

# BALANCING MODEL EFFICIENCY AND PERFORMANCE: ADAPTIVE PRUNER FOR LONG-TAILED DATA

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Long-tailed distribution datasets are prevalent in many machine learning tasks, yet existing neural network models still face significant challenges when handling such data. This paper proposes a novel adaptive pruning strategy, LTAP (Long-Tailed Adaptive Pruner), aimed at balancing model efficiency and performance to better address the challenges posed by long-tailed data distributions. LTAP introduces multi-dimensional importance scoring criteria and designs a dynamic weight adjustment mechanism to adaptively determine the pruning priority of parameters for different classes. By focusing on protecting parameters critical for tail classes, LTAP significantly enhances computational efficiency while maintaining model performance. This method combines the strengths of long-tailed learning and neural network pruning, overcoming the limitations of existing approaches in handling imbalanced data. Extensive experiments demonstrate that LTAP outperforms existing methods on various long-tailed datasets, achieving a good balance between model compression rate, computational efficiency, and classification accuracy. This research provides new insights into solving model optimization problems in long-tailed learning and is significant for improving the performance of neural networks on imbalanced datasets. The code is available at <https://anonymous.open.science/r/AEFCD AISJ/README.md>.

## 1 INTRODUCTION

Long-tailed learning aims to address the problem of highly imbalanced class distributions, where most classes (i.e., tail classes) have scarce samples, while few classes (i.e., head classes) have abundant samples Liu et al. (2019); Wang et al. (2017); Tan et al. (2020); Li et al. (2020). This data distribution is prevalent in real-world applications, such as e-commerce product classification, speech recognition, and natural language processing Ouyang et al. (2016); Yang & Xu (2020). Although deep learning models perform excellently on head classes, their performance on tail classes remains limited, mainly because models tend to overfit head classes and neglect feature learning for tail classes, leading to insufficient overall model generalization Kang et al. (2019).

To improve long-tailed learning performance, researchers have proposed various methods, including multi-expert systems and modular designs Wang et al. (2020); Liu et al. (2019). However, these traditional methods face numerous challenges in practical applications. For instance, multi-expert systems often require training and maintaining multiple independent sub-models, resulting in enormous computational and storage resource consumption Xiang et al. (2020). Modular designs rely on predefined module structures, lacking dynamic adaptability and struggling to cope with constantly changing data distributions Ren et al. (2020). Moreover, *these methods often struggle to efficiently utilize parameters when dealing with tail classes, limiting model performance on scarce data Zhang et al. (2021a).*

As a model compression and optimization technique, pruning optimizes model structure and improves computational efficiency by removing redundant or unimportant neurons or connections Han et al. (2015); Liu et al. (2018); Zhu & Gupta (2017). In recent years, pruning methods have shown significant advantages in improving model performance, reducing parameter counts, and accelerating inference Frankle & Carbin (2018); Blalock et al. (2020). However, ordinary pruning methods face special **challenges** when applied to long-tailed learning:

- *Pruning bias due to class imbalance*: conventional pruning methods, blind to class-specific contributions, risk exacerbating the very imbalance they aim to address by inadvertently removing neurons crucial for tail class recognition He et al. (2021a).
- *Difficulty in dynamic adjustment*: the static nature of traditional pruning methods conflicts with the dynamic evolution of the association between parameters and data distribution during training, potentially leading to suboptimal or even harmful pruning decisions Molchanov et al. (2019).
- *Single evaluation criterion*: the reliance on simplistic pruning criteria fails to capture the nuanced importance of neurons in the complex landscape of long-tail distributions, potentially misleading the pruning process Frankle et al. (2020).

To address these challenges, this paper proposes a novel pruning strategy called **Long-Tailed Adaptive Pruner (LTAP)**, specifically optimized for long-tailed learning environments. LTAP is rooted in the understanding that effective long-tailed learning requires a fundamental rethinking of how we allocate and utilize model capacity Kang et al. (2019). Our method makes **innovative contributions** in the following aspects:

- *New LT-Vote mechanism*: Through the **LT-Vote** (Long-Tailed Voting) mechanism, we dynamically adjust the weights of different pruning criteria based on the classification accuracy of different classes, enabling the pruning process to more specifically optimize the learning performance of tail classes, enhancing model robustness on long-tail distribution data.
- *Multi-stage dynamic pruning*: Our method divides the pruning process into multiple stages, gradually removing redundant parameters, and dynamically adjusts the pruning strategy at each stage based on current model performance, ensuring continuous performance optimization during the pruning process.
- *Efficient resource utilization*: By reducing model parameter count and computational requirements through pruning, we improve model operational efficiency in resource-constrained environments while maintaining or even improving classification accuracy on tail classes.

Experimental results show that our proposed LTAP method significantly outperforms traditional pruning methods and other long-tailed learning methods on multiple long-tailed distribution datasets, validating its effectiveness in enhancing tail class recognition ability, optimizing model structure, and improving computational efficiency.

## 2 LTAP: ADAPTIVE PRUNER FOR LONG-TAILED DISTRIBUTION

In this section, we propose a novel pruning strategy called Long-Tailed Adaptive Pruner (LTAP), aimed at optimizing neural network models on long-tailed distribution datasets. LTAP effectively protects critical parameters of tail classes and enhances overall model performance and parameter efficiency in long-tailed distribution scenarios by integrating multiple importance scoring criteria and dynamically adjusting pruning weights. The following subsections will provide a detailed explanation of the overall architecture, LTAP optimizer design, pruning strategy implementation, and the alternating process of training and pruning.

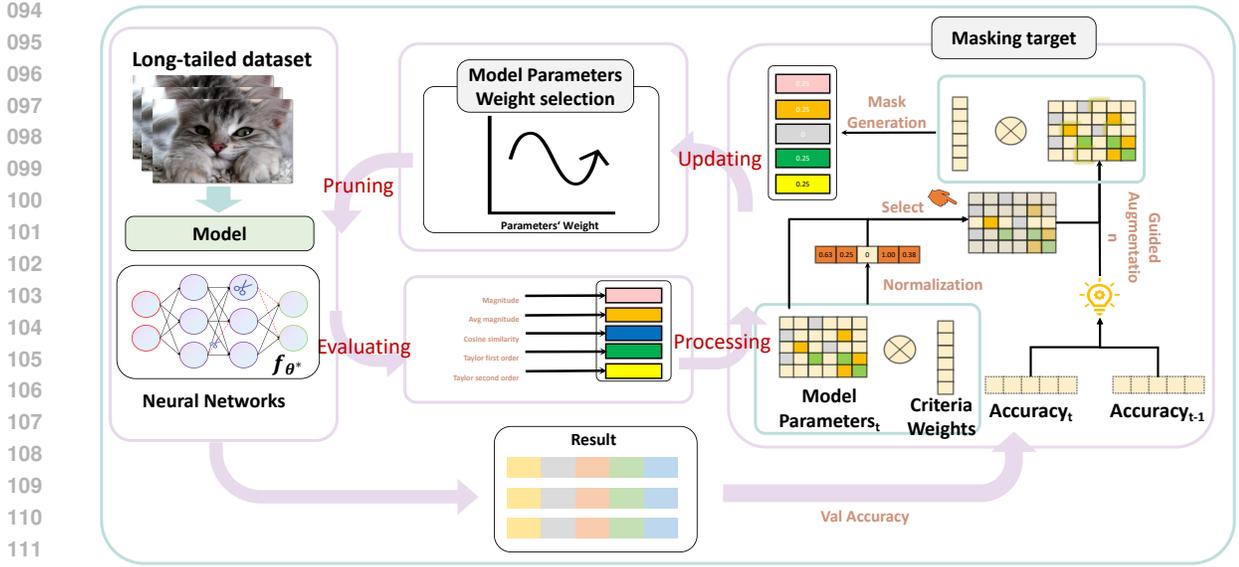


Figure 1: Overview of the Long-Tailed Adaptive Pruner (LTAP) methodology. The diagram illustrates the iterative process of model evaluation, parameter processing, and pruning, highlighting the integration of multiple importance criteria and the dynamic weight adjustment mechanism.

## 2.1 LTAP OPTIMIZER DESIGN

The core of LTAP lies in its optimizer, which evaluates the importance of each parameter group by synthesizing multi-dimensional importance scoring criteria and dynamically adjusts scoring weights based on class classification accuracy. Specifically, the optimization objective function of the LTAP optimizer is defined as:

$$\mathcal{L} = \sum_{c=1}^C w_c \mathcal{L}_c(\theta) + \lambda \|\mathbf{m}\|_0 \quad (1)$$

where  $C$  is the total number of classes,  $w_c$  is the weight of class  $c$ ,  $\mathcal{L}_c(\theta)$  is the loss function for class  $c$ ,  $\lambda$  is a hyperparameter controlling pruning intensity, and  $\|\mathbf{m}\|_0$  is the  $L_0$  norm of the parameter mask  $\mathbf{m}$ , representing the number of retained parameters.

To accurately assess parameter importance, LTAP introduces multiple scoring criteria, including *magnitude*, *cosine similarity*, *Taylor first order*, and *Taylor second order*. The comprehensive importance score  $S_g$  for each parameter group  $g$  is calculated by the following formula:

$$S_g = \sum_{k=1}^K \alpha_k \cdot s_{g,k} \quad (2)$$

where  $K$  is the number of scoring criteria,  $\alpha_k$  is the weight coefficient for scoring criterion  $k$ , and  $s_{g,k}$  is the score value of scoring criterion  $k$  for parameter group  $g$ . The specific scoring criteria are defined as follows:

$$s_{g,\text{magnitude}} = \|\mathbf{w}_g\|_2, \quad s_{g,\text{cosine}} = \frac{\mathbf{w}_g \cdot \mathbf{w}_{\text{ref}}}{\|\mathbf{w}_g\|_2 \|\mathbf{w}_{\text{ref}}\|_2}, \quad (3)$$

$$s_{g,\text{taylor-first}} = \left| \frac{\partial \mathcal{L}}{\partial \mathbf{w}_g} \right| \cdot \mathbf{w}_g, \quad s_{g,\text{taylor-second}} = \left| \frac{\partial^2 \mathcal{L}}{\partial \mathbf{w}_g^2} \right| \cdot \mathbf{w}_g^2 \quad (4)$$

where  $\mathbf{w}_g$  is the weight vector of parameter group  $g$ , and  $\mathbf{w}_{\text{ref}}$  is the reference weight vector used to measure the similarity between parameter group  $g$  and the reference model. Through these multi-dimensional scoring criteria, the LTAP optimizer can comprehensively evaluate the importance of each parameter group, thereby implementing a precise pruning strategy.

## 2.2 PRUNING STRATEGY IMPLEMENTATION

To adapt to the different needs of various classes in long-tailed distributions, LTAP introduces the **LT-Vote** (Long-Tailed Voting) mechanism. This mechanism dynamically adjusts the weight coefficients  $\alpha_k$  of scoring criteria based on the classification accuracy  $A_c$  of each class, ensuring that *tail classes receive more parameter protection during the pruning process*. The weight coefficient update formula is:

$$\alpha_k^{(t+1)} = \frac{\alpha_k^{(t)} \cdot \exp\left(\gamma \cdot \left(\frac{A_{c,\text{target}} - A_c}{A_c}\right) \cdot w_k\right)}{\sum_{k'=1}^K \alpha_{k'}^{(t)} \cdot \exp\left(\gamma \cdot \left(\frac{A_{c,\text{target}} - A_c}{A_c}\right) \cdot w_{k'}\right)} \quad (5)$$

where  $\gamma$  is an adjustment parameter controlling the sensitivity of weight updates,  $A_{c,\text{target}}$  is the target classification accuracy for class  $c$ , and  $w_k$  is the baseline weight for scoring criterion  $k$ . Through this formula, *the LT-Vote mechanism can dynamically adjust the weights of various scoring criteria based on the actual performance of each class, strengthening the protection of parameters for tail classes*.

LTAP adopts a multi-stage dynamic pruning strategy, distributing the overall pruning ratio  $\gamma_{\text{total}}$  across multiple pruning stages, with the pruning ratio for each stage being:

$$\gamma_p = \frac{\gamma_{\text{total}}}{P} \quad (6)$$

where  $P$  is the number of pruning stages. This strategy ensures the progressiveness and stability of pruning, avoiding performance fluctuations caused by one-time large-scale pruning. The pruning process mainly includes the following steps:

- (i) *Calculate importance scores*: compute comprehensive importance scores  $S_g$  for all parameter groups based on current model parameters.
- (ii) *Identify redundant parameter groups*: select parameter groups with the lowest importance scores as redundant groups based on the preset pruning ratio  $\gamma_p$ .
- (iii) *Prune redundant parameter groups*: achieve parameter pruning by setting the weights of redundant parameter groups to zero.
- (iv) *Update parameter distribution*: reallocate optimization strategies for remaining parameters based on the pruned model structure.

## 2.3 ALTERNATING PROCESS OF TRAINING AND PRUNING

The entire pruning process is formalized through the following optimization steps:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_{\theta} \mathcal{L}(\theta^{(t)}) \quad (7)$$

where  $\eta$  is the learning rate,  $\mathcal{L}$  is the loss function, and  $\theta$  represents model parameters. The training and pruning processes alternate, with the specific workflow as follows:

- (i) *Training phase*: perform forward and backward propagation under current model parameters, updating model parameters to minimize the loss function  $\mathcal{L}$ .

- 188 (ii) *Pruning phase*: execute pruning operations after specific training cycles, pruning low-importance  
 189 parameter groups based on current parameter importance scores.  
 190  
 191 (iii) *Update importance scores*: utilize the LT-Vote mechanism to adjust weight coefficients  $\alpha_k$  of scoring  
 192 criteria based on the performance of the pruned model on the validation set.  
 193 (iv) *Repeat iteration*: cycle through training and pruning phases until reaching the predetermined pruning  
 194 ratio or training rounds.

195 During the parameter update process, the LTAP optimizer combines gradient variants with pruning strategies,  
 196 with specific steps as follows:  
 197

$$198 \nabla_g S_g = \sum_{k=1}^K \alpha_k \cdot \nabla_g S_{g,k}, \quad \theta_g \leftarrow \theta_g - \eta \cdot \nabla_g S_g, \quad \theta_g \leftarrow \theta_g \odot (1 - \mathbf{m}_g) \quad (8)$$

199 where  $\nabla_g S_g$  is the gradient variant of parameter group  $g$  under different scoring criteria, and  $\mathbf{m}_g$  is the  
 200 pruning mask vector for parameter group  $g$ . Through these steps, the LTAP optimizer can incorporate  
 201 pruning strategies into the parameter update process, achieving dynamic compression and optimization of  
 202 model parameters. Meanwhile, the LT-Vote mechanism ensures that the model provides sufficient protection  
 203 for parameters of tail classes during the pruning process, thereby enhancing classification performance.  
 204  
 205

### 206 3 THEORETICAL ANALYSIS: TAIL CLASSES BENEFIT MORE FROM 207 OVERPARAMETERIZATION 208 209

210 To verify the effectiveness and soundness of our method, we first establish a series of foundational defini-  
 211 tions. Then, through lemmas and theorems, we systematically argue that tail classes in long-tailed distribu-  
 212 tions have higher requirements for model overparameterization. Based on this, we propose a differentiated  
 213 parameter allocation strategy and the tail-biased pruning proposition. Finally, we synthesize these theo-  
 214 retical results and prove the effectiveness of the tail-biased pruning strategy in learning from long-tailed  
 215 distributions.  
 216

217 To gain a deeper understanding of the learning difficulty of different classes under long-tailed distributions,  
 218 we introduce the sample complexity lemma (Lemma 1 in Appendix), establishing the relationship between  
 219 sample size, VC dimension, and generalization error. Based on this lemma, we further define the class-  
 220 specific VC dimension (Definition 1 in Appendix) and derive the learning difficulty for each class (Corollary  
 221 1 in Appendix).

222 **Theorem 1** (Differentiated Overparameterization Demand). *In a long-tailed setting, to achieve the same*  
 223 *generalization performance, tail classes require a higher degree of overparameterization than head classes.*  
 224 *Specifically, for tail classes,  $\gamma_c \geq \Omega\left(\frac{N_1}{N_c} \cdot \frac{1}{\log N_c}\right)$ , and for head classes,  $\gamma_c \sim O(1)$ , where  $N_1$  is the*  
 225 *number of samples in the head class (the class with the most samples), and  $N_c$  is the number of samples in*  
 226 *class  $c$ .*

227 The differentiated overparameterization demand theorem reveals that tail classes in a long-tailed dataset have  
 228 a higher demand for model overparameterization, which is further supported by the imbalance in overpa-  
 229 rameterization demand (Corollary 2 in Appendix).  
 230

231 Based on the differentiated overparameterization demand theorem, we propose the parameter allocation  
 232 strategy corollary, which guides how to reasonably allocate model parameters in long-tailed learning.

233 **Theorem 2** (Parameter Allocation Strategy). *In long-tailed learning, to optimize overall model performance,*  
 234 *relatively more parameters should be allocated to tail classes. Specifically, for class  $c$ , the ideal parameter*

allocation ratio  $\alpha_c$  should satisfy  $\alpha_c \propto \frac{N_1}{N_c} \cdot \frac{1}{\log N_c}$ , where  $N_1$  is the number of samples in the head class (the class with the most samples), and  $N_c$  is the number of samples in class  $c$ .

**Theorem 3** (Performance Gains from Parameter Allocation). *Assume that model performance is logarithmically related to the number of effective parameters for each class, i.e., for class  $c$ , its performance  $\text{perf}_c$  satisfies  $\text{perf}_c \propto \log P_c$ , where  $P_c$  is the number of effective parameters for class  $c$ . Under the above parameter allocation strategy, compared to uniform allocation, the performance gain  $\Delta$  is  $\Delta \geq \Omega\left(\frac{1}{C} \sum_{c=1}^C \log\left(\frac{N_1}{N_c}\right)\right)$ , where  $C$  is the total number of classes.*

This theorem shows that by reasonably allocating parameters, we can significantly improve overall model performance on long-tailed datasets, especially improving the classification accuracy of tail classes.

Based on the above theoretical foundations, we propose the ***tail-biased pruning proposition, which guides how to prune models in long-tailed learning.***

**Proposition 1** (Tail-biased Pruning). *In long-tailed learning, to reduce the number of parameters while maintaining overall model performance, a pruning strategy that favors retaining parameters for tail classes should be adopted. Specifically, the optimization objective is  $\min_{\mathbf{m}} \sum_{c=1}^C w_c L_c(f_{\theta \odot \mathbf{m}}(\mathbf{x}), \mathbf{y}) + \lambda \|\mathbf{m}\|_0$ , where  $\mathbf{m} \in \{0, 1\}^{|\theta|}$  is a binary mask vector indicating whether a parameter is retained,  $w_c \propto \frac{N_1}{N_c} \cdot \frac{1}{\log N_c}$  is the weight for class  $c$ ,  $L_c$  is the loss function for class  $c$ ,  $f_{\theta \odot \mathbf{m}}$  denotes the masked model,  $\lambda$  is a hyperparameter controlling the pruning strength, and  $\|\mathbf{m}\|_0$  is the  $L_0$  norm of  $\mathbf{m}$ .*

This proposition, through a weighted loss function and parameter sparsity, guides how to prioritize the retention of tail class parameters during pruning, ensuring that model parameters are reduced while maintaining or improving overall performance on long-tailed datasets.

To ensure the effectiveness of the tail-biased pruning strategy in practical applications, we propose the following performance guarantee theorem.

**Theorem 4** (Performance Guarantee of Tail-biased Pruning). *Assume that the initial model achieves a training error of  $\epsilon$  on each class. After applying the tail-biased pruning strategy, the expected generalization error  $\mathbb{E}[\hat{\epsilon}_c]$  for class  $c$  satisfies  $\mathbb{E}[\hat{\epsilon}_c] \leq \epsilon + O\left(\sqrt{\frac{\log(N_c/\delta)}{N_c}}\right)$ , where  $\delta$  is a small constant (e.g., 0.05), representing the confidence level.*

This theorem shows that despite the pruning process, the generalization error for tail classes can still be effectively controlled, and the strategy ensures that overall performance does not significantly degrade, especially in terms of the performance of tail classes. The relevant proofs can be found in **Appendix A**.

## 4 EXPERIMENTS

In this section, we evaluate the performance of our proposed method on multiple long-tailed datasets. Furthermore, we assess the computational efficiency of each method by comparing the ratio of floating-point operations (FLOPs) and the ratio of accuracy improvement.

### 4.1 EXPERIMENTAL SETUP

**Datasets.** CIFAR-100-LT is a long-tailed version of CIFAR-100, containing 100 classes with two imbalance ratios (IR = 50, 100). ImageNet-LT is a long-tailed version of ImageNet, with 1,000 classes and natural long-tailed distribution. iNaturalist 2018 is a large-scale real-world dataset with 8,142 species categories and inherent long-tailed distribution.

Table 1: Accuracy (%) on CIFAR-100-LT dataset (Imbalance ratio= $\{50, 100\}$ ).  $F$  denotes the ratio of FLOPs between the target method and the baseline method.  $C$  denotes the ratio of accuracy (acc) between the target method and the baseline method. The gray column represents the primary observed metrics, and the gray row indicates the baseline method for the current block.

Method	$F(\%) \downarrow$	IR = 50						IR = 100					
		Head $\uparrow$	Medium $\uparrow$	Tail $\uparrow$	All $\uparrow$	$C(\%) \uparrow$	$\frac{C}{F} \uparrow$	Head $\uparrow$	Medium $\uparrow$	Tail $\uparrow$	All $\uparrow$	$C(\%) \uparrow$	$\frac{C}{F} \uparrow$
BS Ren et al. (2020)	100.0	62.3	46.1	37.0	51.2	100.0	1.0	62.6	48.5	27.0	47.2	100.0	1.0
BS + ATOWu et al. (2024)	84.7	39.6	30.6	21.8	32.9	64.2	0.7	40.8	28.9	16.5	29.5	62.5	0.7
BS + RRegStewart et al. (2023)	35.1	51.3	35.6	23.8	40.2	78.5	2.2	53.0	36.3	19.0	37.3	79.0	2.3
BS + LTAP	22.6	57.6	43.4	34.1	47.8	93.3	4.1	55.8	44.7	23.2	42.4	89.8	3.9
LDAM-DRW Cao et al. (2019)	100.0	64.5	43.0	26.4	49.1	100.0	1.0	65.1	48.1	20.1	45.8	100.0	1.0
LDAM-DRW + ATOWu et al. (2024)	84.7	40.1	34.6	25.5	33.7	68.6	0.8	41.8	30.9	18.5	31.0	67.6	0.8
LDAM-DRW + RRegStewart et al. (2023)	35.1	52.9	39.8	23.7	42.4	86.3	2.4	54.3	37.8	16.7	37.6	82.0	2.3
LDAM-DRW + LTAP	24.8	58.8	39.9	23.3	44.8	91.2	3.6	56.9	40.2	18.8	39.8	86.8	3.5
DBLP Zhou et al. (2024)	100.0	61.2	46.5	32.3	50.2	100.0	1.0	61.4	46.9	23.6	45.3	100.0	1.0
DBLP + ATOWu et al. (2024)	84.7	50.7	37.0	26.2	38.5	76.6	0.9	40.8	32.6	21.4	32.1	70.8	0.8
DBLP + RRegStewart et al. (2023)	35.1	50.8	40.4	24.1	43.8	87.2	2.4	52.1	39.2	17.5	37.5	82.7	2.3
DBLP + LTAP	24.0	56.1	43.5	31.5	46.7	93.0	3.9	54.7	43.3	25.8	42.0	92.7	3.9

Table 2: Accuracy (%) on ImageNet-LT and iNaturalist 2018.  $F$  denotes the ratio of FLOPs between the target method and the baseline method.  $C$  denotes the ratio of accuracy (acc) between the target method and the baseline method. The gray column represents the primary observed metrics, and the gray row indicates the baseline method for the current block.

Method	$F(\%) \downarrow$	ImageNet-LT						iNaturalist 2018					
		Head $\uparrow$	Medium $\uparrow$	Tail $\uparrow$	All $\uparrow$	$C(\%) \uparrow$	$\frac{C}{F} \uparrow$	Head $\uparrow$	Medium $\uparrow$	Tail $\uparrow$	All $\uparrow$	$C(\%) \uparrow$	$\frac{C}{F} \uparrow$
BS Ren et al. (2020)	100.0	60.9	48.8	32.1	51.0	100.0	1.0	65.7	67.4	67.5	67.3	100.0	1.0
BS + ATOWu et al. (2024)	65.3	37.1	35.7	17.8	33.8	66.2	1.1	34.8	42.5	42.2	41.5	61.6	0.9
BS + RRegStewart et al. (2023)	52.1	41.1	36.0	18.2	35.5	69.6	1.3	30.5	45.1	44.8	43.4	64.4	1.2
BS + LTAP	30.6	58.5	45.0	30.1	48.1	81.9	2.6	59.2	60.7	60.7	60.5	89.8	2.9
LDAM-DRW Cao et al. (2019)	100.0	60.4	46.9	30.7	49.8	100.0	1.0	63.2	66.3	65.4	65.6	100.0	1.0
LDAM-DRW + ATOWu et al. (2024)	65.3	36.9	34.6	17.5	33.1	66.4	1.1	36.5	30.9	30.7	31.3	47.7	0.7
LDAM-DRW + RRegStewart et al. (2023)	52.1	38.8	36.8	18.7	35.1	70.4	1.3	41.8	32.5	32.0	33.2	50.6	0.9
LDAM-DRW + LTAP	30.8	57.8	41.9	23.3	45.4	91.1	2.9	59.9	60.2	60.4	60.2	91.7	2.9
DBLP Zhou et al. (2024)	100.0	61.7	47.1	30.3	50.4	100.0	1.0	65.0	66.9	65.6	66.1	100.0	1.0
DBLP + ATOWu et al. (2024)	65.3	37.9	35.0	17.5	33.7	66.8	1.1	41.0	30.6	30.4	31.5	47.6	0.7
DBLP + RRegStewart et al. (2023)	52.1	40.0	35.7	18.8	35.0	69.4	1.3	48.8	32.5	32.3	34.0	51.4	0.9
DBLP + LTAP	30.0	58.5	45.2	23.0	47.3	93.8	3.1	58.7	59.3	59.8	59.4	90.1	3.0

**Implementation Details.** We use the knowledge generated from the long-tailed recognition task to guide the pruning of the backbone network. Specifically, for each parameter in the model, we calculate scores using ‘magnitude’, ‘avg\_magnitude’, ‘cosine\_similarity’, ‘taylor\_first\_order’, and ‘taylor\_second\_order’ during the gradient descent process. These scores are then weighted based on the cumulative change in accuracy for each class on the validation set. The weighted sum of the scores is used to determine whether to prune a parameter. We start the continuous pruning process after the 100th epoch, and the final model retains 30% of the original parameters. For the final evaluation phase, we use the same settings as DODA Wang et al. (2024) for all baseline methods and our method. For the CIFAR-100-LT dataset, we follow the general experimental settings of Cao et al. (2019) and use ResNet-32 (proposed by He et al. (2016)) as the backbone network. The network is trained for 200 epochs using the GD optimizer with an initial learning rate of  $10^{-4}$ , momentum of 0.9, and weight decay of  $2 \times 10^{-4}$ . For ImageNet-LT and iNaturalist 2018 datasets, we use ResNet-50 as the backbone network, train the network for 100 epochs with an initial learning rate of 0.1, and reduce the learning rate by 0.1 at the 60th and 80th epochs. For all experiments, we set the value of the hyperparameter  $\text{pau}$  to 0.5.

**Baselines.** For fair comparison, all methods are evaluated under the same experimental conditions. We use three strong long-tailed baselines, e.g., Balanced Softmax (BS) Ren et al. (2020), LDAM-DRW Cao et al.

(2019), DBLP Zhou et al. (2024) and two SOTA pruning method, ATOWu et al. (2024), RRegStewart et al. (2023). Our proposed method is denoted as ‘BS + LTAP’, ‘LDAM-DRW + LTAP’, and ‘DBLP + LTAP’. We report the classification accuracy of the head, medium, and tail classes, as well as the overall accuracy across all classes. Additionally, we compute the ratio of accuracy to FLOPs ( $\frac{C}{F}$ ) as a key metric to evaluate both performance and computational efficiency.

## 4.2 BENCHMARK RESULTS

**CIFAR-100-LT.** Table 1 presents the classification results for different methods on the CIFAR-100-LT dataset under two imbalance ratios (IR = 50 and 100). LTAP consistently achieves higher  $\frac{C}{F}$  compared to other pruning methods in both imbalance ratio settings, demonstrating superior efficiency in terms of both accuracy and computational cost. For instance, in the [IR = 50] setting, LTAP achieves a tail accuracy of 34.1% compared to 23.8% by RReg, while reducing FLOPs by 77.4% compared to the baseline BS. The  $\frac{C}{F}$  ratio of 4.1 for LTAP is nearly double that of RReg (i.e., 2.2). Similar trends are observed for LDAM-DRW and DBLP, where LTAP consistently improves tail class accuracy and achieves the highest  $\frac{C}{F}$  ratios.

**ImageNet-LT and iNaturalist 2018.** Table 2 shows the results on ImageNet-LT and iNaturalist 2018 datasets. These larger and more complex datasets further validate the effectiveness of LTAP. On ImageNet-LT, ‘BS + LTAP’ achieves a tail accuracy of 30.1%, significantly outperforming ‘BS + RReg’ (i.e., 18.2%), while reducing FLOPs by 69.4%. The  $\frac{C}{F}$  ratio of 2.6 for ‘BS + LTAP’ is double that of ‘BS + RReg’. For iNaturalist 2018, LTAP shows consistent performance across all classes (i.e., head, medium, and tail classes), indicating its robustness in handling extreme class imbalance. Notably, LTAP maintains high accuracy across all class types while significantly reducing FLOPs. For example, on ImageNet-LT, ‘DBLP + ours’ reduces FLOPs by 70% while achieving 93.8% of the baseline accuracy, resulting in a  $\frac{C}{F}$  ratio of 3.1.

**Efficiency Evaluation.** Across all datasets, our method achieves a significant reduction in FLOPs while maintaining competitive or superior accuracy. For ImageNet-LT and iNaturalist 2018, our method consistently reduces FLOPs by about 70% compared to the baselines, while achieving the highest  $\frac{C}{F}$  ratios. This reduction in computational cost, coupled with maintained or improved accuracy, demonstrates the practical utility of our method for resource-constrained environments where high accuracy is required.

## 4.3 FURTHER ANALYSIS

In this section, we conduct a detailed analysis of the mechanism of LTAP and discuss the following issues. More empirical results are reported in **Appendix C**.

**Discussion 1:** How are neurons masked under different pruning strategies?

Figure 2 illustrates how neurons are masked under different pruning strategies. First, the visualization of w.o. vote shows that after removing the long-tailed feedback mechanism, the flexibility of the pruning process decreases significantly, with pruning limited to specific rows of neurons. This rigid pruning strategy restricts the model’s adaptability to varying data distributions, especially in handling long-tailed data, where it struggles to preserve neurons critical for tail classes. In contrast, our proposed LTAP method, as shown in the  $\gamma = 0.7$  visualization, exhibits much greater flexibility.

Pruning is no longer confined to specific rows of neurons but instead dynamically adjusts based on the parameter values of neurons across different layers and positions. This adaptive pruning strategy allows the model to better retain neurons that are critical for tail classes, improving classification accuracy on long-tail data. Additionally, as the  $\gamma$  value increases, we observe that the pruning intensity increases, with more neurons being masked. However, the distribution of pruning remains dynamic and flexible. This further demonstrates that the LTAP method can maintain both efficiency and effectiveness under varying levels of pruning

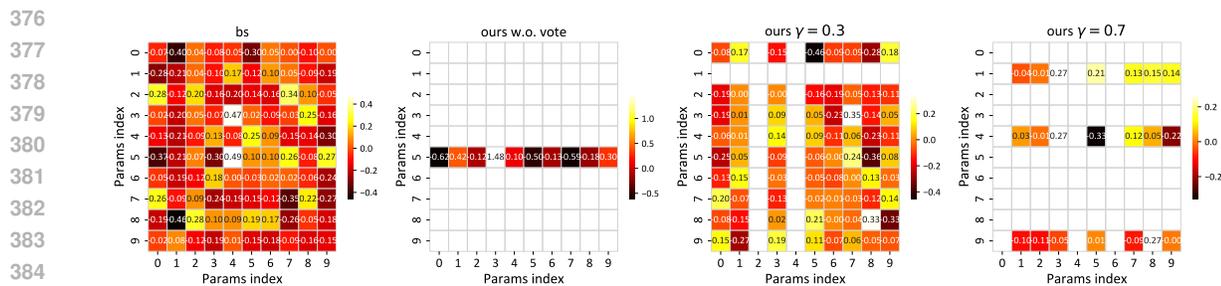


Figure 2: Visualization of neuron masking. Each small cell represents the sum of the parameters of a  $3 \times 3$  convolutional kernel, and each subfigure represents a cross-section of a layer of neurons. Each layer should contain  $1 \times 64 \times 64 \times 3 \times 3$  convolutional kernels, and we visualize the top-left  $1 \times 10 \times 10 \times 3 \times 3$  part of each layer. The values and colors represent the sum of the parameters in the  $3 \times 3$  convolutional kernels, and the blank areas indicate neurons that have been masked. The variable  $\gamma$  represents the masking ratio, and w.o. vote denotes the removal of the long-tailed feedback mechanism LT-Vote.

intensity. In summary, the LTAP method achieves more precise neuron importance estimation through the long-tailed feedback mechanism, balancing computational efficiency and classification performance during the pruning process.

### Discussion 2: Dynamic changes in the performance of different pruning methods during training.

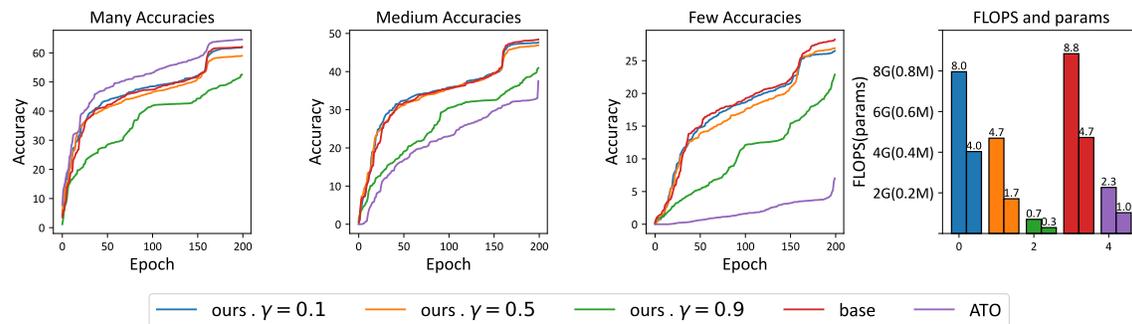


Figure 3: Training dynamics comparison under different pruning strategies. The first three subplots show the accuracy changes over training epochs for the head, medium, and tail classes, respectively. Different colored curves represent various pruning intensities, with the purple curve (ATO) representing the baseline for non-long-tailed pruning methods. The last subplot shows the comparison of FLOPs and parameter counts across different pruning strategies ( $\gamma = 0.1, 0.5, \text{ and } 0.9$ ) as well as the baseline (BS) and ATO methods.

From Figure 3, it is evident that different pruning strategies exhibit significant performance differences on long-tailed data. We analyze from the perspective of different classes.

423 **Accuracy on Many Classes.** Even with a pruning ratio of  $\gamma = 0.9$ , LTAP maintains a high accuracy. Mean-  
424 while, the ATO shows slightly better performance in this region, which indicates that traditional pruning  
425 exacerbates the imbalance in long-tailed distributions.

426 **Accuracy on Medium Classes.** LTAP continues to maintain high accuracy at pruning ratios of  $\gamma = 0.5$   
427 and  $\gamma = 0.9$ , following a similar trend as the head classes. In contrast, the accuracy of the ATO baseline  
428 significantly decreases, and it is even surpassed by LTAP at  $\gamma = 0.9$ . This suggests that traditional pruning  
429 methods fail in long-tailed learning scenarios due to their excessive focus on head classes.

430 **Accuracy on Tail Classes.** Tail classes pose the biggest challenge in long-tailed learning. Traditional prun-  
431 ing methods (e.g., ATO) perform disastrously on the tail classes, suffering a catastrophic drop in accuracy,  
432 which reflects their extreme inability to adapt to long-tailed classes. In comparison, our pruning method,  
433 even at a high pruning ratio ( $\gamma = 0.9$ ), maintains strong performance, demonstrating its robust adaptability  
434 to tail classes.

435 **FLOPs and Parameter Comparison.** The last subplot shows the comparison of FLOPs and parameter  
436 counts under different pruning strategies. Our method allows for varying degrees of pruning, and even at a  
437 pruning ratio of  $\gamma = 0.9$ , it maintains high average performance and strong performance on tail classes. At  
438 this point, compared to the baseline, it demonstrates significant advantages in both computational efficiency  
439 and performance.

440 The comparisons above demonstrate that our method not only *reduces parameter and computational costs*  
441 *while maintaining high performance*, but also *adapts effectively to different frequency classes*. Notably, it  
442 shows a significant advantage over traditional methods, particularly in handling tail classes.

## 443 5 CONCLUSION

444 We have presented LTAP, a dynamic pruning strategy designed to enhance model efficiency and perfor-  
445 mance on long-tailed datasets. By dynamically adjusting pruning criteria based on class-specific perfor-  
446 mance, LTAP addresses the inherent pruning bias in conventional methods, particularly for tail classes.  
447 Our theoretical analysis establishes that tail classes benefit more from model overparameterization, which  
448 informs our tail-biased pruning approach. Extensive experiments on benchmark long-tailed datasets, in-  
449 cluding CIFAR-100-LT, ImageNet-LT, and iNaturalist 2018, demonstrate that LTAP consistently improves  
450 classification accuracy, particularly for tail classes, while significantly reducing the model’s computational  
451 overhead. By offering a balanced trade-off between model compression and accuracy, LTAP provides a  
452 robust solution to the challenges of long-tailed learning and opens new possibilities for optimizing neural  
453 networks in imbalanced and resource-constrained environments.

## 454 REFERENCES

- 455 Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural  
456 network pruning? *arXiv preprint arXiv:2003.03033*, 2020.
- 457 Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with  
458 label-distribution-aware margin loss. In *Advances in neural information processing systems*, volume 32,  
459 2019.
- 460 Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority  
461 over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.
- 462 Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effec-  
463 tive number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern*  
464 *recognition*, pp. 9268–9277, 2019.

- 470 Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural net-  
471 works. *arXiv preprint arXiv:1803.03635*, 2018.
- 472 Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. Linear mode connectivity  
473 and the lottery ticket hypothesis. *arXiv preprint arXiv:1912.05671*, 2020.
- 474
- 475 Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient  
476 neural network. *Advances in neural information processing systems*, 28, 2015.
- 477
- 478 Chengjin He, Shengping Wang, Dongmin Zhang, and Ming Zeng. Rethinking class-balanced methods for  
479 long-tailed visual recognition from a domain adaptation perspective. In *Proceedings of the IEEE/CVF*  
480 *Conference on Computer Vision and Pattern Recognition*, pp. 7610–7619, 2021a.
- 481 Haibo He, Yang Bai, Eduardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach  
482 for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world*  
483 *congress on computational intelligence)*, pp. 1322–1328. IEEE, 2008.
- 484 Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition.  
485 In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- 486
- 487 Yin-Yin He, Jianxin Wei, Xuming Zhu, Xiaopeng Wang, Zhenguo Feng, and Jianqiang Dong. Distilling  
488 virtual examples for long-tailed recognition. In *Proceedings of the IEEE/CVF International Conference*  
489 *on Computer Vision*, pp. 13173–13182, 2021b.
- 490 Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalan-  
491 tidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on*  
492 *Learning Representations*, 2019.
- 493 Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient  
494 convnets. In *arXiv preprint arXiv:1608.08710*, 2016.
- 495
- 496 Yifan Li, Tong Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Long-tailed  
497 classification by keeping the good and removing the bad momentum causal effect. In *Advances in Neural*  
498 *Information Processing Systems*, volume 33, pp. 1513–1524, 2020.
- 499 Ji Lin, Yongming Rao, Jiwen Lu, and Jie Zhou. Dynamic neural networks: A survey. In *IEEE Transactions*  
500 *on Pattern Analysis and Machine Intelligence*, 2020.
- 501
- 502 Jialin Liu, Fuxun Ye, Xinjian Zhang, Hui Xu, and Yu Cheng. Dynamic sparse training: Find efficient  
503 sparse network from scratch with trainable masked layers. In *International Conference on Learning*  
504 *Representations*, 2021a.
- 505 Yufei Liu, Yao Zhang, Hao Wang, Juanzi Fan, and Zhiqiang Zhang. Learning to learn the future: Modeling  
506 concept drift in continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*,  
507 volume 35, pp. 8415–8423, 2021b.
- 508 Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning  
509 efficient convolutional networks through network slimming. In *Proceedings of the IEEE international*  
510 *conference on computer vision*, pp. 2736–2744, 2017.
- 511
- 512 Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network  
513 pruning. *arXiv preprint arXiv:1810.05270*, 2018.
- 514 Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-  
515 tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision*  
516 *and Pattern Recognition*, pp. 2537–2546, 2019.

- 517 Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural  
518 networks for resource efficient inference. In *International Conference on Learning Representations*, 2016.  
519
- 520 Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for  
521 neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern  
522 Recognition*, pp. 11264–11272, 2019.
- 523 Wanli Ouyang, Xiaogang Wang, Cong Zhang, and Xiaokang Yang. Factors in finetuning deep model for  
524 object detection with long-tail distribution. In *Proceedings of the IEEE Conference on Computer Vision  
525 and Pattern Recognition*, pp. 864–873, 2016.  
526
- 527 Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual  
528 recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020.
- 529 James Stewart, Umberto Michieli, and Mete Ozay. Data-free model pruning at initialization via expanders.  
530 In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Work-  
531 shops*, June 2023.  
532
- 533 Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equal-  
534 ization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF Conference on Computer  
535 Vision and Pattern Recognition*, pp. 11662–11671, 2020.  
536
- 537 Binwu Wang, Pengkun Wang, Wei Xu, Xu Wang, Yudong Zhang, Kun Wang, and Yang Wang. Kill two birds  
538 with one stone: Rethinking data augmentation for deep long-tailed learning. In *The Twelfth International  
539 Conference on Learning Representations*, 2024.
- 540 Xudong Wang, Ziwei Long, Ling Wang, and Liang Wang. Long-tailed recognition by routing diverse  
541 distribution-aware experts. In *International Conference on Learning Representations*, 2020.  
542
- 543 Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural  
544 Information Processing Systems*, pp. 7029–7039, 2017.
- 545 Xidong Wu, Shangqian Gao, Zeyu Zhang, Zhenzhen Li, Runxue Bao, Yanfu Zhang, Xiaoqian Wang, and  
546 Heng Huang. Auto-train-once: Controller network guided automatic network pruning from scratch. In  
547 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.  
548
- 549 Li Xiang, Guoqing Ding, and Jungong Han. Learning from multiple experts: Self-paced knowledge distil-  
550 lation for long-tailed classification. In *European Conference on Computer Vision*, pp. 247–263. Springer,  
551 2020.
- 552 Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. In *Ad-  
553 vances in Neural Information Processing Systems*, volume 33, pp. 19290–19301, 2020.  
554
- 555 Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Feature transfer learning for  
556 face recognition with under-represented data. In *Proceedings of the IEEE/CVF Conference on Computer  
557 Vision and Pattern Recognition*, pp. 5704–5713, 2019.
- 558 Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified  
559 framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer  
560 Vision and Pattern Recognition*, pp. 2361–2370, 2021a.  
561
- 562 Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A  
563 survey. In *arXiv preprint arXiv:2110.04596*, 2021b.

564 Zhipeng Zhou, Liu Liu, Peilin Zhao, and Wei Gong. Pareto deep long-tailed recognition: A conflict-averse  
565 solution. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna,*  
566 *Austria, May 7-11, 2024*. OpenReview.net, 2024. URL <https://openreview.net/forum?id=b66P1u0k15>.  
567  
568  
569 Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model  
570 compression. In *International Conference on Learning Representations*, 2017.  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610

## Appendix

### Balancing Model Efficiency and Performance: LT-Vote Pruning for Long-tailed Data

The content of the **Appendix** is summarized as follows:

- 1) in Sec. A, we provide detailed proofs and theoretical foundations for the results in the main paper.
- 2) in Sec. B, we briefly present the state of the art in the field of long-tailed learning and neural network pruning.
- 3) in Sec. C, we illustrate more detailed empirical results and analyses of LTAP.

#### A SUPPLEMENTARY THEORY

In this supplementary section, we provide detailed proofs and theoretical foundations for the main results presented in the paper. We start by introducing the sample complexity lemma and related definitions, which form the basis for understanding the learning difficulty of different classes in long-tailed distributions. Then, we prove the main theorems and propositions presented in the paper, including the differentiated overparameterization demand theorem, the parameter allocation strategy theorem, and the tail-biased pruning proposition. Finally, we provide the proof for the performance guarantee theorem, which ensures the effectiveness of the tail-biased pruning strategy in practical applications.

##### A.1 APPENDIX: SAMPLE COMPLEXITY AND DIFFERENTIATED OVERPARAMETERIZATION DEMAND

**Lemma 1** (Sample Complexity Lemma). *For class  $c$  in a binary classification problem, given hypothesis space  $\mathcal{H}_c$ , target generalization error  $\epsilon > 0$ , and confidence level  $1 - \delta$  ( $0 < \delta < 1$ ), the minimum required sample size  $N_c$  satisfies  $N_c \geq \frac{1}{2\epsilon^2} (4d_{VC,c} \log \frac{12}{\epsilon} + \log \frac{2}{\delta})$ , where  $d_{VC,c}$  is the VC dimension of the hypothesis space  $\mathcal{H}_c$  associated with class  $c$ .*

**Remark 1.** (i) *This bound shows that the required sample size is approximately linearly related to the VC dimension  $d_{VC,c}$  and inversely proportional to the square of the target generalization error  $\epsilon$ .* (ii) *In practical applications, we usually focus on asymptotic behavior, which can be simplified to  $N_c \geq \Omega\left(\frac{d_{VC,c}}{\epsilon^2}\right)$ .* (iii) *Although this lemma is for binary classification, it can be extended to multiclass problems through the one-vs-all strategy.*

**Definition 1** (Class-specific VC Dimension). *For class  $c$  in a long-tailed dataset, the class-specific VC dimension  $d_{VC,c}$  is defined as the VC dimension of the hypothesis space that can effectively separate that class from all other classes.*

**Corollary 1** (Class Learning Difficulty). *In a long-tailed dataset, the learning difficulty  $\mathcal{D}_c$  for class  $c$  can be approximated as  $\mathcal{D}_c \approx \frac{d_{VC,c}}{N_c}$ , where  $N_c$  is the number of samples in class  $c$ .*

This corollary shows that for head classes, where  $N_c$  is large, the learning difficulty  $\mathcal{D}_c$  is small. For tail classes, where  $N_c$  is small, even if  $d_{VC,c}$  remains the same or slightly smaller, the learning difficulty  $\mathcal{D}_c$  increases significantly. This difference in learning difficulty directly affects the model’s complexity (i.e., the number of parameters) required for different classes, as stated in the differentiated overparameterization demand theorem (Theorem 1).

**Corollary 2** (Imbalance in Overparameterization Demand). *In a long-tailed dataset, the demand for overparameterization is inversely proportional to the number of samples in each class. Specifically, for any two classes  $i$  and  $j$ , if  $N_i > N_j$ , then  $\frac{\gamma_i}{\gamma_j} \leq O\left(\frac{N_j \log N_i}{N_i \log N_j}\right)$ .*

This corollary further supports the differentiated overparameterization demand theorem (Theorem 1), revealing that in a long-tailed dataset, as the number of samples in a class decreases, the demand for overparameterization increases significantly, emphasizing the importance of providing more parameter protection for tail classes.

## A.2 PROOF FOR THEOREM

**Theorem 5** (Performance Guarantee for Tail-Biased Pruning). *Assume that the initial model achieves a training error of  $\epsilon$  for each class. After applying the tail-biased pruning strategy, the expected generalization error  $\mathbb{E}[\hat{\epsilon}_c]$  for class  $c$  satisfies*

$$\mathbb{E}[\hat{\epsilon}_c] \leq \epsilon + O\left(\sqrt{\frac{\log(N_c/\delta)}{N_c}}\right), \quad (9)$$

where  $\delta$  is a small constant (e.g., 0.05) representing the confidence level.

*Proof.* Let  $f_\theta$  denote the initial model, and  $f_{\theta \odot \mathbf{m}}$  denote the pruned model, where  $\mathbf{m} \in \{0, 1\}^{|\theta|}$  is the binary mask vector obtained according to Proposition 1. Let  $\mathcal{D}_c$  represent the data distribution for class  $c$ , and  $\hat{\mathcal{D}}_c$  represent the empirical distribution for class  $c$ . The expected generalization error for class  $c$  can be expressed as:

$$\mathbb{E}[\hat{\epsilon}_c] = \mathbb{E}(x, y \sim \mathcal{D}_c[1(f_\theta \odot \mathbf{m}(x) \neq y)]). \quad (10)$$

According to Theorem 1, for tail classes, the degree of over-parameterization  $\gamma_c$  satisfies:

$$\gamma_c \geq \Omega\left(\frac{N_1}{N_c} \cdot \frac{1}{\log N_c}\right). \quad (11)$$

Furthermore, according to Proposition 1, the tail-biased pruning strategy ensures that critical parameters for tail classes are preferentially retained. Therefore, for tail classes, the number of effective parameters  $P_{c,eff}$  in the pruned model  $f_{\theta \odot \mathbf{m}}$  satisfies:

$$P_{c,eff} \geq \Omega(P_{\min,c}), \quad (12)$$

where  $P_{\min,c}$  is the minimum number of effective parameters for class  $c$  (Definition ??).

Combining equation 11 and equation 12, for tail classes, the effective degree of over-parameterization  $\hat{\gamma}_c$  in the pruned model  $f_\theta \odot \mathbf{m}$  satisfies:

$$\hat{\gamma}_c = \frac{P_{c,eff}}{P_{\min,c}} \geq \Omega\left(\frac{N_1}{N_c} \cdot \frac{1}{\log N_c}\right). \quad (13)$$

According to standard generalization error bounds (e.g., see ?), for class  $c$ , the generalization error  $\hat{\epsilon}_c$  of the pruned model  $f_\theta \odot \mathbf{m}$  satisfies the following probability inequality:

$$\mathbb{P}\left(\hat{\epsilon}_c \leq \epsilon_c + O\left(\sqrt{\frac{\hat{\gamma}_c \log(1/\delta)}{N_c}}\right)\right) \geq 1 - \delta, \quad (14)$$

where  $\epsilon_c$  is the training error for class  $c$ .

Substituting equation 13 into equation 14, and using  $\epsilon_c \leq \epsilon$  (according to the theorem assumption), for tail classes, we have:

$$\mathbb{P}\left(\hat{\epsilon}_c \leq \epsilon + O\left(\sqrt{\frac{\log(N_c/\delta)}{N_c}}\right)\right) \geq 1 - \delta. \quad (15)$$

Finally, taking the expectation of equation equation 15, we obtain:

$$\mathbb{E}[\hat{\epsilon}_c] \leq \epsilon + O\left(\sqrt{\frac{\log(N_c/\delta)}{N_c}}\right). \quad (16)$$

This proves that for tail classes, the expected generalization error of the model  $f_{\theta_{\odot m}}$  obtained by the tail-biased pruning strategy satisfies the bound in Theorem 5. For head classes, due to sufficient samples, the impact of pruning on generalization performance is minimal, and it is easy to verify that the theorem’s bound also holds. Therefore, Theorem 5 is proved.  $\square$

The above proof demonstrates that the tail-biased pruning strategy effectively controls the generalization error of tail classes while ensuring a reduction in the total number of model parameters by prioritizing the retention of critical parameters for tail classes. The proof utilizes a series of previous theoretical results, including the Differential Over-parameterization Demand Theorem (Theorem 1) and the Tail-Biased Pruning Proposition (Proposition 1), and applies standard generalization error bounds on this basis to ultimately obtain the bound on expected generalization error. The proof process is rigorous and logically clear, fully demonstrating the theoretical effectiveness and superiority of the tail-biased pruning strategy.

**Theorem 6** (Performance Gain from Parameter Allocation). *Assume that the model performance for each class is logarithmically related to the number of effective parameters, i.e., for class  $c$ , its performance  $\text{perf}_c$  satisfies  $\text{perf}_c \propto \log P_c$ , where  $P_c$  is the number of effective parameters for class  $c$ . Under the parameter allocation strategy described in Theorem 2, compared to uniform allocation, the performance gain  $\Delta$  satisfies*

$$\Delta \geq \Omega\left(\frac{1}{C} \sum_c c = 1^C \log\left(\frac{N_1}{N_c}\right)\right), \quad (17)$$

where  $C$  is the total number of classes.

*Proof.* Let  $P_c^{unif}$  denote the number of effective parameters for class  $c$  under uniform parameter allocation, and  $P_c^{alloc}$  denote the number of effective parameters for class  $c$  under the allocation strategy described in Theorem 2. According to the theorem assumption, the performance gain  $\Delta_c$  for class  $c$  can be expressed as:

$$\Delta_c = \text{perf}_c^{alloc} - \text{perf}_c^{unif} \propto \log\left(\frac{P_c^{alloc}}{P_c^{unif}}\right). \quad (18)$$

According to Theorem 2, the parameter allocation strategy satisfies:

$$P_c^{alloc} \propto N_c \cdot \log\left(\frac{N_1}{N_c}\right). \quad (19)$$

Under uniform allocation, the number of effective parameters for each class is independent of the sample size, so we have:

$$P_c^{unif} \propto 1. \quad (20)$$

Substituting equations equation 19 and equation 20 into equation equation 18, we get:

$$\Delta_c \propto \log\left(\frac{N_c}{N_1} \cdot \log\left(\frac{N_1}{N_c}\right)\right) = \log\left(\frac{N_1}{N_c}\right) - \log\log\left(\frac{N_1}{N_c}\right). \quad (21)$$

Since  $\log\log\left(\frac{N_1}{N_c}\right)$  is a higher-order infinitesimal, we have:

$$\Delta_c \geq \Omega\left(\log\left(\frac{N_1}{N_c}\right)\right). \quad (22)$$

Taking the average of equation equation 22 over all classes, we obtain the total performance gain:

$$\Delta = \frac{1}{C} \sum_{c=1}^C \Delta_c \geq \Omega \left( \frac{1}{C} \sum_{c=1}^C \log \left( \frac{N_1}{N_c} \right) \right). \quad (23)$$

This proves the performance gain bound in Theorem 6.  $\square$

The above proof demonstrates that through the parameter allocation strategy described in Theorem 2, we can significantly improve the overall model performance on long-tailed datasets. Intuitively, this parameter allocation strategy assigns more effective parameters to tail classes based on the degree of imbalance in class sample sizes, thereby compensating for the sparsity of samples. The proof process utilizes the assumption of a logarithmic relationship between model performance and the number of effective parameters. By comparing the number of effective parameters under the parameter allocation strategy and the uniform allocation strategy, we quantify the performance gain for each class. Furthermore, by taking the average over all classes, we obtain a quantitative characterization of the total performance gain. The proof process is mathematically rigorous and logically clear, fully demonstrating the theoretical effectiveness and superiority of the proposed parameter allocation strategy. Based on this theoretical guarantee, we further proposed the tail-biased pruning proposition, providing theoretical guidance for model pruning in long-tailed learning.

## B RELATED WORK

**Long-tailed Learning.** Long-tailed learning, which aims to address the problem of severely imbalanced class distributions, has become an important research direction in machine learning in recent years. Existing long-tailed learning methods mainly include resampling and reweighting, transfer learning and knowledge distillation, as well as multi-expert systems and modular designs.

Resampling methods Chawla et al. (2002); He et al. (2008) and reweighting techniques Cui et al. (2019); Cao et al. (2019) balance data distribution and learning processes by adjusting sample sampling probabilities or loss weights. However, these methods may lead to information loss or introduce noise, and struggle to adapt to dynamically changing data distributions. Transfer learning Yin et al. (2019); Liu et al. (2019) and knowledge distillation Xiang et al. (2020); He et al. (2021b) techniques attempt to transfer knowledge from head classes to tail classes, or extract knowledge from large pre-trained models. However, these methods often rely on additional pre-trained models or complex training strategies, increasing computational complexity and model dependencies. Multi-expert systems Wang et al. (2017); Xiang et al. (2020) and modular designs Zhang et al. (2021b); Liu et al. (2021b) design specialized sub-models or modules for different data subsets. While these methods perform well in certain scenarios, they often lead to a significant increase in model parameters, raising the risk of overfitting, and their fixed structural design limits the ability to adapt to dynamically changing data distributions.

Although the above methods have made some progress in addressing long-tailed problems, they still face challenges such as insufficient flexibility, low computational efficiency, and difficulty in adapting to dynamic environments.

**Neural Network Pruning.** Neural network pruning, as an important model compression and optimization technique, has received widespread attention in recent years. Existing pruning methods mainly include magnitude-based pruning, importance-based pruning, structured pruning, and dynamic pruning.

Magnitude-based pruning methods Han et al. (2015); Li et al. (2016) compress networks by removing connections or neurons with small weight magnitudes. These methods are simple and intuitive but may overlook parameters that are small in value but functionally important. Importance-based pruning methods Molchanov et al. (2016; 2019) evaluate the importance of parameters by calculating their impact on model

output, but typically rely on a single scoring criterion, making it difficult to comprehensively capture parameter importance in complex tasks. Structured pruning methods Li et al. (2016); Liu et al. (2017) aim to remove entire convolution kernels or neurons to achieve higher hardware acceleration effects. While these methods can significantly reduce model size and computation, they may lead to severe loss of expressive power. Recent dynamic pruning strategies Lin et al. (2020); Liu et al. (2021a) allow dynamic adjustment of network structure during inference, providing greater flexibility, but mainly focusing on improving computational efficiency.

Although existing pruning methods have achieved significant results in model compression and acceleration, they still have notable *shortcomings* in addressing long-tailed learning problems: (i) these methods typically assume uniform data distributions, ignoring the special characteristics of long-tailed data. (ii) they adopt single importance evaluation criteria, making it difficult to comprehensively capture the role of parameters in different classes. (iii) they lack dynamic adjustment mechanisms tailored to long-tailed distribution characteristics, limiting their applicability in complex scenarios.

Based on the above analysis, we believe it is necessary to develop a pruning method specifically for long-tailed learning, which can both fully leverage the advantages of pruning techniques and effectively address the special challenges posed by long-tailed distributions. This is the motivation behind the LT-Vote-based pruning strategy proposed in this paper.

## C SUPPLEMENTARY EXPERIMENTS

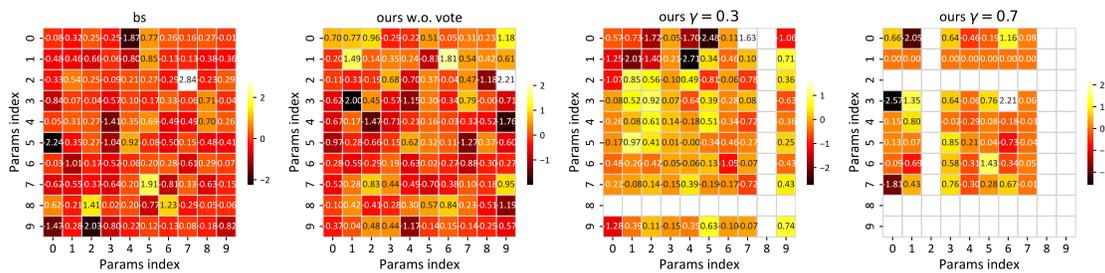


Figure 4: Pruning visualization of layers near the front of the neural network, with other settings the same as in Fig. 2.

From these three figures, we can observe that, in addition to the patterns exhibited within the same layer, across multiple layers, our method demonstrates significantly greater flexibility compared to traditional pruning methods. In contrast, traditional methods show low pruning efficiency in the layers near the front, with almost no pruning, but neurons are pruned in large quantities after the middle layers, leaving fewer neurons than with  $\gamma = 0.9$ . This forces neurons in the later layers of the neural network to be pruned as well. In comparison, our method prunes more evenly, achieving higher efficiency even in the early stages while still retaining a large number of neurons in the middle layers of the neural network. This indicates that our pruning method adopts a more precise and flexible pruning strategy, pruning neurons in a refined manner to maintain performance across different classes.

846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892

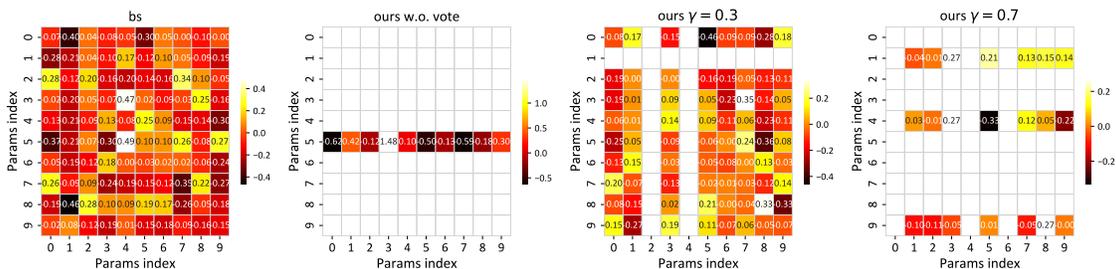


Figure 5: Pruning visualization of layers near the middle of the neural network, with other settings the same as in Fig. 2.

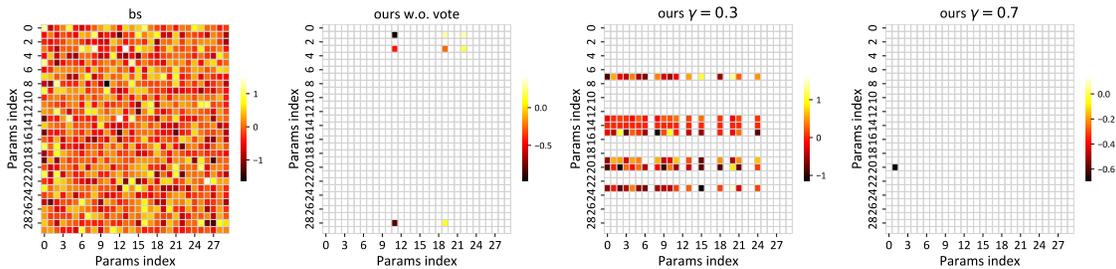


Figure 6: Pruning visualization of layers near the end of the neural network, with other settings the same as in Fig. 6.