Model-Agnostic Meta Learning for Class Imbalance Adaptation

Anonymous ACL submission

Abstract

Class imbalance is a widespread challenge in NLP tasks, significantly hindering robust performance across diverse domains and applications. We introduce Hardness-Aware Meta-Resample (HAMR), a unified framework that adaptively addresses both class imbalance and data difficulty. HAMR employs bi-level optimizations to dynamically estimate instance-level weights that prioritize genuinely challenging samples and minority classes, while a neighborhood-aware resampling mechanism amplifies training focus on hard examples and their semantically similar neighbors. We validate HAMR on six imbalanced datasets covering multiple tasks and spanning biomedical, disaster response, and sentiment domains. Experimental results show that HAMR achieves substantial improvements for minority classes and consistently outperforms strong baselines. Extensive ablation studies demonstrate that our proposed modules synergistically contribute to performance gains and highlight HAMR as a flexible and generalizable approach for class imbalance adaptation.

1 Introduction

Class imbalance naturally exists in a wide range of tasks, such as text classification (Padurariu and Breaban, 2019) and named entity recognition (Nemoto et al., 2024), yet it remains a challenge for training robust models. For instance, in the sentiment analysis of product reviews, where a large volume of generic positive feedback can easily drown out the few, highly informative negative reviews that signal critical product flaws (Henning et al., 2023). This overwhelming prevalence of majority classes biases standard models, causing them to achieve high accuracy by simply predicting the dominant class while failing to learn the features of rare but often more critical minority classes. Consequently, this leads to poor generalization and a significant drop in performance on real-world data where identifying these minority classes is paramount.

To address data imbalance, previous studies predominantly pursued two complementary strategies, reweighting and sampling, to counteract natural bias toward majority classes. The first involves algorithmic modifications that reweight the loss function to penalize errors on minority classes more heavily, such as inverse frequency weighting (Wang et al., 2010), focal loss (Lin et al., 2020), and the Dice loss (Li et al., 2020). The second strategy focuses on data-level interventions that resample the training distribution, ranging from simple oversampling and sophisticated synthetic data generation (Yu et al., 2023; Bowyer et al., 2011). A growing body of work further blends these ideas by calibrating loss terms with the "effective number" of samples or by decoupling representation learning from classifier training (Cui et al., 2019; Kang et al., 2020)

However, these methods typically rely on predefined, static heuristics that treat all samples within a class homogeneously, applying fixed adjustment rates throughout the entire training process (van den Goorbergh et al., 2022). This uniform treatment overlooks a critical insight: example difficulty does not necessarily align with class membership. Not all minority class instances are inherently challenging to classify, nor are all majority class examples trivial. Consequently, static reweighting schemes may inadvertently down-weight informative majority class examples while overemphasizing easy minority instances, leading to suboptimal learning dynamics. A significant research gap thus exists for adaptive methods that can dynamically identify and prioritize genuinely difficult examples regardless of their class affiliation, adjusting their learning strategy in response to the model's evolving understanding of the data distribution (Martins et al., 2023; Jain et al., 2024).

To bridge this gap, we propose **Hardness-Aware**

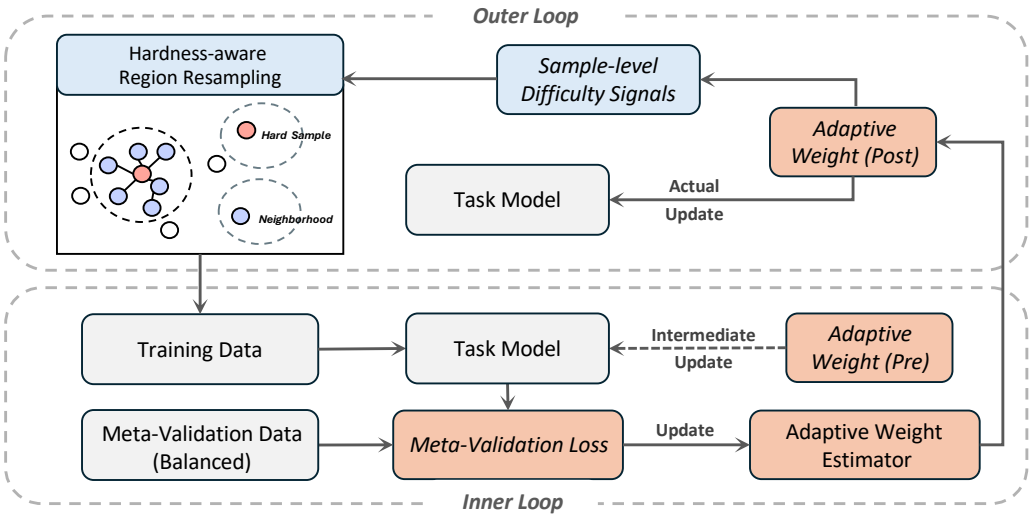

Figure 1: **Framework of Hardness-Aware Meta-Resample (HAMR).** HAMR employs a bi-level optimization: an inner loop performs an intermediate model update using pre-meta weights, and an outer loop updates the weighting network from meta-validation feedback and applies its post-meta weights for the actual model update. Embedding-based neighborhoods guide resampling toward clusters of hard examples, complementing adaptive weighting to boost generalization.

Meta-Resample (HAMR), a unified meta-learning framework. Rather than relying on static class-level heuristics, HAMR dynamically prioritizes training entries and adaptively guides the model learning focus that are most critical for generalization. Specifically, this process integrates *Adaptive Weight Estimation* to adjust the importance of individual training instances and *Hardness-Aware Region Resampling* to reshape the effective training distribution. Our contributions are summarized as follows: 1) We present Hardness-Aware Meta-Resample (HAMR) that jointly addresses class imbalance and instance-level difficulty and dynamically guides learning by identifying and prioritizing samples that are most critical for generalization across both majority and minority classes. 2) We introduce a bi-level meta-optimization that adaptively estimates instance importance based on their performance contribution, which moves beyond frequency-based or fixed reweighting schemes and allows for evolving data difficulty and model learning dynamics. 3) We propose a hardness-aware region resampling paradigm that restructures effective training distribution at semantic neighborhoods level rather than isolated instances, improving robustness under severe long-tailed distributions.

2 Method

Existing approaches including loss reweighting (Li et al., 2020) or sampling (Moreo et al., 2016) commonly focus on minority classes or weigh hard entries higher in a static view or fixed rate. But emerging yet unresolved questions are: *will minority classes be always prioritized during the learning phase, and if we can dynamically tune our training priorities on data imbalance patterns, will this be better?* To guide learning objectives and dynamically adjust training focuses, we adopt meta learning framework (Ren et al., 2018) and propose our imbalance learning approach in Figure 1, **Hardness-Aware Meta-Resample (HAMR)**. HAMR decouples *what* the model should focus on from *how* it should learn from two modules: *Adaptive Weight Estimation* dynamically guides instance-level learning signals by data difficulty across both majority and minority classes; while *Hardness-Aware Region Resampling* reshapes the training sample distribution toward challenging semantic regions. Algorithm 1 jointly optimize adaptive weighting and region resampling towards imbalance generalization in inner and outer learning loops.

2.1 Adaptive Weight Estimation

Static or heuristic sample weights are hardly fit evolving needs of training, in contrast, our module guides what the model should focus dynamically by measuring sample difficulty and learning effectiveness via the weight networks. To quantify the difficulty and effectiveness, we employ

task-specific measures. For token-level tasks (e.g., NER), we use the sentence-level maximum token loss as $\tilde{\ell}_i = \max_t \ell_{i,t}$, where $\ell_{i,t}$ is the token-level loss in sequence i. For sequential labeling or classification tasks (e.g., reasoning or inference), we deploy the per-example cross-entropy loss. Such settings can capture worst-case difficulty while remaining precise for NLP tasks. Since raw loss values can vary across batches, we apply batch-wise z-score normalization as $\hat{\ell}_i = \frac{\tilde{\ell}_i - \mu}{\sigma}$ to create a consistent scale, where μ and σ are the mini-batch mean and standard deviation. These normalized scores are processed by a lightweight weight network f_θ that maps difficulty scores to importance weights w_i. For numerical stability, the weights are renormalized to a mean of 1 per mini-batch and clipped within a fixed range.

$$\phi' = \phi - \alpha \nabla_\phi \sum_{i \in \mathcal{B}} w_i^{\text{pre}} \tilde{\ell}_i \tag{1}$$

$$\theta \leftarrow \theta - \beta \nabla_\theta \mathcal{L}_{\text{meta}}(\mathcal{D}_{\text{meta}}; \phi') \tag{2}$$

We learn the weighting network θ under bi-level meta-learning that guides the task model weight input samples. In the inner loop, we first compute per-example *pre-meta* weights $\{w_i^{\text{pre}}\}_{i \in \mathcal{B}}$ with the current weight network and perform a intermediate gradient update on the task model f_ϕ, yielding updated parameters ϕ' in Eq. (1). In the outer loop, we evaluate the temporarily updated model $f_{\phi'}$ on the balanced meta-validation set $\mathcal{D}_{\text{meta}}$, constructed by taking the full validation split and per-class sampling additional examples from the training split until each class reaches the median class count of the validation split. We then update the weight network parameters θ by minimizing the meta objective $\mathcal{L}_{\text{meta}}$ in Eq. (2). Here, $\mathcal{L}_{\text{meta}}$ denotes the average cross-entropy loss of $f_{\phi'}$ on the balanced meta-validation set $\mathcal{D}_{\text{meta}}$. After updating θ in Eq. (2), we recompute *post-meta* weights $\{w_i^{\text{post}}\}_{i \in \mathcal{B}}$ for the same mini-batch using the updated weight network ($w_i^{\text{post}} = f_\theta\left(\hat{\ell}_i\right)$), and use them for the actual update of the task model (Eq. (3)):

$$\phi \leftarrow \phi - \alpha \nabla_\phi \sum_{i \in \mathcal{B}} w_i^{\text{post}} \tilde{\ell}_i \tag{3}$$

2.2 Hardness-Aware Region Resampling

Although adaptive weighting addresses how we learn from samples, it does not control which training sample task model can see. Existing imbalance approaches (Li et al., 2020; Wu et al., 2023; Hu et al., 2025) assume that data with infrequent labels should be valued more throughout the model learning steps, while ignoring the set of hard data entries can shift. To address this, HAMR introduces a sampling strategy that dynamically adjusts what difficulty training samples task models should see by leveraging both individual hardness and semantic neighborhood patterns.

Our resampling strategy actively explores neighborhood regions of data space, not just isolated hard examples. We employ a neighborhood boost mechanism updated periodically (e.g., at the end of each epoch). It first identifies hard samples by selecting those with the highest hardness scores h_i (e.g., the top 20%). Then, using K-Nearest Neighbors (KNN) (Zhang, 2016) on precomputed embeddings, we build k nearest neighbors for each hard sample. For efficiency, we construct a FAISS index (Douze et al., 2025) for similarity search acceleration. The neighborhood boost score b_i for sample i is calculated as its total hit count, i.e., the number of times it appeared in the neighborhoods of hard samples. This score is normalized by its maximum value to lie in $[0, 1]$. This mechanism diffuses hardness from individual hard examples to their semantically similar neighbors.

$$p_i \propto (h_i + \varepsilon)^\tau \cdot (1 + \lambda\, b_i) \tag{4}$$

The final sampling probability p_i integrates each sample's individual hardness h_i with its neighborhood boost b_i, applying temperature smoothing ($\tau < 1$) to encourage balanced exploration as shown in Eq. (4), where λ controls the neighborhood boost strength and ε ensures numerical stability. To estimate individual hardness h_i, we first use the updated weighting network from §2.1 to initialize a per-sample weight w_i per training sample. The global hardness score h_i is updated over time using an exponential moving average (EMA) (Morales-Brotons et al., 2024) of w_i^{post}. EMA smooths noisy parameter updates and acts as a regularization term to reduce noisy effects and improve optimization robustness. Samples with larger h_i are treated as harder and are assigned higher sampling probabilities when forming training mini-batches.

$$h_i \leftarrow \gamma \cdot h_i + (1 - \gamma) \cdot w_i^{\text{post}} \tag{5}$$

2.3 Unified Training Procedure

The adaptive weighting and region resampling mechanisms integrate seamlessly into unified train-

Algorithm 1 HAMR unified training procedure

Input: dataset $\mathcal{D}$, meta-set $\mathcal{D}_{\text{meta}}$, model f_ϕ, weight network f_θ

Parameters: learning rates α, β, EMA factor γ, KNN refresh interval F

for each epoch **do**
  if epoch mod $F = 0$ **then**
    Update neighbourhood boosts b via KNN on hard samples
  end if
  Compute sampling probabilities p and sample mini-batch $\mathcal{B}$
  Compute per-example losses $\tilde{\boldsymbol{\ell}}$ on $\mathcal{B}$; $\mathbf{w}^{\text{pre}} \leftarrow f_\theta(\hat{\boldsymbol{\ell}})$
  Inner step: $\phi' \leftarrow \phi - \alpha \nabla_\phi \langle \mathbf{w}^{\text{pre}}, \tilde{\boldsymbol{\ell}} \rangle$
  Meta step: $\theta \leftarrow \theta - \beta \nabla_\theta \mathcal{L}_{\text{meta}}(\mathcal{D}_{\text{meta}}; \phi')$; $\mathbf{w}^{\text{post}} \leftarrow f_\theta(\hat{\boldsymbol{\ell}})$
  Outer step: $\phi \leftarrow \phi - \alpha \nabla_\phi \langle \mathbf{w}^{\text{post}}, \tilde{\boldsymbol{\ell}} \rangle$
  Update $\mathbf{h} \leftarrow \gamma \mathbf{h} + (1 - \gamma) \cdot \mathbf{w}^{\text{post}}$
end for

ing procedure in Algorithm 1. At each epoch, the framework refreshes neighborhood boosts using KNN, samples data batches according to learned hardness-aware probabilities, and computes adaptive weights reflecting sample hardness. The core bi-level optimization consists of an inner loop (intermediate model update using pre-meta weights), a meta step (updating the weighting network based on meta-validation feedback), and an outer loop (actual model update using post-meta weights produced by the updated weighting network).

3 Experiment

We conduct experiments comparing HAMR with several state-of-the-art baselines on standard NLP benchmarks from multiple domains with varying degrees of class imbalance and compare in Table 2. Our experiments aim to examine 1) if our approach outperforms baselines on majority and minority classes, 2) if our approach has consistent performance across varying datasets and imbalance settings, and 3) if individual modules of our approach contribute to overall improvements. In this section, we detail the datasets, baselines, experimental settings, ablation studies, and evaluation metrics.

3.1 Data

We evaluate our approach on six publicly available and imbalanced corpora across two tasks: named-entity recognition (NER) and text classification (CLS). For NER, BioNLP (Collier and Kim, 2004) contains PubMed abstracts annotated with five biomedical entities (e.g., protein and DNA); TweetNER (Ushio et al., 2022) comprises English tweets labeled with seven entity types including location and product; and MIT-Restaurant (Liu et al., 2013) includes online reviews with eight restaurant-related entities such (e.g., price and amenity). For CLS, Hurricane-Irma17 and Cyclone-Idai19 (Alam et al., 2021) classify disaster-response tweets into 9 and 10 categories respectively, both showing substantial class imbalance, and SST-5 (Socher et al., 2013) categorizes movie reviews into five ratings. We quantify class imbalance by the ratio $\text{IR} = |\text{maj}|/|\text{min}|$, ranging from 2.1 (SST-5) to 98.4 (Cyclone-Idai19). Table 1 summarizes data statistics with imbalance details in Appendix A.

Name	Size	#Classes	Task	Domain	Splits			IR
					Train	Valid	Test	
BioNLP	22,402	5	NER	Biomedical	16,619	1,927	3,856	33.1
TweetNER	5,768	7	NER	Social	4,616	576	576	3.7
MIT-Rest.	9,181	8	NER	Review	6,900	760	1,521	5.1
Irma17	9,399	9	CLS	Crisis	6,579	958	1,862	18.7
Idai19	3,933	10	CLS	Crisis	2,753	401	779	98.4
SST-5	11,855	5	CLS	Review	8,544	1,101	2,210	2.1

Table 1: Data statistics. IR (Imbalance Ratio) is the ratio of largest class size to smallest class size.

3.2 Baselines

To validate our approach, we compare HAMR with six state-of-the-art imbalance-learning baselines, each tackling class imbalance through a different mechanism.

Dice Loss (Li et al., 2020) and *Focal Loss* (Lin et al., 2020) reformulate the training objective by optimizing the Sørensen-Dice coefficient with overlap-based normalization or introducing a modulating factor $(1 - p_t)^\gamma$ to dynamically down-weight well-classified instances, respectively. *Inverse Class Frequency (ICF)* (He and Garcia, 2009), *Effective Number (EN) Weights* (Cui et al., 2019), and *Gradient-based Clustering (GBC)* (Wu et al., 2023) follow the re-weighting direction. ICF scales the cross-entropy loss of each class by the inverse of its empirical frequency to amplify minority class contributions. EN reduces returns of additional samples through class-balanced weights $w_y = \frac{1-\beta}{1-\beta^{n_y}}$ where n_y is the class frequency. GBC is a close meta-learning method that performs weighted k-means clustering in gradient space to supervise adaptive loss re-weighting. Our approach differs from GBC; instead of using meta re-weighting alone, we further introduce region-

Method	NER						CLS					
	BioNLP		MIT-Restaurant		TweetNER		Hurricane-Irma17		Cyclone-Idai19		SST-5	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
Dice	70.6	74.9	80.4	81.3	59.0	62.6	70.0	73.6	58.6	77.2	56.0	56.9
Focal	69.8	74.1	79.1	80.1	58.0	61.7	70.7	73.1	63.5	81.1	55.9	56.2
ICF	68.0	70.6	79.5	80.1	57.2	61.2	72.7	74.4	63.8	80.4	56.3	56.6
EN	69.2	74.0	78.5	79.1	53.9	56.9	70.0	73.3	61.5	78.3	55.9	56.3
GBC	68.4	72.4	78.7	79.3	57.0	60.2	68.8	71.9	60.0	79.9	52.9	53.9
LNR	70.1	74.6	79.4	80.6	59.0	62.7	71.0	73.4	63.4	78.8	55.7	56.4
HAMR (Ours)	**72.7**	**75.4**	**81.1**	**82.1**	**60.2**	**63.7**	**73.4**	**75.2**	**65.7**	**81.4**	**57.0**	**57.7**

Table 2: Overall performance (%) by macro- and micro-F1 scores across 6 datasets. Numbers are averaged over three runs with different random seeds. Best scores are in **bold**.

aware resampling to prioritize hard regions during training. Label-Noise Rebalancing (LNR) (Hu et al., 2025) is a close study in data augmentation by deliberately flipping a subset of majority-class labels into minority classes without discarding data or synthesizing samples. We provide more detailed baseline configurations in Appendix B.

3.3 Experimental Settings

To ensure consistency with the baselines, we use DeBERTa-v3-base (He et al., 2021) as the default task model while examine decoder-only model options such as Qwen3 (Yang et al., 2025) (Sec. 4.4). We report Micro-/Macro-F1 under the standard protocols; for NER, scores are computed at the span level under the BIO scheme excluding the O tag. All reported results are averaged over three independent runs with different random seeds, and the corresponding standard deviations are reported in Appendix D. Additional implementation details, hyperparameter sensitivity analyses, and computational overhead are provided in Appendix C, Appendix E, and Appendix F, respectively.

4 Results

This section discusses overall results and ablation studies. We conduct three ablation studies to answer the following questions: 1) does each module of our HAMR approach contribute effectively to better performance? 2) has our approach successfully encountered imbalance effects on minority classes? 3) will our approach be generalizable on other base models, such as ModernBERT (Warner et al., 2024) and Llama 3 (Grattafiori et al., 2024)?

4.1 Overall performance

Table 2 reports the performance of HAMR and six baseline methods on six benchmark datasets with varying imbalance ratios. Overall, HAMR consistently outperforms all baselines across datasets, achieving absolute gains of 0.7–7.1 percentage points in Macro-F1 and 0.3–6.8 percentage points in Micro-F1. The improvements are particularly notable on datasets with severe class imbalance, such as Cyclone-Idai19 with an imbalance ratio of 98.4. Importantly, HAMR also maintains competitive performance on less skewed datasets. For example, on TweetNER, HAMR achieves a Macro-F1 score of 60.2, compared to 59.0 for Dice and LNR, and 57.0 for the meta-learning-based GBC method. These results indicate that HAMR can automatically adapt its imbalance learning behavior across both highly imbalanced and relatively balanced settings.

HAMR also shows consistent gains over baselines that address imbalance with different mechanisms. On Hurricane-Irma17, HAMR improves over the re-weighting baseline ICF (Macro-F1 73.4 vs. 72.7; Micro-F1 75.2 vs. 74.4). On Cyclone-Idai19, HAMR surpasses the data-level rebalancing method LNR (Macro-F1 65.7 vs. 63.4; Micro-F1 81.4 vs. 78.8). On TweetNER, HAMR exceeds the meta-reweighting approach GBC (Macro-F1 60.2 vs. 57.0; Micro-F1 63.7 vs. 60.2). These comparisons suggest that combining adaptive weighting with region-aware resampling yields more consistent improvements than using a single strategy. Moreover, the consistent gains observed across both sequence-level and entity-level tasks confirm the generality of our approach. To further analyze the contribution of individual components, we next present a module-level ablation study.

4.2 Ablation Study 1: Modules

We conduct a component-level ablation study by selectively disabling key modules in HAMR: Adaptive Weight Estimation ("w/o AWE"), Hardness-Aware Region Resampling ("w/o Resampling"), the KNN-based region propagation mechanism ("w/o KNN"), and a full ablation that removes all modules ("w/o All"). Table 3 summarizes the

Method	NER						CLS					
	BioNLP		MIT-Restaurant		TweetNER		Hurricane-Irma17		Cyclone-Idai19		SST-5	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
Full	72.7	75.4	81.1	82.1	60.2	63.7	73.4	75.2	65.7	81.4	57.0	57.7
w/o AWE	71.2	74.9	80.8	81.8	57.1	60.9	71.3	73.7	64.1	80.7	56.5	57.5
w/o Resampling	72.3	75.3	80.6	81.6	59.6	63.5	71.8	74.2	61.5	79.9	56.5	57.2
w/o KNN	71.7	75.1	80.9	81.8	59.2	63.4	71.0	73.4	64.6	80.7	56.5	57.2
w/o All	71.0	74.3	79.7	80.8	57.4	60.2	70.0	72.7	61.3	79.6	55.6	56.6

Table 3: Ablation of individual modules in our HAMR. **w/o AWE** disables adaptive weight estimation, **w/o Resampling** removes hardness-aware region resampling, **w/o KNN** removes KNN-based neighborhood boost, and **w/o All** removes all modules.

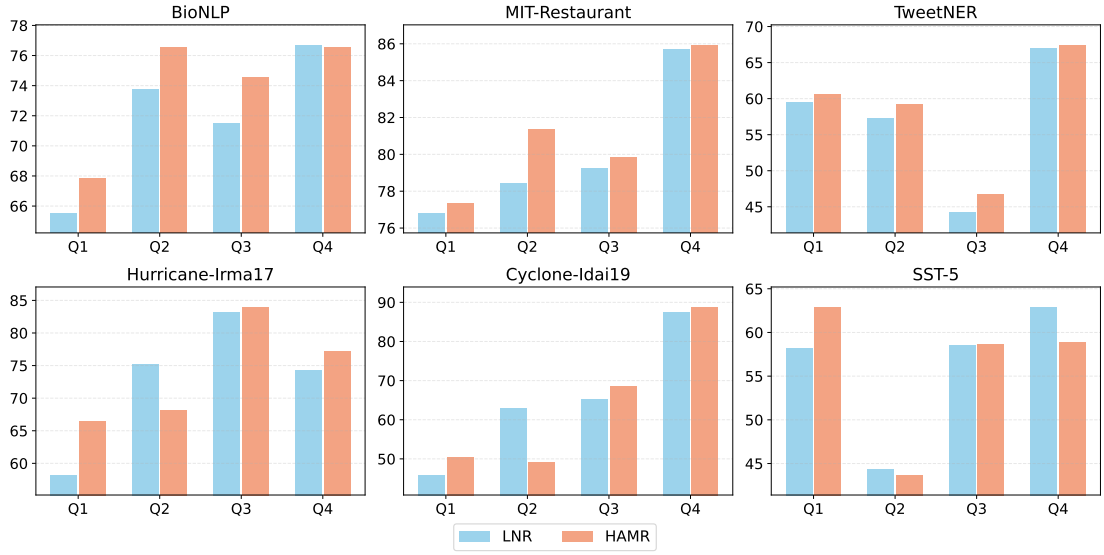

Figure 2: Model performance on majority and minority classes, grouped by quartiles, with Q1 denoting the rarest labels. Bars report the mean of per-label F1 scores within each quartile.

resulting Macro- and Micro-F1 scores across all datasets.

Across all benchmarks, the complete HAMR model consistently yields the strongest performance, indicating that the proposed components are mutually complementary. Removing all modules ("w/o All") typically yields the weakest performance, and the full HAMR model brings consistent gains over this vanilla baseline, e.g., improving Macro-F1 from 61.3 to 65.7 on Cyclone-Idai19 and from 70.0 to 73.4 on Hurricane-Irma17. Among single-module ablations, removing Adaptive Weight Estimation causes substantial degradation, particularly under severe class imbalance, with Macro-F1 decreasing from 72.7 to 71.2 on BioNLP and from 60.2 to 57.1 on TweetNER. Performance also deteriorates noticeably when Hardness-Aware Region Resampling is removed, most prominently on disaster-related datasets with extreme imbalance, such as Cyclone-Idai19 (Macro-F1 decreases from 65.7 to 61.5) and Hurricane-Irma17 (from 73.4 to 71.8). By contrast, disabling the KNN-based region propagation results in smaller yet consistent losses, for example, reductions of 0.2 and 1.0 Macro-F1 points on MIT-Restaurant and BioNLP, respectively, suggesting that this component provides complementary refinements rather than primary gains.

4.3 Ablation Study 2: Minority Class

A persistent challenge in imbalance learning is improving performance on rare classes without compromising frequent-class performance. To examine whether HAMR effectively benefits minority labels, we compare it against LNR, a representative imbalance-aware baseline. Figure 2 reports the unweighted mean of per-label F1 within label-frequency quartiles, where Q1 contains the rarest 25% of labels and Q4 the most frequent. An effective imbalance learning approach is generally expected to improve performance in Q1 and Q2 while staying competitive results in Q3 and Q4.

As shown in Figure 2, HAMR consistently yields substantial gains in the lower-frequency quar-

Method	NER						CLS					
	BioNLP		MIT-Restaurant		TweetNER		Hurricane-Irma17		Cyclone-Idai19		SST-5	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
Dice	64.2	67.1	71.5	72.0	47.5	52.1	**70.1**	73.0	42.7	52.2	49.9	54.1
Focal	62.2	66.0	71.2	72.0	44.7	49.4	68.2	72.0	65.4	80.4	52.7	55.1
ICF	66.3	68.8	74.5	74.8	47.4	51.7	**70.1**	72.5	60.7	79.0	52.9	54.2
EN	66.6	69.2	74.9	75.6	47.7	52.0	69.0	72.2	62.5	78.8	53.1	54.8
GBC	66.9	69.8	73.9	74.5	49.6	54.1	66.7	71.1	62.8	79.1	50.6	52.9
LNR	63.4	67.8	70.9	71.7	45.9	51.0	69.2	72.6	65.7	**81.0**	53.2	**55.2**
HAMR (Ours)	**68.8**	**71.3**	**76.1**	**76.8**	**52.2**	**56.9**	**70.1**	**73.1**	**66.8**	**81.0**	**54.1**	54.3

Table 4: Performance by macro- and micro-F1 (%) using ModernBERT-base on six datasets.

Model	Method	NER						CLS					
		BioNLP		MIT-Restaurant		TweetNER		Hurricane-Irma17		Cyclone-Idai19		SST-5	
		Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
Qwen3-4B	LNR	36.4	49.6	60.7	61.8	39.0	42.6	70.4	73.0	56.5	78.3	56.4	57.3
	HAMR	**40.1**	**51.4**	**64.4**	**65.5**	**40.1**	**43.5**	**71.0**	**74.1**	**58.3**	**82.0**	**56.5**	**58.2**
Qwen3-8B	LNR	37.9	50.4	62.0	63.6	39.1	43.2	69.7	72.9	52.0	78.7	56.3	57.2
	HAMR	**40.0**	**52.6**	**65.2**	**66.6**	**40.0**	**44.3**	**73.7**	**74.9**	**58.8**	**80.2**	**56.4**	**58.2**
LLaMa3-8B	LNR	38.0	50.9	63.1	64.3	43.0	47.1	69.6	72.6	58.3	79.1	56.0	57.7
	HAMR	**42.5**	**54.7**	**66.5**	**67.9**	**43.4**	**47.7**	**71.2**	**73.7**	**59.3**	**80.9**	**56.8**	**58.9**

Table 5: Performance by macro- and micro-F1 (%) using three decoder-only models on NER and CLS datasets.

tiles compared to LNR, without sacrificing performance on frequent classes. For example, on SST-5, HAMR improves Q1 mean per-label F1 by 4.7 points over LNR. Similar improvements in Q1 are observed across other datasets, including BioNLP and Cyclone-Idai19, indicating that HAMR effectively targets the most challenging rare classes. Meanwhile, performance on Q3 and Q4 remains stable or shows slight improvements, demonstrating that gains on minority classes do not come at the expense of frequent-class performance. Overall, these results confirm that HAMR achieves a favorable balance between rare-class improvement and frequent-class stability.

4.4 Ablation Study 3: Base Model

We investigate whether HAMR generalizes across different backbone architectures, including both encoder-based and decoder-only language models. To this end, we evaluate HAMR on newly released ModernBERT-base (Warner et al., 2024) model, as well as decoder-only models including Qwen3-4B, Qwen3-8B (Yang et al., 2025), and LLaMA3-8B (Grattafiori et al., 2024).

Encoder-based models. Table 4 shows that replacing DeBERTa with ModernBERT-base leads to performance reductions across all methods, indicating that absolute task performance remains sensitive to the representational capacity of the underlying encoder. Despite this shift, HAMR continues to outperform or match competing approaches across datasets, achieving the best results on 5 out of 6 benchmarks and yielding notable improvements over strong baselines such as GBC and LNR. For example, HAMR improves Macro-F1 by 2.6 points over GBC on TweetNER and by 5.4 points over LNR on BioNLP. These results suggest that the effectiveness of HAMR is robust to changes in encoder architectures.

Decoder-only models. For decoder-only architectures, we fine-tune Qwen3-4B, Qwen3-8B, and LLaMA3-8B using LoRA (Hu et al., 2021), and compare HAMR against LNR, the strongest baseline in our main experiments. As reported in Table 5, HAMR consistently outperforms LNR across all evaluated settings. The improvements are most pronounced on highly imbalanced datasets; for instance, on Cyclone-Idai19 (IR = 98.4), HAMR increases Macro-F1 by 1.8 points with Qwen3-4B and by 6.8 points with Qwen3-8B. Overall, HAMR yields larger gains on named entity recognition tasks than on text classification tasks, with median improvements of 3.2 Macro-F1 points for NER and 1.0 point for CLS, indicating that the method is particularly effective for structured prediction under severe token-level imbalance.

Encoder vs. decoder observations. We further observe a clear architectural distinction between encoder-based and decoder-only models. While decoder-only models achieve performance comparable to encoders on classification tasks, they exhibit substantially weaker results on named en-

tity recognition. For example, although LLaMA3-8B with HAMR attains competitive performance on Hurricane-Irma17, its Macro-F1 on BioNLP is lower by more than 30 points, reflecting the limitations of unidirectional context for sequence labeling. Overall, these findings confirm that HAMR generalizes well across model architectures and scales, while delivering the strongest benefits when paired with encoder-based backbones for structured prediction tasks.

5 Related Work

Meta Learning is a framework of "learning to learn" and optimizes a learner's update rule, initialization, or data-usage policy via a higher-level meta-objective, so that a few inner updates yield better generalization (Finn et al., 2017). Common approaches to generalize models are to learn how to weight or sample training data entries (Ren et al., 2018; Heck et al., 2024; Wu et al., 2023). Ren et al. adjust per-example (or per-batch) weights so that inner-loop steps improve a held-out objective; Heck et al. develops a small network or rule maps training signals (loss/margin) to weights data samples and control the weighting process by a meta-loss; and Jain et al. optimize selection of the validation set used in learned reweighting training to improve classifier generalization on images.

However, few studies leverage meta learning on the data imbalance, a critical while widely existing issue in NLP tasks (Liu et al., 2020; Henning et al., 2023). A close study developed a meta reweighting approach via gradient space clustering that constructs representative meta-sets and clusters per-sample gradients to supervise loss reweighting (Wu et al., 2023). But our approach differs from the existing work (Zhang, 2016; Ren et al., 2018; Jain et al., 2024) with a particular focus on the data imbalance (Wu et al., 2023) by unifying 'how to weight' and 'what to sample' under the meta-optimized look, which complements the prior class-level reweighting and pointwise meta-sampling.

Imbalance Learning for NLP tasks learns to handle data imbalance issues by correcting majority bias, improving minority performance, and calibrating under unequal priors (Johnson and Khoshgoftaar, 2019; Liu et al., 2020; Henning et al., 2023), which has three primary directions, loss re-engineering or class re-weighting (Li et al., 2020; Cao et al., 2019; Lin et al., 2020) and data augmentation (Edunov et al., 2018; Chen et al., 2023; Song et al., 2024; Hu et al., 2025). For example, Song et al. introduces two samplers for imbalanced labels and generates minority-augmented instances with high diversity to augment infrequent classes; Zhao et al. aligns the distributions of feature representations and label representations to alleviate the impact of the distribution gap between the training set and the test set caused by the imbalance issue; and LDAM (Cao et al., 2019) dynamically adjusts the learning process to emphasize harder classes.

Our study differs from and complements existing imbalance learning approaches (Lin et al., 2020; Cao et al., 2019; Li et al., 2020; Zhao et al., 2023; Wu et al., 2023; Hu et al., 2025) by introducing an end-to-end meta-learning framework with neighborhood-based resampling. Our meta-learning approach accumulates hardness over training and propagates it to semantic neighbors to drive resampling and sampling-based rebalancing. Recent studies show that not all minority samples are equally informative, and some majority examples remain hard to learn and underused (Cao et al., 2019; van den Goorbergh et al., 2022; Jain et al., 2024; Song et al., 2024). We meta-learn example importance to prefer "hard-but-useful" samples (even in head classes), and resample semantic neighborhoods so that training focus flows to difficult manifolds.

6 Conclusion

We presented Hardness-Aware Meta-Resample (HAMR), a framework that jointly addresses class imbalance and instance-level difficulty in NLP through adaptive sample weighting and semantic neighborhood-based resampling. HAMR leverages bi-level meta-learning to dynamically adjust sample importance and incorporates semantic neighborhood information to focus training on challenging regions of the data space. Experiments on six NER and text classification benchmarks demonstrate that HAMR consistently improves performance over strong baselines, with more pronounced gains under severe class imbalance. Quartile analysis further shows that HAMR enhances performance on infrequent classes without sacrificing performance on frequent ones. Ablation studies confirm that both adaptive weighting and neighborhood-based resampling contribute to these improvements. Overall, HAMR provides a flexible and effective approach for mitigating class imbalance across diverse tasks and model architectures.

Limitations

Despite its effectiveness, this study has several limitations that suggest directions for future research. The proposed HAMR framework involves a bi-level optimization process and periodic neighborhood updates, which introduce additional computational cost compared to conventional single-loop training. Although the overhead remains manageable in our experimental settings, scalability to very large models or corpora may require further optimization. Moreover, the neighborhood-based resampling depends on the quality of precomputed embeddings, and its performance may degrade when semantic representations are unstable or domain coverage is limited. Another limitation lies in the experimental scope. The current evaluation focuses on text classification and named entity recognition tasks, and thus the generalizability of HAMR to other NLP problems, such as relation extraction or multi-modal learning, remains to be verified. These aspects merit further investigation toward improving the efficiency, robustness, and applicability of HAMR across broader settings.

References

Firoj Alam, Umair Qazi, Muhammad Imran, and Ferda Ofli. 2021. Humaid: Human-annotated disaster incidents data from twitter with deep learning benchmarks. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 15, pages 933–942.

Naif Radi Aljohani, Ayman Fayoumi, and Saeed-Ul Hassan. 2023. A novel focal-loss and class-weight-aware convolutional neural network for the classification of in-text citations. *Journal of Information Science*, 49(1):79–92.

Kevin W. Bowyer, Nitesh V. Chawla, Lawrence O. Hall, and W. Philip Kegelmeyer. 2011. SMOTE: synthetic minority over-sampling technique. *CoRR*, abs/1106.1813.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2023. An empirical survey of data augmentation for limited data learning in nlp. *Transactions of the Association for Computational Linguistics*, 11:191–211.

Nigel Collier and Jin-Dong Kim. 2004. Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 73–78, Geneva, Switzerland. COLING.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. *Preprint*, arXiv:1901.05555.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2025. The faiss library. *IEEE Transactions on Big Data*, pages 1–17.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1126–1135. JMLR.org.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. Technical report, Meta Research. ArXiv preprint arXiv:2407.21783.

Haibo He and Edwardo A. Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Michael Heck, Christian Geishauser, Nurul Lubis, Carel van Niekerk, Shutong Feng, Hsien-Chin Lin, Benjamin Matthias Ruppik, Renato Vukovic, and Milica Gašić. 2024. Learning from noisy labels via self-taught on-the-fly meta loss rescaling. *Preprint*, arXiv:2412.12955.

Sophie Henning, William Beluch, Alexander Fraser, and Annemarie Friedrich. 2023. A survey of methods for addressing class imbalance in deep-learning based natural language processing. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 523–540, Dubrovnik, Croatia. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Guangzheng Hu, Feng Liu, Mingming Gong, Guanghui Wang, and Liuhua Peng. 2025. Learning imbalanced data with beneficial label noise. In *Forty-second International Conference on Machine Learning*.

Nishant Jain, Arun S. Suggala, and Pradeep Shenoy. 2024. Improving generalization via meta-learning on hard samples. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27590–27599.

Justin M Johnson and Taghi M Khoshgoftaar. 2019. Survey on deep learning with class imbalance. *Journal of big data*, 6(1):1–54.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. 2020. Decoupling representation and classifier for long-tailed recognition. *Preprint*, arXiv:1910.09217.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020. Dice loss for data-imbalanced NLP tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 465–476, Online. Association for Computational Linguistics.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2020. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327.

Jingjing Liu, Panupong Pasupat, Scott Cyphers, and Jim Glass. 2013. Asgard: A portable architecture for multilingual dialogue systems. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8386–8390.

Pei Liu, Xuemin Wang, Chao Xiang, and Weiye Meng. 2020. A survey of text data augmentation. In *2020 International Conference on Computer Communication and Network Security (CCNS)*, pages 191–195. IEEE.

Vinicius Eiji Martins, Alberto Cano, and Sylvio Barbon Junior. 2023. Meta-learning for dynamic tuning of active learning on stream classification. *Pattern Recognition*, 138:109359.

Daniel Morales-Brotons, Thijs Vogels, and Hadrien Hendrikx. 2024. Exponential moving average of weights in deep learning: Dynamics and benefits. *Transactions on Machine Learning Research*.

Alejandro Moreo, Andrea Esuli, and Fabrizio Sebastiani. 2016. Distributional random oversampling for imbalanced text classification. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, page 805–808, New York, NY, USA. Association for Computing Machinery.

Sota Nemoto, Shunsuke Kitada, and Hitoshi Iyatomi. 2024. Majority or minority: Data imbalance learning method for named entity recognition. *IEEE Access*.

Cristian Padurariu and Mihaela Elena Breaban. 2019. Dealing with data imbalance in text classification. *Procedia Computer Science*, 159:736–745.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Mengye Ren, Wenyuan Zeng, Binh Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Hwanjun Song, Minseok Kim, and Jae-Gil Lee. 2024. Toward robustness in multi-label classification: a data augmentation strategy against imbalance and noise. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sébastien Ourselin, and M. Jorge Cardoso. 2017. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. *CoRR*, abs/1707.03237.

Asahi Ushio, Leonardo Neves, Vitor Silva, Francesco. Barbieri, and Jose Camacho-Collados. 2022. Named Entity Recognition in Twitter: A Dataset and Analysis on Short-Term Temporal Shifts. In *The 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, Online. Association for Computational Linguistics.

Ruben van den Goorbergh, Maarten van Smeden, Dirk Timmerman, and Ben Van Calster. 2022. The harm of class imbalance corrections for risk prediction models: illustration and simulation using logistic regression. *Journal of the American Medical Informatics Association*, 29(9):1525–1534.

Deqing Wang, Hui Zhang, Wenjun Wu, and Mengxiang Lin. 2010. Inverse category frequency based supervised term weighting scheme for text categorization. *CoRR*, abs/1012.2609.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, better,

faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *Preprint*, arXiv:2412.13663.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yinjun Wu, Adam Stein, Jacob Gardner, and Mayur Naik. 2023. Learning to select pivotal samples for meta re-weighting. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. Technical report, Alibaba Cloud. ArXiv preprint arXiv:2505.09388.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large language model as attributed training data generator: A tale of diversity and bias. In *Advances in Neural Information Processing Systems*, volume 36, pages 55734–55784. Curran Associates, Inc.

Zhongheng Zhang. 2016. Introduction to machine learning: K-nearest neighbors. *Annals of Translational Medicine*, 4:218–218.

Xingyu Zhao, Yuexuan An, Ning Xu, Jing Wang, and Xin Geng. 2023. Imbalanced label distribution learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11336–11344.

A Data Imbalance Details

BioNLP contains five entity types—DNA, RNA, Protein, Cell-type and Cell-line (Table 6) from PubMed abstracts (Collier and Kim, 2004). Entities appear in only about 3% of tokens, and the class-frequency ratio between the most and least common entity types is 33.1, marking an extreme imbalance that stresses biomedical NER systems.

MIT-Restaurant has 7,136 restaurant-search queries with eight domain-specific slots (e.g., Cuisine, Price, Hours) (Table 7) (Liu et al., 2013). Despite its modest size, skew across the eight classes (IR = 5.1) and short utterance length test a model's ability to exploit limited context.

TweetNER comprises time-stamped English tweets labeled with seven entity types (Table 8) (Ushio et al., 2022). Social-media noise, rapid topic drift and an IR of 3.7 make it a realistic low-resource setting for NER in the wild.

Hurricane-Irma17 contains 9,399 annotated tweets of Hurricane Irma with nine humanitarian categories (e.g., Infrastructure and utility damage, Rescue volunteering or donation effort) (Table 9) (Alam et al., 2021). The moderate class imbalance (IR = 18.7) and social media context test a model's robustness to noisy, informal text with uneven class distributions.

Cyclone-Idai19 has 3,900 annotated tweets from Cyclone Idai (2019) with ten humanitarian categories (e.g., Rescue volunteering or donation effort, Sympathy and support) (Table 10) (Alam et al., 2021). Despite its smaller size, the severe class imbalance (IR = 98.4) tests a model's robustness to extreme distributional skew in crisis communication scenarios.

SST-5 is fine-grained Stanford Sentiment Treebank with 11,855 movie-review sentences and a five-point scale from *very negative* to *very positive* (Table 11) (Socher et al., 2013). The minority *very positive* class appears in only 2 % of sentences (IR = 2.1), making SST-5 an established but still imbalanced benchmark for sentiment analysis.

Label	Train	Val	Test	Overall
Protein	27240	3029	5067	35336
DNA	8273	1260	1056	10589
Cell_type	6090	628	1921	8639
Cell_line	3325	505	500	4330
RNA	820	131	118	1069

Table 6: BioNLP Dataset Distribution

Label	Train	Val	Test	Overall
Location	3355	462	812	4629
Cuisine	2532	307	532	3371
Amenity	2249	292	533	3074
Name	1755	146	402	2303
Dish	1353	122	288	1763
Hours	871	119	212	1202
Rating	987	83	201	1271
Price	655	75	171	901

Table 7: MIT-Restaurant Dataset Distribution

Label	Train	Val	Test	Overall
Person	4666	598	596	5860
Event	2242	256	265	2763
Product	1850	241	220	2311
Group	2242	227	311	2780
Creative_work	1661	208	179	2048
Corporation	1700	203	191	2094
Location	1259	181	165	1605

Table 8: TweetNER Dataset Distribution

Label	Train	Val	Test	Overall
Other relevant information	1651	240	467	2358
Infrastructure and utility damage	1317	192	372	1881
Rescue volunteering or donation effort	1113	162	315	1590
Injured or dead people	626	91	177	894
Displaced people and evacuations	528	77	150	755
Not humanitarian	430	63	122	615
Caution and advice	429	62	122	613
Sympathy and support	397	58	112	567
Requests or urgent needs	88	13	25	126

Table 9: Hurricane-Irma17 Dataset Distribution

Label	Train	Val	Test	Overall
Rescue volunteering or donation effort	1308	191	370	1869
Sympathy and support	338	49	95	482
Injured or dead people	303	44	86	433
Other relevant information	285	41	81	407
Infrastructure and utility damage	248	36	70	354
Requests or urgent needs	100	15	28	143
Caution and advice	62	9	18	89
Not humanitarian	56	8	16	80
Displaced people and evacuations	40	6	11	57
Missing or found people	13	2	4	19

Table 10: Cyclone-Idai19 Dataset Distribution

Label	Train	Val	Test	Overall
Negative	2218	289	633	3140
Positive	2322	279	510	3111
Neutral	1624	229	389	2242
Very positive	1288	165	399	1852
Very negative	1092	139	279	1510

Table 11: SST-5 Dataset Distribution

B Baseline Details

Dice Loss (Li et al., 2020) optimizes the Sørensen-Dice coefficient and has demonstrated effective performance on several data imbalance benchmarks (Sudre et al., 2017). Its overlap-based normalization naturally de-emphasizes the majority background, making it robust to skewed class distributions. We adopt its multi-class extension, defined as $\mathcal{L}_{\text{Dice}} = 1 - \frac{2\sum_i y_i \hat{y}_i + \varepsilon}{\sum_i y_i + \sum_i \hat{y}_i + \varepsilon}$, where $y_i \in \{0, 1\}$ is a ground-truth indicator for a given class, and $\hat{y}_i \in [0, 1]$ is predicted probability. We initialize the smoothing constant ε as 10^{-5} and optimize hyperparameters by different tasks and datasets.

Focal Loss (Lin et al., 2020) adds a modulating factor to focus on data samples with hard minority classes and down-weight well-classified (easy) instances, which formulates as $\mathcal{L}_{\text{Focal}} = -\alpha(1 - p_t)^{\gamma} \log p_t$, where p_t refers to a predicted probability for the true class. The approach has achieved leading performance in classification tasks (Aljohani et al., 2023). We follow those practice to set the focusing parameter $\gamma = 2$ and adopt a class-balanced weighting schema for α.

Inverse Class Frequency (ICF) (He and Garcia, 2009) is a classic re-weighting strategy that scales the loss of each class by the inverse of its empirical frequency in the training set. This weighting alleviates the dominance of majority classes by amplifying the contribution of minority classes during optimization. Formally, for class c with empirical frequency f_c, the weight is defined as $w_c = \frac{1/f_c}{\frac{1}{C}\sum_{j=1}^{C} 1/f_j}$, normalized to have mean 1. We apply ICF on the cross-entropy loss, i.e., $\mathcal{L}_{\text{ICF}} = -\sum i w_{y_i} \log p_{y_i}$, where y_i is the ground-truth label and p_{y_i} is the predicted probability.

Effective Number (EN) Weights (Cui et al., 2019) reweight cross-entropy loss using the effective number of samples, which models diminishing returns when more data is added to the same class. This weighting naturally increases the contribution of minority classes while suppressing the dominance of frequent ones, making it robust to long-tailed distributions. We adopt the class-balanced cross-entropy formulation, defined as $\mathcal{L}_{\text{CB-CE}} = -\sum_i w_{y_i} \log p_\theta(y_i \mid x_i)$, where $w_y = \frac{1-\beta}{1-\beta^{n_y}}$ with class frequency n_y and hyperparameter $\beta \in [0, 1)$. Following prior work, we set $\beta = 0.9999$ across all tasks.

Gradient-based Clustering (GBC) (Wu et al., 2023) is a meta-reweighting method that replaces heuristic meta-set construction with gradient-space clustering. Each sample is represented by concatenating per-sample gradients from a fixed number of randomly sampled layers, and weighted k-means is performed in this gradient space to select representative meta samples, which then supervise loss re-weighting. We set the number of clusters to $M = \lceil \rho N \rceil$ (with $\rho = 0.1$ in our experiments), extract gradient features from three layers, use $|\cos|$ distance for clustering, and choose the nearest point to each centroid as the meta set.

Label-Noise Rebalancing (LNR) (Hu et al., 2025) has achieved top performance in classification tasks and introduced asymmetric label noise as a plug-and-play data-level remedy for class imbalance. By deliberately flipping a small, carefully chosen subset of majority-class labels into minority classes, LNR shifts the decision boundary toward rare classes without discarding data or synthesizing new samples. We employ LNR source codes and adopt the best-practice settings of noise rate ($\rho = 0.15$) and temperature schedule.

C Implementation Details

To validate our approach, we implement all methods by HuggingFace Transformers (Wolf et al., 2020) and PyTorch (Paszke et al., 2019) and conduct experiments on the same data splits. All experiments were conducted on a server with an NVIDIA H100 GPU. We fine-tuned the microsoft/deberta-v3-base[1] model as the backbone for all NER and classification tasks. Our proposed HAMR framework was implemented as a custom subclass of the Hugging Face Trainer, integrating a meta-learning module (WNet) for dynamic sample reweighting and a k-NN mechanism for neighbor boosting. The k-NN component utilized pre-computed sentence embeddings generated by the bge-large-en-v1.5 model[2] and was accelerated by a faiss-gpu index[3] for efficient similarity search. We used the AdamW optimizer with FP16 mixed-precision for 8 epochs. The task model learning rate, along with the specific hyperparameters for our HAMR framework, were tuned for each dataset to achieve optimal performance.

Parameter	Search space
`hardness_alpha`	$[0, 1]$
`knn_lambda`	$[0, 1]$
`knn_ratio`	$[0, 1]$
`knn_k`	$\{5, 10, 15, 20\}$
`learning_rate`	$\{1, 2, 3, 4, 5\} \times 10^{-5}$
`wnet_lr`	$\{1, 2, 3, 4, 5\} \times 10^{-4}$
`meta_update_lr`	$\{1, 2, 3, 4, 5\} \times 10^{-4}$

Table 12: Hyperparameter search spaces.

For each dataset, we performed multiple runs within predefined hyperparameter search spaces (see Table 12) and selected the configuration that achieved the best validation score. The resulting settings used for evaluating the test split are summarized in Table 13.

[1] https://huggingface.co/microsoft/deberta-v3-base

[2] https://huggingface.co/BAAI/bge-large-en-v1.5

[3] https://faiss.ai/index.html

D Stability Analysis

To assess run-to-run stability, we repeat each experiment three times with different random seeds (42, 43, 44) and report the standard deviation ($\pm\sigma$) of Macro-F1 and Micro-F1 on all six datasets in Table 14. Overall, the observed variability is moderate and consistent with seed sensitivity commonly seen in imbalanced NLP benchmarks, with larger deviations typically appearing in Macro-F1 due to its higher reliance on minority-class performance. Across datasets, the standard deviation generally falls below 1.0 point on BioNLP, MIT-Restaurant, TweetNER, and SST-5, while the two crisis datasets (Hurricane-Irma17 and Cyclone-Idai19) exhibit higher variability (often around 1.0–1.8 points), reflecting noisier social-media text and more extreme label skew. For our method, HAMR shows comparable stability to strong baselines: its standard deviation is $0.72/0.61$ (Macro/Micro) on BioNLP, $0.54/0.48$ on MIT-Restaurant, $0.85/0.72$ on TweetNER, $1.05/0.91$ on Hurricane-Irma17, $1.42/1.02$ on Cyclone-Idai19, and $0.82/0.74$ on SST-5. These results indicate that the bi-level meta-weighting and neighbor-based resampling do not introduce excessive sensitivity to initialization and that performance trends are reproducible across seeds.

E Hyperparameter Sensitivity Analysis

We conduct a sensitivity analysis of HAMR with respect to three hyperparameters—hardness weighting (`hardness_alpha`), neighbor count (`knn_k`), and hard-neighbor ratio (`knn_ratio`) on two high-imbalance datasets: Cyclone-Idai19 (CLS; IR=98.4) and BioNLP (NER; IR=33.1). Here, `hardness_alpha` is the exponent used to power-scale the per-sample weights w in `HardnessAwareSampler`, i.e., $(w + \epsilon)^{\alpha}$. `knn_k` denotes the number of nearest neighbors (top-k) retrieved by FAISS for each hard sample during neighbor boosting. `knn_ratio` specifies the top-$N\%$ samples with the largest weights that are treated as hard samples and used to trigger boosting.

For `hardness_alpha`, Table 15 shows that BioNLP varies by at most 1.2 Macro-F1 points

Dataset	hardness_alpha	knn_lambda	knn_k	knn_ratio	learning_rate	wnet_lr	meta_update_lr
BioNLP	0.7	0.7	15	0.5	2×10^{-5}	3×10^{-4}	2×10^{-4}
MIT-Restaurant	0.7	0.1	10	0.1	2×10^{-5}	3×10^{-4}	2×10^{-4}
TweetNER	0.7	0.7	10	0.5	2×10^{-5}	3×10^{-4}	2×10^{-4}
Hurricane-Irma17	0.7	0.1	10	0.1	4×10^{-5}	1×10^{-4}	2×10^{-4}
Cyclone-Idai19	0.7	0.1	10	0.1	4×10^{-5}	1×10^{-4}	2×10^{-4}
SST-5	0.6	0.3	10	0.2	2×10^{-5}	5×10^{-4}	2×10^{-4}

Table 13: Hyperparameter settings for different datasets.

	BioNLP		MIT-Restaurant		TweetNER		Hurricane-Irma17		Cyclone-Idai19		SST-5	
Method	**Macro**	**Micro**	**Macro**	**Micro**	**Macro**	**Micro**	**Macro**	**Micro**	**Macro**	**Micro**	**Macro**	**Micro**
Dice	0.58	0.49	0.45	0.38	0.92	0.81	0.88	0.81	1.15	0.79	0.68	0.59
Focal	0.62	0.55	0.51	0.44	0.95	0.84	0.91	0.83	1.32	0.91	0.75	0.62
ICF	0.74	0.68	0.55	0.49	1.05	0.92	1.12	0.98	1.45	1.05	0.84	0.71
EN	0.82	0.75	0.68	0.58	1.18	1.05	1.25	1.10	1.68	1.18	0.92	0.82
GBC	0.95	0.88	0.74	0.65	1.35	1.18	1.38	1.22	1.85	1.34	1.05	0.91
LNR	0.68	0.58	0.52	0.46	0.98	0.82	1.02	0.92	1.28	0.85	0.78	0.68
HAMR (Ours)	0.72	0.61	0.54	0.48	0.85	0.72	1.05	0.91	1.42	1.02	0.82	0.74

Table 14: Standard deviation ($\pm\sigma$) for Macro-F1 and Micro-F1 scores over 3 runs.

and 0.8 Micro-F1 points over hardness_alpha $\in$ [0.1,0.9]. On Cyclone-Idai19, the best results occur at hardness_alpha $= 0.7$ (66.2 Macro-F1, 81.3 Micro-F1), and Micro-F1 decreases at hardness_alpha $= 0.9$ (79.3). For knn_k, Table 16 shows that BioNLP is stable across knn_k $\in \{5, 10, 15, 20\}$: Macro-F1 ranges from 71.4 to 72.8, and changes by only 0.1 for knn_k $\in [10, 20]$. Cyclone-Idai19 shows a larger Macro-F1 range (64.0–66.1); Micro-F1 peaks at knn_k $= 10$ (81.4), whereas Macro-F1 peaks at knn_k $= 20$ (66.1). For knn_ratio, Table 17 shows the same pattern: BioNLP varies within 1.1 Macro-F1 points and 0.9 Micro-F1 points across the tested ratios. Cyclone-Idai19 is sensitive to very small ratios (e.g., knn_ratio $= 0.3$ yields 62.8 Macro-F1), while moderate-to-high ratios produce similar results, with the best at knn_ratio $= 0.7$ (66.2 Macro-F1, 81.3 Micro-F1). Overall, HAMR is insensitive to these hyperparameters on BioNLP. On Cyclone-Idai19, moderate settings yield consistent performance; in these sweeps, knn_k ≈ 10, hardness_alpha ≈ 0.7, and knn_ratio ≈ 0.7 are reasonable defaults.

hardness_alpha	Cyclone-Idai19		BioNLP	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1
0.1	65.1	80.1	71.5	74.7
0.3	65.6	80.8	72.2	75.0
0.5	65.8	80.9	72.2	75.0
0.7	**66.2**	**81.3**	**72.7**	**75.4**
0.9	66.0	79.3	72.6	74.6

Table 15: Sensitivity Analysis of hardness weighting

knn_k	Cyclone-Idai19		BioNLP	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1
5	64.0	80.2	71.4	74.9
10	**65.7**	**81.4**	72.7	75.1
15	65.2	80.6	72.7	**75.4**
20	66.1	80.7	**72.8**	75.3

Table 16: Sensitivity Analysis of Neighbor Count

knn_ratio	Cyclone-Idai19		BioNLP	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1
0.1	65.4	79.7	72.7	75.3
0.3	62.8	80.0	72.3	74.9
0.5	65.8	80.4	71.6	74.5
0.7	**66.2**	**81.3**	**72.7**	**75.4**
0.9	66.0	81.1	72.2	74.8

Table 17: Sensitivity Analysis of KNN hard-neighbor ratio

F Computational Overhead

We report training-time and peak GPU-memory cost on one high-imbalance dataset per task: Cyclone-Idai19 (CLS; IR=98.4) and BioNLP (NER; IR=33.1). Table 18 compares HAMR with a meta-learning baseline (GBC) and a data-augmentation baseline (LNR). HAMR is far more efficient than GBC: on BioNLP it cuts training time by 47% (4012s vs. 7603s) and peak memory by 68% (11.3GB vs. 35.2GB). This comes from performing KNN retrieval only once per epoch instead of every step. HAMR is more expensive than LNR, but the added cost brings clear accuracy gains; on Idai19, HAMR improves Macro-F1 by +1.7 and Micro-F1 by +2.6. These results clarify

Baselines	Cyclone-Idai19		BioNLP	
	Time (s)	Memory (GB)	Time (s)	Memory (GB)
GBC	849.2	15.5	7603.4	35.2
LNR	566.0	9.0	1804.7	3.0
HAMR	686.0	10.2	4011.9	11.3

Table 18: Efficiency Comparison on Different Datasets

the cost–benefit profile.