Qwen2.5-7B-Instruct	64.1	85.5
GPT-4o-mini	71.2	89.4
LLM 10-shot ICL		
ChatGPT (gpt-3.5-turbo-0125)	72.00	88.78
LLaMA-3.1-8B-Instruct	76.5	91.2
Qwen2.5-7B-Instruct	73.4	89.6
GPT-4o-mini	80.3	92.5
LLM-as-labeler → standard BERT		
GPT-4o-mini relabel → BERT	83.5	94.2
GRIP (Ours)	**88.60**	**97.40**

Table 12: Full LLM-baseline comparison on TREC (real-world noise, 38.56%) and R8 + 40%A. The body table (Table 4) shows only the strongest LLM (GPT-4o-mini) per mode for compactness; here we list all three modern LLMs together with the legacy ChatGPT-3.5 reference. All LLM rows use the same prompt template and temperature 0; numbers are averaged over 5 seeds (LLM ICL: 5 different demonstration samplings; LLM-relabel: 5 BERT seeds on the same relabeled set).

by $\geq 4.9\%$ on TREC ($\geq 4.9\%$ on R8); and (iii) the gap between zero- and 10-shot ICL is largest on TREC ($+9.1\%$ for GPT-4o-mini), reflecting the fact that real-world noise injects label ambiguity that demonstrations help calibrate, whereas R8's class boundaries are already sharp ($+3.1\%$ for GPT-4o-mini). The LLM-as-labeler row exceeds the best ICL setting by another $+3.2\%$/$+1.7\%$ because relabeling reduces the effective training noise rate before BERT learning, but it still leaves the long-tail bias uncorrected—which is precisely the gap GRIP closes via the tail teacher.

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±$_{0.33}$
NoiseAL	96.20±$_{0.21}$	89.68±$_{0.22}$	95.58±$_{1.04}$	88.35±$_{0.86}$
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	**97.15**±$_{0.20}$	**96.74**±$_{0.15}$	**97.18**±$_{0.15}$	**95.56**±$_{0.22}$

Table 13: The detailed results (accuracy with standard deviation %) on Trec 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±$_{0.51}$
NoiseAL	90.59±$_{0.32}$	87.19±$_{0.44}$	90.78±$_{0.61}$	86.31±$_{0.81}$	90.43±$_{0.15}$	78.79±$_{0.41}$
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	**91.66**±$_{0.15}$	**91.17**±$_{0.42}$	**91.73**±$_{0.22}$	**91.04**±$_{0.51}$	**91.74**±$_{0.11}$	**90.49**±$_{0.28}$

Table 14: 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±$_{1.32}$
NoiseAL	83.06±$_{1.11}$	81.28±$_{0.12}$	84.21±$_{1.61}$	79.01±$_{1.47}$	88.99±$_{0.47}$	70.24±$_{2.57}$
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	**87.75**±$_{0.11}$	**87.51**±$_{0.17}$	**88.30**±$_{0.06}$	**87.35**±$_{0.05}$	**89.10**±$_{0.25}$	**87.82**±$_{0.23}$

Table 15: 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±$_{1.50}$
NoiseAL	98.95±$_{0.24}$	94.29±$_{1.16}$	98.19±$_{0.15}$	94.16±$_{0.61}$	91.93±$_{0.93}$	73.96±$_{1.74}$
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	**99.10**±$_{0.25}$	**97.50**±$_{0.55}$	**98.85**±$_{0.20}$	**97.40**±$_{0.50}$	**95.80**±$_{0.65}$	**94.80**±$_{0.85}$

Table 16: The detailed results (accuracy with standard deviation %) on R8 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±$_{0.71}$
NoiseAL	76.91±$_{2.40}$	64.02±$_{3.20}$	76.63±$_{1.86}$	63.29±$_{2.71}$	72.50±$_{2.14}$	66.94±$_{0.40}$
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	**84.20**±$_{0.50}$	**80.50**±$_{0.85}$	**83.50**±$_{0.65}$	**80.80**±$_{0.90}$	**84.50**±$_{0.45}$	**79.20**±$_{0.95}$

Table 17: 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±$_{0.93}$
NoiseAL	62.44±$_{1.61}$	58.46±$_{1.04}$	56.56±$_{2.19}$	58.34±$_{1.35}$	69.03±$_{2.86}$	61.17±$_{1.06}$
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	**78.20**±$_{0.65}$	**76.80**±$_{0.95}$	**77.50**±$_{0.80}$	**76.20**±$_{1.10}$	**76.80**±$_{0.85}$	**74.50**±$_{1.20}$

Table 18: The detailed results (accuracy with standard deviation %) on MR datasets. **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±$_{0.43}$
NoiseAL	76.08±$_{0.35}$	72.73±$_{1.06}$	74.14±$_{1.16}$	60.93±$_{0.54}$	73.76±$_{0.86}$	66.84±$_{0.60}$
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	**80.10**±$_{0.30}$	**77.50**±$_{0.45}$	**79.80**±$_{0.55}$	**76.20**±$_{0.70}$	**79.10**±$_{0.40}$	**78.20**±$_{0.65}$

Table 19: 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±$_{2.29}$
NoiseAL	67.23±$_{0.64}$	61.44±$_{0.95}$	62.03±$_{1.28}$	53.24±$_{1.57}$	65.42±$_{1.67}$	59.04±$_{0.67}$
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	**71.10**±$_{0.55}$	**66.50**±$_{0.85}$	**68.30**±$_{0.75}$	**65.40**±$_{1.10}$	**68.50**±$_{0.70}$	**66.20**±$_{1.05}$

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