Adaptive Fine-Tuning for Class Imbalance in NLP

Ding Zhu
The Ohio State University
zhu.3723@osu.edu

Xiukun Wei
The Ohio State University
wei.1418@osu.edu

Zhongteng Cai
The Ohio State University
cai.1125@osu.edu

Abstract

Class imbalance remains a major challenge in Natural Language Processing (NLP), often leading models to underperform on minority classes. Existing solutions typically assume shared parameters across all classes, limiting their ability to distinct class-specific characteristics. Inspired by approaches in computer vision and parameter-efficient tuning, we propose a class-aware LoRA expert routing framework that dynamically selects adaptation modules based on input class. This enables class-specific adaptation without significantly increasing computational overhead, improving performance on minority classes while maintaining efficiency.

1 Introduction

Natural Language Processing (NLP) face the challenge of class imbalance, where certain classes appear significantly less frequently than others. This imbalance leads to biased model performance, as deep learning models tend to prioritize the majority class by sacrificing the accuracy of the minority class (Henning et al., 2023), as shown in Fig.1. However, in applications such as hate speech detection (Waseem and Hovy, 2016) and rare disease classification (Mullenbach et al., 2018), minority class instances often carry greater significance, where class imbalance can severely impact the reliability and fairness of these critical tasks. The issue of bias in language models is often attributed to class imbalance in the training data.

Several approaches have been proposed to address class imbalance, including data resampling (Pouyanfar et al., 2018; Tepper et al., 2020), data augmentation (Zhang et al., 2022; Li et al., 2024b), loss function modifications (Tepper et al., 2020) and staged training strategies (Jang et al., 2021). While these approaches offer improvements, they also have limitations. Resampling can introduce distributional bias, data augmentation methods rely on additional annotations or synthetic data generation, and loss function modifications often require careful hyperparameter tuning.

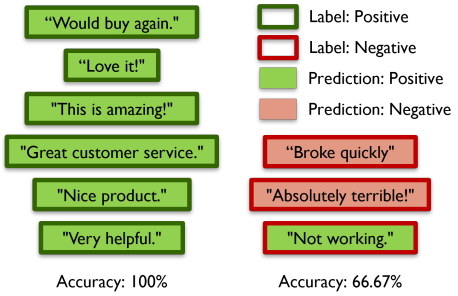


Figure 1: An example of class imbalance. Although the overall classification accuracy appears high (88.89%), the model performs poorly (66.67%) on the negative class due to insufficient training.

Most existing approaches to class imbalance assume that all classes share the same model parameters. As a result, it fails to capture the distinct data distributions and semantic features associated with minority classes. In vision domain, Kang et al. (2020) addressed the class imbalance by decoupling the representation and classification components and assigning separate classifiers to different classes, effectively mitigating the dominance of majority classes. Inspired by this insight, we argue that NLP models can also benefit from class-specific parameterization to better capture minority class characteristics.

However, allocating different parameters for each class is computationally expensive, particularly for large-scale language models. To balance flexibility and efficiency, we draw inspiration from MixLoRA (Li et al., 2024a), which introduces a parameter-efficient fine-tuning framework that dynamically routes inputs to LoRA modules based on input features. We propose a class-aware LoRA expert routing framework, where inputs are directed to class-relevant adaptation modules. This design

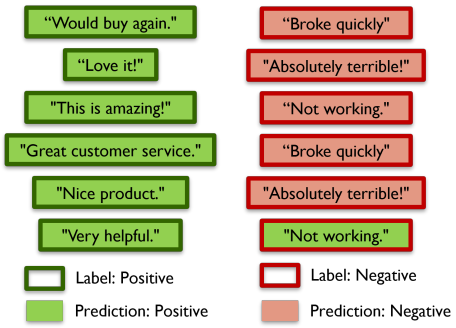


Figure 2: An example of data resampling applied to the dataset shown in Fig.1.

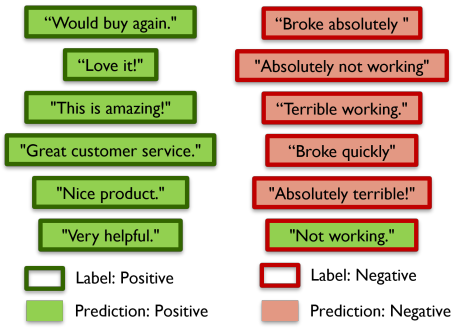


Figure 3: An example of data augmentation applied to the dataset shown in Fig.1.

enables selective parameter optimization tailored to class-specific characteristics, without replicating the full parameter set for each class. As a result, our method enhances learning on imbalanced data while maintaining a low computational overhead.

Our experiments on the imbalanced dataset created from CIFAR10 and CoNLL-2003 datasets demonstrate that class-aware LoRA improves macro and weighted F1 scores, particularly benefiting the minority classes, while maintaining computational efficiency.

2 Related Work

Class imbalance is a long-standing challenge in both computer vision and NLP. Class imbalance in NLP is especially challenging because textual data is high-dimensional and discrete (Henning et al., 2023). Moreover, rare classes often lack sufficient labeled examples (Waseem and Hovy, 2016). Several solutions have been proposed to tackle this issue.

Data resample. One of the most widely adopted techniques is data resampling. Oversampling duplicates minority class samples to balance class distribution, while undersampling reduces the number of majority class samples (Pouyanfar et al., 2018; Tepper et al., 2020), as shown in Fig.2. Pouyanfar et al. (2018) used dynamic sampling strategies for convolutional networks, showing that adaptive sample balancing can mitigate overfitting. However, such methods often introduce distributional shifts or remove potentially informative samples as minority class samples often carry unique and diverse semantic information.

Data augmentation. Data augmentation aims to increase the diversity and quantity of training data by generating additional synthetic examples from existing samples, as shown in Fig.3. Easy Data Augmentation (EDA) (Wei and Zou, 2019) apply synonym replacement, random insertion, and sentence shuffling to generate new training samples. Adam (Zhang et al., 2022) is an attentional augmentation method tailored for extreme multi-label classification, while RePrint (Li et al., 2024b) generates samples by randomized extrapolation based on principal components. Despite their effectiveness, these methods often rely on heuristics or synthetic generation, which may not fully capture the complexity of minority class semantics.

Loss function modification. Modifying the loss function helps the model focus on minority classes by assigning higher penalties to their misclassifications. For example, Tepper et al. (2020) combined synthetic data generation with a reweighted loss to better match class distributions in multi-class text classification. Focal loss (Lin et al., 2018) emphasizes hard-to-classify examples and class-balanced loss (Cui et al., 2019) adjusts weights based on the effective number of samples per class. However, these approaches often require careful hyperparameter tuning, such as choosing appropriate class weights or focusing factors.

Staged training stragies. Staged training strategies progressively adjust the model's focus from majority to minority classes. Jang et al. (2021) proposed Sequential Targeting, a continual learning framework that fine-tunes models on difficult subsets to improve performance on underrepresented classes. A common approach follows three stages: initial training on the full dataset, focused fine-tuning on the minority class, and a final refinement on the entire dataset to retain generalization. These methods can be effective but increase training complexity and may require dataset-specific schedules.

Existing methods often assume shared parame-

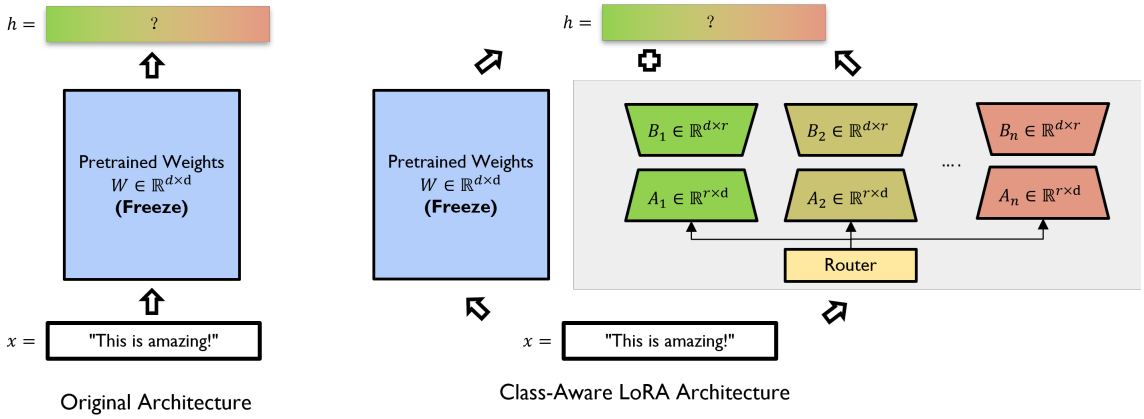


Figure 4: The architectures of the original framework (left) and the proposed class-aware LoRA framework (right). Our method dynamically routes inputs to class-specific LoRA modules by a lightweight router, allowing each class to have its own dedicated adaptation parameters.

ters across all classes, which overlooks the distinct characteristics of different classes. To better capture the unique features of minority classes, we introduce class-specific parameterization for NLP models. To avoid the overhead of maintaining separate parameters, we adopt a design inspired by MixLoRA (Li et al., 2024a) that dynamically routes inputs to a set of LoRA experts. This enables class-aware adaptation while maintaining parameter efficiency.

3 Proposed Methodology

In standard architectures, it is assumed that all inputs share a common set of parameters W_0, and the output for a given input x is computed as:

$$h = W_0 x \tag{1}$$

However, this design has limitations under class imbalance, where certain classes are significantly underrepresented. In such cases, the model tends to focus on majority classes during training, resulting in poor generalization on minority classes.

To address this issue, we propose a lightweight and extensible module that provides class-specific adaptation with low computational overhead. The architecture of the proposed framework is shown on the left side of Fig.4. Given an input x, the router outputs a selection signal, which activates the corresponding class-specific LoRA module. The selected module then calculates the input using its own set of parameters to produce the final output.

3.1 Class-Adaptive LoRA module

Our design follows the LoRA-based method (Hu et al., 2021), extended to assign different adaptation matrices to different data points based on their class.

Consider a pretrained weight matrix $W_0 \in \mathbb{R}^{d\times d}$ in a linear transformation layer and an input x_i belonging to class i. Instead of being updated directly, the pretrained weight matrix W_0 is kept frozen, and a trainable low-rank residual ΔW is added. The output is computed as:

$$\begin{aligned} h &= W_0 x_i + \Delta W_i x_i \\ &= W_0 x_i + B_i A_i x_i \end{aligned} \tag{2}$$

where $A_i \in \mathbb{R}^{r\times d}$ and $B_i \in \mathbb{R}^{d\times r}$ are low-rank trainable matrices specific to class i. The rank r is a hyperparameter that determines the size of the low-rank adaptation, offering a trade-off between fine-tuning performance and computational efficiency. Only A_i and B_i are updated during training, while the pretrained weight W_0 remains frozen.

This design has three key advantages:

- **Focus on class-specific features:** By assigning separate adaptation modules to each class, the model can learn minority class independently (Ren et al., 2020). This can reduce interference among minority classes and allow the model to better capture their unique features.

- **Parameter efficiency:** Since only low-rank matrices are trained while the backbone remains frozen, the approach is well-suited for resource-constrained environments. Compared to full fine-tuning, it updates only a small fraction of parameters.

- **Plug-and-play design:** Our approach introduces minimal modifications to the model architecture, allowing it to function as a plug-in that can be easily integrated with other methods.

3.2 Rank Router Design

In practical, class labels are not available during inference. And in some cases, class labels may even be inaccessible during training. To enable class-specific adaptation without relying on explicit labels, the model must learn to automatically select which set of parameters to apply for each input.

To achieve this, we draw inspiration from the mixture-of-experts framework (Li et al., 2024a), where a routing mechanism is used to assign inputs to one or more specialized expert sub-networks. In our case, a router is used to dynamically select the appropriate class-specific LoRA module. Specifically, We consider three increasingly flexible routing strategies:

Pseudo Labels. The simplest routing strategy uses the prediction from the origin model as the pseudo label $\hat{i}$ to select the class-specific LoRA module. Each input is routed to a single module that corresponds to the most likely class $\hat{i}$. Then the h would be:

$$h = W_0 x_{\hat{i}} + B_{\hat{i}} A_{\hat{i}} x_{\hat{i}}. \tag{3}$$

Although this routing strategy is simple and efficient, it introduces a risk that incorrect pseudo-labels driving the model away from the optimal solution, potentially leading to suboptimal performance.

Top-k selection. To improve robustness, the router can be extended to select the top-k most likely classes. Instead of selecting only the highest-scoring class, the router activates the adaptation modules corresponding to the top-k predicted classes.

The outputs of the selected modules are combined based on a diagonal mask that controls which class-specific parameters contribute to the final output. Specifically, the output is computed as:

$$\begin{aligned} h &= W_0 x_i + BMAx_i \\ M_{jk} &= \begin{cases} 1 & \text{if } j = k \text{ and } j \in \text{Top-}k(x_i) \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{4}$$

where $B \in \mathbb{R}^{d_{out} \times c}$, $A \in \mathbb{R}^{c \times d_{in}}$ and $M \in \mathbb{R}^{c \times c}$ is a diagonal binary mask matrix, where only M_{jj} is set to 1 if j is among the top-k predicted classes for input x_i. k is a controlled hyperparameter that trades off between adaptation specificity and robustness.

Compared to a single pseudo-label, the top-k selection reduces the risk of misrouting due to noisy or uncertain predictions. It also allows smoother learning in early stages of training when class decision boundaries are less stable.

Routing at the Final Layer. Prior research (Kang et al., 2020) suggests that, except in cases of extreme imbalance, the classifier is the primary underperforming component in imbalanced datasets.

Motivated by this observation, we apply class-specific LoRA modules only at the final classification layer. This avoids modifications to earlier layers and enables efficient, single-pass inference.

4 Experiments

Datasets: We have examined the performance of our algorithm on two datasets: a computer vision dataset CIFAR-10-LT, and an NLP dataset CoNLL-2003. Both datasets have imbalanced classes.

Metric: We evaluate the results using six metrics: weighted and macro averages of precision, recall, and F1-score. The weighted average computes per-class metrics weighted by the number of true instances in each class, giving more influence to frequent classes. In contrast, the macro average calculates the unweighted mean of the per-class metrics, treating all classes equally. For imbalanced datasets, the weighted average metrics are typically higher, while the macro average metrics tend to be lower due to the greater influence of minor classes with inferior model performance.

4.1 Results for CIFAR-10-LT

CIFAR-10-LT (CIFAR-10 Long Tail) dataset is an imbalanced dataset created from the original **CIFAR-10** dataset (Krizhevsky et al., 2009). CIFAR-10 is a dataset composed of 60,000 tiny images of size 32×32 with 10 non-overlappig classes, 50,000 for training and 10,000 for testing. To create an imbalanced dataset, we keep the original validation set, but resample the training set, such that the number of samples in each class decreases exponentially, and the degree of imbalance can be controlled by the imbalance ratio ρ, which defines the ratio between the size of the most frequent class and the least frequent class, i.e., $\rho = \max_i\{n_i\}/\min_j\{n_j\}$, where n_i is the

number of samples belonging to class i. In our experiment, we choose $\rho = 50$ so that class 0 would have 5000 samples for training while class 9 only has 100 samples to train.

The backbone model we use for this experiment is ResNet18 (He et al., 2015). We only add class-adaptive LoRA in the last classification head, which have been shown to be effective enough in prior works (Yang et al., 2022). The rank for each class is 1. We choose the top 2 classes as the pseudo labels according to the outputs of the pre-trained classifier.

Results are as shown in Fig 5 and Table 1. Results show that our Class-Adaptive LoRA (CALoRA) achieves higher accuracy on minor classes (e.g., Class 4-9), and better precision and F1 score, compared with the backbone ResNet18 model.

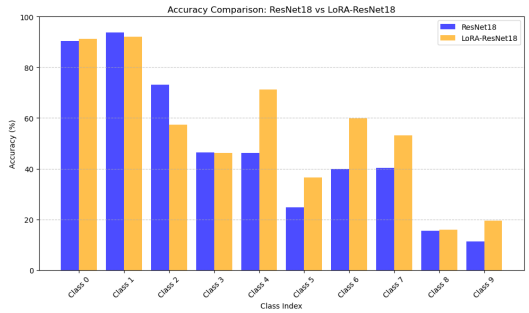


Figure 5: The accuracy of different classes in the CIFAR-10-LT dataset. Results show that our method can achieve higher accuracy on minor classes.

Metric	ResNet18	CALoRA
macro-precision	0.6310	**0.6396**
macro-recall	0.5386	0.5340
macro-F1 score	0.5129	**0.5131**
weighted-precision	0.6310	**0.6396**
weighted-recall	0.5386	0.5340
weighted-F1 score	0.5129	**0.5131**

Table 1: The macro and weighted averages of precision, recall and F1 score of ResNet18 and ResNet18 with class-adaptive LoRA (CALoRA) in CIFAR-10-LT dataset. Results show improvements on precision and F1 score.

4.2 Results for CoNLL- 2003

CoNLL-2003 is a dataset (Tjong Kim Sang and De Meulder, 2003) for language-independent named entity recognition, containing sentences collected from English and German articles. Each word in the dataset is annotated with a part-of-speech (POS) tag, a syntactic chunk tag, and a named entity tag, which can be roughly classified into four types: persons, locations, organizations, and entities that do not belong to previous three groups. These four types can be further categorized into 11 labels for our classification task including "O", "B-MISC", "I-MISC", "B-PER", etc. Since a sentence sample includes many tokens, it is hard to sample a certain number of tokens in one class as what we do in image multi-class classification. Therefore, we sample 100 sentences including both "B-MISC" and "I-MISC" tokens, and drop other sentences which includes only either of them. Then we append the rest of the training data, which includes none of the tokens above, to these 100 samples, hence creating an imbalanced dataset.

The backbone model for this task is BERT (Devlin et al., 2019) with a linear classifier head to classify each token. We only add class-adaptive LoRA in the classification head. The rank of each class is 8. We choose top 2 pseudo labels generated by the vanilla model. Results show that our CALoRA show improvements on all metrics compared with the backbone model.

Metric	BERT	CALoRA
macro-precision	0.8031	**0.8319**
macro-recall	0.8752	**0.8830**
macro-F1 score	0.8321	**0.8542**
weighted-precision	0.8207	**0.8632**
weighted-recall	0.8867	**0.8925**
weighted-F1 score	0.8462	**0.8761**

Table 2: The macro and weighted averages of precision, recall, and F1 score of BERT and BERT with class-adaptive LoRA in CoNLL 2003 dataset. Results show improvements on all metrics.

4.3 Ablation Study

We conduct ablation studies to explore the influence of model hyperparameters, including the rank chosen for each class, and the number of used pseudo labels k. We fix all other settings and vary only the targeted hyperparameters to isolate its effect. All experiments are conducted on CIFAR-10 dataset.

Rank: Firstly, we explore the influence of rank in our experiments. Higher rank means the class-specific adapter can have more extensive representation on the minority class. We show the model performance associated with different ranks in Ta-

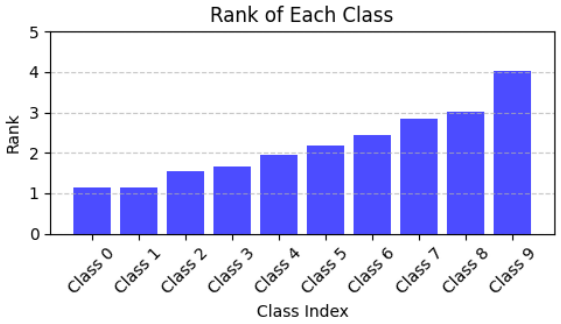


Figure 6: Average position index of the true label in the predictions sorted by confidence

ble. 3. Metrics include the macro, micro, and weighted averages of the F1 score. From increasing rank from 1 to 4, the F1 score also increases. But when the rank is higher than 8, the F1 score can slightly decrease. One possible reason is that a higher adapter rank increases the model's capacity, making it more prone to overfitting, which can lead to poorer performance on the test set.

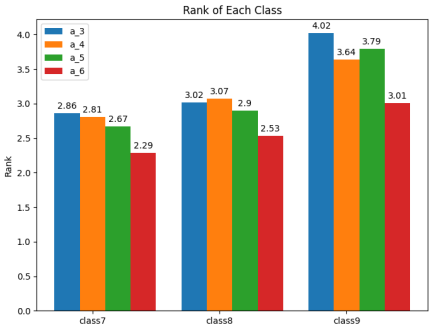
rank	macro-F1	micro-F1	weighted-F1
1	0.5051	0.5280	0.5051
2	0.5157	0.5363	0.5157
4	**0.5249**	**0.5398**	**0.5249**
8	0.5228	0.5397	0.5228

Table 3: The macro, micro, and weighted averages of F1 score under different rank

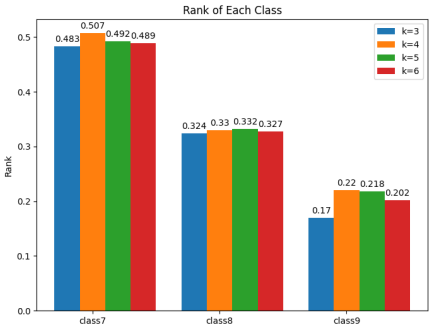
Top-k: The selection of k is important in our experiments due to the following reasons: if k is too large, in the extreme case, the CALoRA would degenerate to another linear layer in the model; however, if k is too small, the chosen top-k pseudo labels may not include the ground truth label.

We first explore the average index of the true label among predictions sorted by confidence. Results are shown in the Fig. 6. It can be observed that, for minority classes, the average position of the true label is around 4. Therefore, if k is set to a smaller value, such as 2 or 3, the correct label may not be included in the candidate set. As a result, some samples may not be routed to the appropriate CALoRA module, limiting its effectiveness in fine-tuning and performance enhancement.

Furthermore, we choose three minority classes (Class 7, 8, 9) to show the true label position and accuracy on the trained models, with k ranging from 3 to 6. The results are shown in Fig. 7. When k increases from 3 to 4, which is the average position



(a)



(b)

Figure 7: The experiment results of class 7, 8, 9(minority class) in average ranking and accuracy. (a)Avg prediction Rank of different k in the 3 minority classes; (b)Accuracy of different k in the 3 minority classes

index of samples in Class 9, the accuracy shows the greatest improvement. When k increases from 4 to 5, the average position index shows significant decrease. This indicates that it is important to make sure the ground truth label included in the top-k predictions. In comparison, when $k = 3$, which makes the ground truth label of Class 9 very likely to be excluded from the top-k predictions, the corresponding accuracy drops significantly. Table 4 shows the macro, micro, and weighted averages of F1 score under different k in Class 9. When k=4, the trained model achieves the optimal results on all three metrics.

k	macro-F1	micro-F1	weighted-F1
3	0.5405	0.5572	0.5405
4	**0.5508**	**0.5646**	**0.5508**
5	0.5490	0.5630	0.5490
6	0.5466	0.5621	0.5466

Table 4: The macro/micro/weighted and accuracy of different k

5 Discussion and future work

Although our current design has shown promising results, there remain important aspects to be further explored.

5.1 Validate Fairness Issues in Finetuning

To learn the fairness implications of class imbalance, we plan to conduct the experiments to illustrate that training data with imbalanced classes can compromise model performance.

Specifically, we assume a pretrained classifier classifier f_0 on dataset D_0, which is composed two datasets, D_1 and D_2, with imbalanced number of samples ($|D_1| > |D_2|$). Fine-tuning f_0 on D_1 and D_2 yield model f_1 and f_2, respectively. Denote the accuracy of model f_i regarding dataset D_j as a_{ij}. We expect to observe:

- $a_{01} > a_{02}$, indicating that a pre-trained model has better performance on the major class.
- $a_{12} < a_{02}$, $a_{21} < a_{01}$, meaning that fine-tuning model on one class can compromise its performance on another class.

These results would indicate that fine-tuning model purely on minor class samples can potentially harm the performance on major group, which can be undesirable in applications like health care. Such observation illustrating the necessities of using decoupled LoRA matrices to mitigate cross-class interference.

5.2 Evaluation on Additional Datasets

We also plan to validate our proposed method on two NLP binary classification tasks with significant class imbalance:

- **Incident detection.** Evaluated on Incident-Related Tweet3 (**IRT**) dataset (Schulz et al., 2016). IRT dataset contains tweets not related or related with incidents.
- **Sentiment classification.** Conducted on **Amazon Review** dataset (He and McAuley, 2016), which collects reviews that are either positive or negative.

Both dataset have a highly skewed distribution regarding samples from different classes.

We still need to compare our method against two baselines:

- **SetConv** (Gao et al., 2020), which uses representation learning to form a balanced distribution.
- **EDA** (Wei and Zou, 2019), which uses data augmentation to mitigate class imbalance.

Two metrics will be collected to compare model performance: Accuracy and F1-measure (F1).

6 Conclusion

In this work, we present Class-Adaptive LoRA, a lightweight and flexible framework designed to address class imbalance. By dynamically routing inputs to class-specific low-rank adaptation modules, our approach enables better specialization for minority classes with low overhead.

Our experimental results on imbalanced datasets demonstrate that Class-Adaptive LoRA benefit underrepresented classes. In future work, we aim to further investigate the fairness implications of fine-tuning under imbalanced settings, validate our method on a broader range of real-world tasks.

Contribution

All authors contributed equally to this project. The following tasks were collaboratively completed by the team:

- **Idea and Method Design:** Ding Zhu proposed the core concept of using class-aware LoRA for the problem of class imbalance. Xiukun Wei and Zhongteng Cai contributed to shaping the final framework through collaborative brainstorming and refinement.
- **Implementation and Experiments**: Ding Zhu was primarily responsible for experimental design and execution. Xiukun Wei and Zhongteng Cai focused on analyzing the experimental results.
- **Presentation**: Xiukun Wei made the presentation slides. Zhongteng Cai gave the presentation. Ding Zhu improved and finalized the slides and the speaking notes.
- **Writing**: Ding Zhu drafted Sections 3, 4, and 5, and improved all the other section. Xiukun Wei drafted Sections 1, 2, and 6, and contributed to improving Sections 3 and 5. Zhongteng Cai contributed to polishing and refining all the sections.

Acknowledgements

We gratefully acknowledge the use of OpenAI's ChatGPT for assistance in language refinement and grammatical corrections. However, the core ideas, arguments, experimental design, and conclusions presented in the paper are entirely the responsibility of the authors.

References

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Yang Gao, Yi-Fan Li, Yu Lin, Charu Aggarwal, and Latifur Khan. 2020. SetConv: A New Approach for Learning from Imbalanced Data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1284–1294, Online. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *Preprint*, arXiv:1512.03385.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 507–517, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

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. *Preprint*, arXiv:2210.04675.

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

Joel Jang, Yoonjeon Kim, Kyoungho Choi, and Sungho Suh. 2021. Sequential targeting: A continual learning approach for data imbalance in text classification. *Expert Systems with Applications*, 179:115067.

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.

Alex Krizhevsky, Geoffrey Hinton, and 1 others. 2009. Learning multiple layers of features from tiny images.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. 2024a. Mixlora: Enhancing large language models fine-tuning with lora-based mixture of experts. *Preprint*, arXiv:2404.15159.

Le Li, Jiale Wei, Pai Peng, Qiyuan Chen, Benjamin Guedj, and Bo Cai. 2024b. Reprint: a randomized extrapolation based on principal components for data augmentation. *Preprint*, arXiv:2204.12024.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2018. Focal loss for dense object detection. *Preprint*, arXiv:1708.02002.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1101–1111, New Orleans, Louisiana. Association for Computational Linguistics.

Samira Pouyanfar, Yudong Tao, Anup Mohan, Haiman Tian, Ahmed S. Kaseb, Kent Gauen, Ryan Dailey, Sarah Aghajanzadeh, Yung-Hsiang Lu, Shu-Ching Chen, and Mei-Ling Shyu. 2018. Dynamic sampling in convolutional neural networks for imbalanced data classification. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 112–117.

Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. 2020. Balanced meta-softmax for long-tailed visual recognition. *Preprint*, arXiv:2007.10740.

Axel Schulz, Christian Guckelsberger, and Frederik Janssen. 2016. Semantic abstraction for generalization of tweet classification: An evaluation of incident-related tweets. *Semantic Web*, 8(3):353–372.

Naama Tepper, Esther Goldbraich, Naama Zwerdling, George Kour, Ateret Anaby Tavor, and Boaz Carmeli. 2020. Balancing via generation for multi-class text classification improvement. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1440–1452, Online. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Zhuoyi Yang, Ming Ding, Yanhui Guo, Qingsong Lv, and Jie Tang. 2022. Parameter-efficient tuning makes a good classification head. *arXiv preprint arXiv:2210.16771*.

Jiaxin Zhang, Jie Liu, Shaowei Chen, Shaoxin Lin, Bingquan Wang, and Shanpeng Wang. 2022. Adam: An attentional data augmentation method for extreme multi-label text classification. In *Advances in Knowledge Discovery and Data Mining: 26th Pacific-Asia Conference, PAKDD 2022, Chengdu, China, May 16–19, 2022, Proceedings, Part I*, page 131–142, Berlin, Heidelberg. Springer-Verlag.