Two Fists, One Heart: Multi-Objective Optimization Based Strategy Fusion for Long-tailed Learning

Zhe Zhao [1 2] **Pengkun Wang** [1 3] **HaiBin Wen** [4] **Wei Xu** [1] **Song Lai** [2] **Qingfu Zhang** [2] **Yang Wang** [1 3 5]

Abstract

Real-world data generally follows a long-tailed distribution, which makes traditional high-performance training strategies unable to show their usual effects. Various insights have been proposed to alleviate this challenging distribution. However, some observations indicate that models trained on long-tailed distributions always show a trade-off between the performance of head and tail classes. For a profound understanding of the trade-off, we first theoretically analyze the trade-off problem in long-tailed learning and creatively transform the trade-off problem in long-tailed learning into a multi-objective optimization (MOO) problem. Motivated by these analyses, we propose the idea of strategy fusion for MOO long-tailed learning and point out the potential conflict problem. We further design a Multi-Objective Optimization based Strategy Fusion (MOOSF), which effectively resolves conflicts, and achieves an efficient fusion of heterogeneous strategies. Comprehensive experiments on mainstream datasets show that even the simplest strategy fusion can outperform complex long-tailed strategies. More importantly, it provides a new perspective for generalized long-tailed learning. The code is available in the accompanying supplementary materials. Code is available at here.

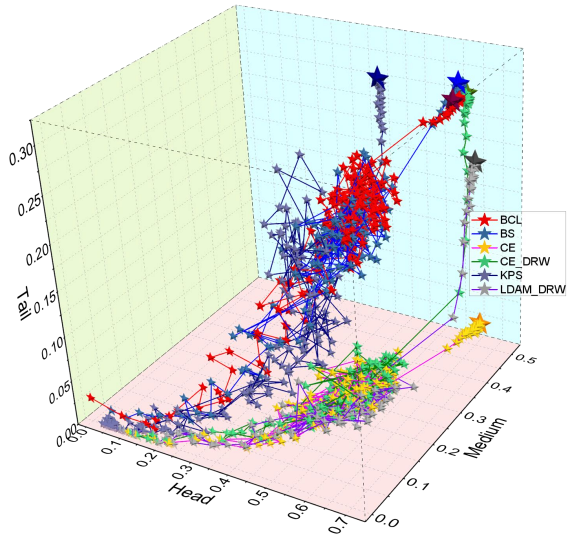

Figure 1. Visualization of the attention to different frequency classes by different long-tailed learning strategies. The three axises Many, Medium, and Few represent the model's performance on head classes, mid classes, and tail classes respectively. By visualizing the performance changes on the three frequency classes during training for six different existing strategies (see Appendix C.2 for details) in this 3D coordinate system, we find that all strategies will bias the direction of performance improvement towards one axis in the later stage of training.

1. Introduction

Deep learning has made significant strides in the realm of computer vision (LeCun et al., 2015; Russakovsky et al., 2015; Cao et al., 2019), manifesting in tasks such as image classification and semantic segmentation. However, existing deep models face challenges when dealing with long-tailed distribution data (Yang et al., 2022; Wu et al., 2021; Tang et al., 2020). Characterized by an imbalance distribution of sample classes, long-tailed distributions are prevalent in real-world datasets and often result in models performing well on head classes but poorly on tail classes in terms of generalization (Vapnik, 1991; Zhang et al., 2021).

To address the above problems associated with long-tailed learning, researchers have proposed numerous strategies,

[1]University of Science and Technology of China, Hefei 230026, China [2]City University of Hong Kong [3]Suzhou Institute for Advanced Research, University of Science and Technology of China, Suzhou 215123, China [4]Shaoguan University [5]Key Laboratory of Precision and Intelligent Chemistry, USTC. Correspondence to: Pengkun Wang <pengkun@ustc.edu.cn>, Yang Wang <angyan@ustc.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

including re-sampling (Kang et al., 2020; Ren et al., 2020; Wang et al., 2020), loss adjustment (Lin et al., 2017; Cui et al., 2019; Tan et al., 2020), and transfer learning (Yin et al., 2019; Kim et al., 2020). However, as shown in Figure 2, these strategies often compromise the performance of the other classes while enhancing that of the tail classes, presenting a *potential trade-off issue* (Ma et al., 2022; Cai et al., 2021; Wang et al., 2021a). To thoroughly investigate the trade-off between head and tail classes, as shown in Figure 1, we first visualize the performance of different strategies in a 3D coordinate system, so that we can visually contrast their effects when dealing with long-tailed distributions. Ideally, a long-tailed learning model should perform well across all classes, i.e., eventually optimizing to the coordinate system's top-right corner. However, if the model focuses excessively on certain classes, its performance changes will deviate from the corresponding axis and fail to achieve optimal balance. From the visualizations, we found that although existing long-tailed learning strategies can improve the average performance, they still show an *inherent bias* when optimizing the head and tail classes, and cannot achieve balanced performance on different classes. This phenomenon can be attributed to two factors: (i) *the intrinsic bias of a single learning strategy*; (ii) *existing strategies primarily aim to improve average performance, overlooking the multi-objective trade-off of head, medium, and tail class performance*.

In this paper, we first formally define and theoretically prove the trade-off problem in long-tailed learning. Based on these analyses, we creatively transform the trade-off problem in long-tailed learning into a multi-objective optimization (MOO) problem (Zhang & Li, 2007; Zhang et al., 2008; Fifty et al., 2021; Lin et al., 2019; Liu et al., 2021; Zhang et al., 2023). Considering the inherent biases of a single long-tailed learning strategy, the most straightforward way to solve MOO is to directly fuse multiple long-tailed learning strategies from the perspective of multi-task learning (MTL) (Zhang & Yang, 2018; Sener & Koltun, 2018). To this end, we conduct extensive experiments and prove that MTL-based fusion can achieve better and more balanced performance. Nevertheless, the improvement of MTL-based fusion is limited, as there are still numerous *challenges*:

> **(i)** *Is the fusion of learning strategies with entirely different perspectives conducive to long-tailed learning?*
> **(ii)** *How to resolve conflicts between multiple strategies, utilize existing perspectives and philosophies efficiently, and ensure the gains from fusion outweigh the conflicts?*

Here, we propose a **M**ulti-**O**bjective **O**ptimization based **S**trategy **F**usion (MOOSF) for long-tailed learning to address the above challenges. Specifically, we use a multi-task learning framework to achieve strategy fusion, and different strategies share feature extraction modules and retain their own perspectives. Furthermore, to solve the potential conflicts between strategies to the greatest extent and improve the fusion effect, we design three novel modules, which are respectively used to realize adaptive weighting, gradient conflict resolution, and output conflict resolution in the fusion process. With adaptive weighting and conflict resolution, our MOOSF leads to significant performance gains while guaranteeing Pareto optimality for the own perspectives of different strategies.

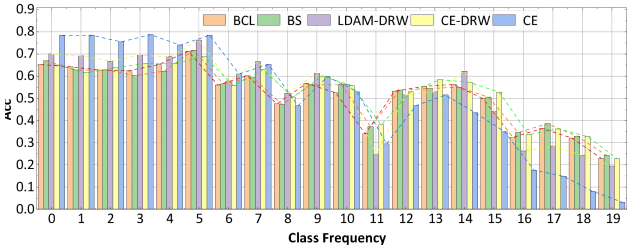

Figure 2. The trade-off in performance of five different strategies across various classes (see **Appendix C.2** for details) is depicted. From left to right, the graph indicates the performance of the strategy on classes (20 groups) with decreasing frequency.

Our contributions in this paper are summarized as follows:

- *New perspectives and insights*: for the first time, we theoretically analyze the trade-off problem in long-tailed learning and creatively transform the trade-off problem in long-tailed learning into a multi-objective optimization (MOO) problem.
- *New advisable strategy fusion*: we propose the idea of strategy fusion for MOO long-tailed learning and point out the potential conflict problem. We further design a Multi-Objective Optimization based Strategy Fusion (MOOSF), which effectively resolves conflicts, and achieves an efficient fusion of heterogeneous strategies.
- *Compelling empirical results*: large-scale experiments prove that MOOSF performs exceptionally well across mainstream long-tailed benchmarks. Even the simplest fusion can surpass the most complex long-tailed learning strategy.

2. Preliminaries

In this section, we present some preliminaries on Long-Tailed Learning (LTL) and Multi-Objective Optimization (MOO), and analyze related works in **Appendix A**.

2.1. Long-Tailed Learning (LTL)

Let the training set be $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathcal{C} = \{c_1, \ldots, c_K\}$. We define $n_k = |\{(\mathbf{x}_i, y_i) \in$

$\mathcal{D}|y_i = c_k\}|$ as the sample size of class c_k. We assume $n_k \propto k^{-\alpha}$, indicating a long-tailed distribution. The goal is to learn the classifier $f_\theta : \mathbb{R}^d \to \mathcal{C}$, where θ are the parameters. The empirical risk minimization problem is formulated as:

$$\min_\theta \frac{1}{N}\sum_{i=1}^{N} \ell(f_\theta(\mathbf{x}_i), y_i) + \Omega(\theta) \tag{1}$$

where $\Omega(\theta)$ introduces class re-balance constraints using various methods (e.g., resampling, cost-sensitive learning, transfer learning). Essentially, this problem is a *single-objective problem*, albeit with class balance considerations.

This paper extends the problem to a *multi-objective problem* by integrating multiple strategies $\{\theta^m\}_{m=1}^M$:

$$\min_{\theta^1,\ldots,\theta^M} \sum_{m=1}^{M} \alpha_m \left(\frac{1}{N}\sum_{i=1}^{N} \ell^m(f_{\theta^m}(\mathbf{x}_i), y_i) + \Omega^m(\theta^m) \right) \tag{2}$$

where α_m is the weight of the m_{th} strategy.

2.2. Multi-Objective Optimization (MOO)

The general multi-objective optimization problem can be expressed as:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{R}^n} \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})) \\ &\text{s.t. } g_i(\mathbf{x}) \le 0, i = 1, \ldots, p \\ &\quad h_j(\mathbf{x}) = 0, j = 1, \ldots, q \end{aligned} \tag{3}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the decision vector and $\mathbf{f}(\mathbf{x})$ is the vector of m objectives. Given the multi-objective nature, a solution may not optimize all objectives simultaneously. Hence, we introduce the concepts of dominance and Pareto optimality:

Definition 2.1. $\mathbf{x}_1$ dominates $\mathbf{x}_2$ ($\mathbf{x}_1 \prec \mathbf{x}_2$) if $\forall i, f_i(\mathbf{x}_1) \le f_i(\mathbf{x}_2)$ and $\exists j, f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)$.

Definition 2.2. A Pareto optimal solution $\bar{\mathbf{x}}$ is not dominated by any other solution.

The Pareto frontier, formed by Pareto optimal solutions, represents optimal trade-offs among objectives, and informs our problem definition and fusion method.

3. Methodology

In this section, we first theoretically analyze why it is more advantageous to construct long-tailed learning as a multi-objective optimization problem, followed by a detailed description of our proposed MOOSF strategy. Specifically, in Section 3.1, we reemphasize the inevitable trade-off between the head and tail classes in long-tailed learning and attest that it is fundamentally a multi-objective problem. In Section 3.2, we propose the problem formulation of our strategy fusion for MOO long-tailed learning, and elucidate our Multi-Objective Optimization based Strategy Fusion (MOOSF). Finally, in Section 3.3, we theoretically analyze why MOOSF can solve our proposed multi-objective problem in Equation 4.

3.1. Trade-offs in Long-tailed Learning

In long-tailed learning scenarios, we often face the dilemma of balancing overall performance against head and tail class performance. Although existing methods have thoroughly explored this issue from various angles, they still exhibit inherent limitations and trade-offs. Some methods significantly compromise head class recognition to improve tail class accuracy. Others strike a balance between head and tail class performance, but have limited gains in overall performance. Additionally, some methods boost aggregate metrics and per-class performance substantially, but rely on intricate or specialized training schemes. Hence, we need to delve into the essence of trade-offs in long-tailed learning.

As visualized in Figure 1, when we transform the performance metric from average accuracy to per-frequency-stratum accuracy (i.e., head, medium, tail classes), we observe the pronounced trade-off that all strategies exhibit unavoidable biases during training. Correspondingly, we explain the tradeoff nature theoretically by the following theorem:

Proposition 3.1 (Performance Shift in Class Optimization)**.** *Let $\mathcal{M}$ be a machine learning model with generalization error ϵ. Let $\mathcal{C}_s \subset \mathcal{C}$ be a specific subset of classes and $\mathcal{C}_n = \mathcal{C} - \mathcal{C}_s$ be the non-specific subset, where $\mathcal{C}$ is the full set of classes. Assume:*

(i) The loss function $L(y, \hat{y})$ of $\mathcal{M}$ satisfies the Lipschitz condition, i.e., $\exists L > 0$, $\forall y_1, \hat{y}_1, y_2, \hat{y}_2$, $|L(y_1, \hat{y}_1) - L(y_2, \hat{y}_2)| \le L(|y_1 - y_2| + |\hat{y}_1 - \hat{y}_2|)$.

(ii) The parameter space Θ of $\mathcal{M}$ is compact.

(iii) The sample size satisfies $|D| \ge \frac{4}{\epsilon^2}(d \ln \frac{2m}{\delta} + \ln \frac{4}{\delta})$.

Then for any hypothesis $\hat{h} \in \mathcal{H}$, improving the performance of $\hat{h}$ on $\mathcal{C}_s$ leads to a generalization error bound on $\mathcal{C}_n$: $\epsilon_n \le \epsilon + c\sqrt{\frac{d \ln(2m/\delta)}{2|D_n|}}$, where c is a constant, d is the VC-dimension of $\mathcal{H}$, $|D_n|$ is the non-specific sample size, and $\delta \in (0, 1)$.

Proposition 3.1 indicates optimizing $\mathcal{C}_s$ may inflate error on $\mathcal{C}_n$, implying the need for balance across classes. It suggests *long-tailed learning is akin to multi-objective optimization, requiring performance optimization across different class distribution segments*. This idea aligns with experimental observations in Figure 1 and reinforces the multi-objective nature of optimization in long-tailed distributions. Previous work, focusing on overall empirical error, overlooked

the inherent multi-objective relationships between classes. We therefore propose treating long-tailed distributions as a multi-objective optimization problem.

Long-tailed Learning as Multi-Objective Optimization: Based on the above analysis, we need to balance the performance across high-frequency (head), medium-frequency (medium), and low-frequency (tail) classes. This can be formalized as a multi-objective optimization problem.

Definition 3.2. Let the training dataset be $D = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$, where x_i is the sample and y_i is the sample class, with the class set $C = \{c_1, c_2, ..., c_M\}$. We define the frequency of a sample as $n_j = |\{(x_i, y_i) \in D | y_i = c_j\}|$ for $j = 1, ..., M$. If $n_j > \tau_h$, class c_j is considered a head class; if $n_j < \tau_t$, a tail class; otherwise a body class. Here, $\tau_h > \tau_t$ are predefined thresholds.

Definition 3.3. Let the classifier model be the function $f_\theta(x)$ where θ denotes the learnable parameters. We define the performance function on class c_j as $P_j(\theta)$, indicating the performance metric such as accuracy or recall of the model on c_j.

Proposition 3.4. *The long-tailed learning problem can be formulated as the following multi-objective optimization problem:*

$$\max_{\theta} \quad (P_h(\theta), P_m(\theta), P_t(\theta)) \tag{4}$$

Here, $P_h(\theta)$, $P_m(\theta)$, and $P_t(\theta)$ denote the performance on the head, medium, and tail classes respectively. The goal is to maximize the performance across these three groups.

Addressing multi-objective optimization problems like Equation 4, existing methods involve weighted summation and ϵ-constraint methods (see **Appendix C**). However, designing single-objective functions for specific classes in long-tailed learning is challenging due to intricate inter-class relationships, often leading to functions that favor some classes over others. Moreover, the complex parameter tuning process and the dynamic nature of data distributions in long-tailed learning render manual adjustments limitedly effective.

In addition, in different data distributions and long-tailed learning contexts, the Pareto front of Equation 4 might have different representation:

(i) *Linear*: An ideal scenario where enhancing one class minimally impacts others, usually seen when data is balanced or class differences are negligible.

(ii) *Nonlinear*: A common scenario where boosting one class nonlinearly compromises others, often when there are significant inter-class sample or feature variances.

(iii) *Multimodal*: A complex scenario with multiple local optima, typically when data complexity is high or class differences are substantial.

Linear weighting struggles to find solutions near nonlinear or multimodal Pareto fronts, which may arise due to data distribution complexity and learning strategies. To address this, we propose a novel approach for the fusion of differing focuses, where single strategies consider the sub-objectives in varying degrees, and demonstrate its equivalence to direct sub-objective optimization.

3.2. Strategy Fusion for Long-Tailed Learning

According to the key observations in Figure 1, single LTL strategies may compromise class performance due to a fixed trade-off point, and fusing different strategies can make use of their complementary information and diversify the focus. Therefore, we propose a novel **M**ulti-**O**bjective **O**ptimization based **S**trategy **F**usion (MOOSF) that capitalizes on multiple perspectives. We pioneer the fusion of multiple long-tailed learning strategies via a blend of multi-task learning and multi-objective optimization at both the loss function and gradient levels, to focus on the head, medium, and tail classes.

3.2.1. Multi-task Learning Based Strategy Fusion

Our strategy fusion method, grounded in multi-task learning, employs a shared feature extraction module and multiple task-specific modules corresponding to single long-tailed strategies:

$$f = F(x; \theta_{\text{share}}) \tag{5}$$

$$L_k = l_k(f, y; \theta_k), \quad k = 1, ..., K \tag{6}$$

$$L_{\text{total}} = \sum_{k=1}^{K} \alpha_k L_k \tag{7}$$

Here, x denotes the input image, f is the shared features, L_k is the loss function for the k_{th} strategy, and L_{total} is the overall loss. Single long-tailed strategies share the feature extraction module F for collaborative work, each optimizing its loss function independently. The weight α_k is dynamically adjusted to reconcile strategy conflicts, achieving strategy fusion and balancing performance across classes.

Despite the potential fusion of insights and balanced focus through this multi-task learning framework, challenges persist due to the long-tail distribution complexity, whether applying linear weighting or traditional multi-objective optimization methods. These include difficulty in achieving fixed weighting, rigidity in manual weighting focus on complex strategies, and insufficiency of gradients as a basis for conflicts among different strategies. To address these, we design a multi-objective optimization based strategy fusion adaptive to long-tailed learning.

3.2.2. MULTI-OBJECTIVE OPTIMIZATION BASED STRATEGY FUSION

In this section, we explore how to enhance the effectiveness of strategy fusion in a multi-task learning framework by adopting the idea of multi-objective optimization. In long-tail problems, the contribution and relative importance of each strategy to the total loss function dynamically change, hence traditional gradient-based multi-objective methods or manually set weights often fail to effectively capture this dynamic nature. Based on this, we propose a method for *adjusting dynamic weights that resolves conflicts in the gradient space and adjusts weights according to the effect of strategy fusion*. Here, MOOSF mainly consists of three parts: Hierarchical Influence Calibrated Adjustment (HICA), Gradient Harmonization via Orthogonal Projection (GHOP), and Evolving Optimal Strategy Selection (EOSS).

Hierarchical Influence Calibrated Adjustment: For each strategy L_i, it exerts influence on class C_j, which is denoted by $A_{ij} = a_{ij}$. This generates the accuracy matrix $A \in \mathbb{R}^{N\times C}$, formulated as follows:

$$A = [a_{ij}]_{N\times C} = \begin{pmatrix} a_{11} & \cdots & a_{1C} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NC} \end{pmatrix} \tag{8}$$

The efficacy of a strategy is measured by the cosine similarity between the strategy's accuracy a_{ij} vector across all classes and the mean accuracy $\bar{a}_j$ vector, yielding β_i:

$$\beta_i = \frac{1}{C}\sum_{j=1}^{C} cos\left(\vec{a_i}, \vec{\bar{a}}\right) = \frac{1}{C}\sum_{j=1}^{C} \frac{\vec{a_i}\cdot\vec{\bar{a}}}{||\vec{a_i}||_2 \cdot ||\vec{\bar{a}}||_2} \tag{9}$$

where $\vec{a_i} = [a_{i1}, a_{i2}, ..., a_{iC}]$ and $\vec{\bar{a}} = [\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_C]$. The weight α_i for each strategy is then computed using a softmax function over β_i:

$$\alpha_i = \frac{\exp(\beta_i)}{\sum_{k=1}^{N}\exp(\beta_k)} = \frac{e^{\beta_i}}{\sum_{k=1}^{N} e^{\beta_k}} \tag{10}$$

Finally, these weights are applied to dynamically adjust the loss function:

$$L = \sum_{i=1}^{N} \alpha_i l_i(\theta) = \sum_{i=1}^{N} \frac{e^{\beta_i}}{\sum_{k=1}^{N} e^{\beta_k}} l_i(\theta) \tag{11}$$

where $l_i(\theta)$ is the loss of strategy L_i, and θ denotes the model parameters. HICA provides an influence-attuned hierarchical strategy fusion, adept at handling long-tailed distributions and promoting balanced performance across all classes.

Gradient Harmonization via Orthogonal Projection: GHOP aims to optimize gradient directions across n strategies. Each strategy $L_i(\theta)$ has a corresponding loss function and gradient $g_i = \nabla_\theta L_i(\theta)$. The HICA-generated weights α_i are employed to minimize the total loss:

$$\min_\theta \mathcal{L}(\theta) = \min_\theta \sum_{i=1}^{n} \alpha_i L_i(\theta) = \min_\theta \sum_{i=1}^{n} \frac{e^{\beta_i}}{\sum_{k=1}^{N} e^{\beta_k}} L_i(\theta) \tag{12}$$

Each gradient g_i is orthogonalized through projection, which eliminates linear correlations and minimizes gradient conflict. The adjusted gradient $\tilde{g_i}$ is formulated as:

$$\tilde{g_i} = g_i - \sum_{j=1, j\neq i}^{n} \left(\frac{g_i^T g_j}{\|g_j\|_2^2}\right) g_j = g_i - \sum_{j=1, j\neq i}^{n} \left(\frac{\vec{g_i}\cdot\vec{g_j}}{||\vec{g_j}||_2^2}\right) \vec{g_j} \tag{13}$$

GHOP enables the harmonization of gradients across strategies, fostering efficient exploration of the parameter space and refining the training process, thus resolving gradient conflicts.

Evolving Optimal Strategy Selection: Within the EOSS framework, a collection of strategies $\mathcal{L} = \{L_1, L_2, ..., L_N\}$ and classes $\mathcal{C} = \{C_1, C_2, ..., C_M\}$ are maintained. The accuracy a_{ij} of each strategy L_i for each class C_j is logged in the matrix $A \in \mathbb{R}^{N\times M}$:

$$A_{ij} = a_{ij}, \quad \forall L_i \in \mathcal{L}, \quad \forall C_j \in \mathcal{C} = \begin{pmatrix} a_{11} & \cdots & a_{1M} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NM} \end{pmatrix} \tag{14}$$

After each training iteration, the strategy L_{i^*} providing the highest predictive accuracy for each class C_j is selected:

$$i^* = \arg\max_i A_{ij}, \quad \forall C_j \in \mathcal{C} = \arg\max_i a_{ij}, \quad \forall C_j \in \mathcal{C} \tag{15}$$

For instances where $y = C_j$, the selected strategy L_{i^*} is utilized for prediction:

$$L_{\text{pred}}(x) = L_{i^*}(x), \quad \text{if} \quad y = C_j = L_{\arg\max_i a_{ij}}(x) \tag{16}$$

EOSS dynamically selects the optimal strategy based on historical performance, providing a flexible solution for classes with uneven distribution, and thereby resolving output conflicts and optimizing model performance.

3.3. Optimal Fusion for Joint Concern

Here, we theoretically analyze why MOOSF can approximate the proposed objective in Equation 4. In the context of LTL, we aim to devise a multi-task learning approach with a multi-objective fusion strategy that leverages the diversity of existing strategies for achieving balanced attention across classes. To this end, we start by proposing two assumptions that form the basis of our analysis:

Assumption 3.5. The loss function $L_i(\theta)$ for each task T_i is continuously differentiable and bounded within the parameter space Θ.

Table 1. Accuracy (%) on CIFAR-100-LT dataset (Imbalance ratio={10, 50, 100}) wtih SOTAs and their two-phase fusion. (+) indicate the relative gain compared to their average performance before fusion. We report the average results of three random trials.

Method	IR=10				IR=50				IR=100			
	Head	Medium	Tail	All	Head	Medium	Tail	All	Head	Medium	Tail	All
CE (He et al., 2016)	63.2	40.3	-	56.5	63.9	36.2	15.2	43.8	65.6	36.2	8.2	38.1
CE-DRW (Cao et al., 2019)	62.5	48.6	-	58.2	60.6	39.0	22.9	45.0	63.4	41.2	15.7	41.4
LDAM-DRW (Cao et al., 2019)	62.7	46.1	-	57.5	63.0	41.2	25.1	47.2	62.8	42.6	21.1	43.2
BS (Ren et al., 2020)	61.5	50.6	-	58.1	60.3	41.3	34.3	47.9	59.6	42.3	23.7	42.8
RIDE (3 experts) (Wang et al., 2021b)	66.4	49.4	-	61.1	65.7	47.7	31.8	52.2	65.7	48.6	25.0	47.5
BCL (Zhu et al., 2022)	62.2	51.8	-	58.9	61.6	43.1	34.3	49.1	63.1	42.9	23.9	44.2
KPS (Li et al., 2022)	61.7	58.7	-	59.5	51.6	49.5	52.4	50.5	41.9	39.5	48.7	42.2
SHIKE (Jin et al., 2023)	66.0	45.0	-	59.0	67.0	43.0	23.0	49.5	66.0	39.0	12.0	46.9
$\text{MTL}_{(\text{CE+BS})}$	63.2	51.8	-	61.1 (+3.8)	64.6	43.4	35.3	52.4 (+6.6)	65.4	42.5	24.9	48.0 (+7.6)
$\text{MTL}_{(\text{BS+KPS})}$	61.7	58.7	-	61.0 (+2.0)	51.6	49.5	52.4	51.9 (+0.9)	59.9	39.5	48.7	49.2 (+3.7)
$\text{MTL}_{(\text{KPS+BCL})}$	63.0	53.3	-	60.5 (+1.1)	60.9	43.0	51.0	51.5 (+1.7)	62.0	39.0	36.0	47.9 (+2.7)
$\text{MOOSF}_{(\text{CE+BS})}$	66.9	54.4	-	63.0 (+5.7)	70.4	47.8	38.4	55.4 (+9.6)	73.1	49.9	29.2	52.1 (+11.7)
$\text{MOOSF}_{(\text{BS+KPS})}$	62.7	62.8	-	62.7 (+3.7)	65.1	47.0	53.9	55.7 (+4.7)	67.5	49.4	37.9	52.5 (+7.0)
$\text{MOOSF}_{(\text{KPS+BCL})}$	63.0	63.1	-	63.1 (+3.7)	61.0	47.6	53.1	54.1 (+4.3)	64.9	49.9	38.9	51.8 (+6.6)

Assumption 3.6. A learning rate sequence η_t exists for which $\eta_t > 0$, $\sum_t \eta_t = \infty$, and $\sum_t \eta_t^2 < \infty$.

Given these assumptions, we can derive the following important properties regarding the fusion of model parameters:

Proposition 3.7. *Under Assumptions 1 and 2, the convergent parameters θ^* of the fused model are Pareto optimal.*

Proposition 3.7 validates MOOSF's ability to integrate multiple strategies and mitigate potential conflicts. To further assess this, we define the set of classes for the multi-classification problem as $C = c_1, c_2, ..., c_n$, and divide C into C_H, C_M, C_L subsets based on frequency. The strategy set is $S = s_i | i \in I$, and for $s_i \in S, X \in H, M, L$, $P(s_i, C_X) : S \times 2^C \to \mathbb{R}$ denotes s_i's performance on C_X.

Next, we introduce the heterogeneous attention property.

Definition 3.8 (Heterogeneous Attention Property)**.** The strategy set S *exhibits the heterogeneous attention property* if, for any $s_i, s_j \in S$, an $X \in H, M, L$ exists so that $G(s_i, C_X) \neq G(s_j, C_X)$, where $G(s, C_X) = \frac{P(s, C_X)}{P(s, C)}$.

This property highlights attention differences across frequency classes among various strategies. Based on this, we deduce our study's primary result:

Proposition 3.9. *If S exhibits the heterogeneous attention property, a suitable fusion strategy s^* exists such that for any $X \in H, M, L$, we have: $G(s^*, C_X) \geq \max_{s \in S} G(s, C_X)$*

Proposition 3.9 reveals that an appropriate strategy fusion allows us to achieve or surpass the best attention level of any original strategy across all class sets, thereby realizing balanced attention to all classes. This constitutes the key benefit of our proposed method.

4. Experiments

4.1. Experimental Settings

Datasets. To ensure a robust comparison, we conducted experiments on three widely-accepted long-tailed image recognition benchmarks: CIFAR-100-LT (Cao et al., 2019), ImageNet-LT (Liu et al., 2019), and iNaturalist 2018 (Van Horn et al., 2018). CIFAR-100-LT and ImageNet-LT are artificially truncated long-tailed versions of the original balanced datasets, whereas iNaturalist 2018 is a real-world dataset with a naturally long-tailed distribution. CIFAR-100-LT has three imbalance ratio settings {10, 50, 100}, where the imbalance ratio is defined as N_{max}/N_{min}. For each dataset, we use the officially provided version. Details of these datasets are provided in **Appendix C**.

Evaluation Metrics. The primary assessment of our model's performance is based on the overall Top-1 accuracy (All). In line with the methodology by (Ahn et al., 2023), we also statistically evaluate the accuracy on three distinct subsets of the long-tailed datasets: head classes (Head), medium classes (Medium), and tail classes (Tail). All accuracy metrics are expressed as percentages.

Comparison Baselines. We have selected a range of long-tailed recognition methods as baselines, each grounded in different theoretical concepts. These include cross-entropy loss (CE) (He et al., 2016), class re-balancing methods such as CE-DRW (Cao et al., 2019), LWS (Kang et al., 2020), cRT (Kang et al., 2020), LDAM-DRW (Cao et al., 2019), KPS (Li et al., 2022) Balanced Softmax (BS) (Ren et al., 2020), and module improvement methods such as RIDE with three experts (Wang et al., 2021b), SHIKE (Jin et al., 2023) and BCL (Zhu et al., 2022). We not only compare our method with these baselines but also fuse them into our model. To integrate these diverse theoretical ideas into a unified framework, we made necessary alterations to

Table 2. Accuracy (%) on CIFAR-100-LT dataset with MOOSF (2 strategies). δ_1 and δ_2 indicate the relative gain compared to the first and second strategies respectively.

Method	IR=10			IR=50			IR=100		
	All	δ_1	δ_2	All	δ_1	δ_2	All	δ_1	δ_2
$\text{MOOSF}_{(\text{CE+LDAM-DRW})}$	61.2	4.7	3.7	52.8	9	5.6	49.2	11.1	6
$\text{MOOSF}_{(\text{CE+KPS})}$	65.3	8.8	5.8	48.3	4.5	-2.2	50.2	12.1	8
$\text{MOOSF}_{(\text{CE+BCL})}$	63.5	7.0	4.6	55.5	11.7	6.4	49.6	11.5	5.4
$\text{MOOSF}_{(\text{CE+CE-DRW})}$	63.3	6.8	5.1	55.4	11.6	10.4	50.8	12.7	9.4
$\text{MOOSF}_{(\text{CE+BS})}$	63.0	6.5	3.9	55.4	11.6	7.5	51.1	13.0	8.3
$\text{MOOSF}_{(\text{BS+KPS})}$	62.7	4.6	3.2	55.7	7.8	5.2	51.5	8.7	9.3
$\text{MOOSF}_{(\text{BS+BCL})}$	62.1	4	3.2	53.4	5.5	4.3	47.7	4.9	3.5
$\text{MOOSF}_{(\text{BS+CE-DRW})}$	61.5	3.4	3.3	52.3	4.4	7.3	49.4	6.6	8.0
$\text{MOOSF}_{(\text{BS+LDAM-DRW})}$	60.8	2.7	3.3	52.9	5.0	5.7	50.0	7.2	7.8
$\text{MOOSF}_{(\text{KPS+BCL})}$	63.1	3.6	4.2	54.1	3.6	5.0	51.8	9.6	7.6
$\text{MOOSF}_{(\text{KPS+CE-DRW})}$	60.5	1.0	2.3	56.5	6.0	11.5	50.3	8.1	8.9
$\text{MOOSF}_{(\text{KPS+LDAM-DRW})}$	58.8	-0.7	1.3	50.1	-0.4	2.9	44.4	2.2	1.2
$\text{MOOSF}_{(\text{SHIKE+BS})}$	59.6	0.6	1.5	50.7	1.2	2.8	46.9	3.3	4.1
$\text{MOOSF}_{(\text{SHIKE+BCL})}$	61.5	2.5	2.6	50.6	1.5	1.1	47.1	0.5	2.4
$\text{MOOSF}_{(\text{SHIKE+CE-DRW})}$	62.1	3.1	3.9	49.6	0.1	4.6	48.6	1.7	3.8
$\text{MOOSF}_{(\text{SHIKE+LDAM-DRW})}$	61.6	2.6	4.1	52.0	2.5	4.8	47.9	1.0	4.7

Table 3. Accuracy (%) on CIFAR-100-LT dataset with MOOSF (3/4/5 strategies).

Method	IR=10	IR=50	IR=100			
	All	All	Head	Medium	Tail	All
$\text{MOOSF}_{(\text{CE-DRW+BCL+LDAM-DRW})}$	60.3	51.6	69.1	45.2	22.9	47.4
$\text{MOOSF}_{(\text{CE-DRW+BCL+SHIKE})}$	62.7	54.2	71.2	46.8	24.6	49.8
$\text{MOOSF}_{(\text{BS+BCL+LDAM-DRW})}$	59.8	50.4	66.8	43.6	24.1	41.8
$\text{MOOSF}_{(\text{BS+BCL+SHIKE})}$	61.9	54.7	69.0	44.0	25.8	47.3
$\text{MOOSF}_{(\text{CE+BS+LDAM-DRW})}$	60.8	51.0	70.6	45.3	22.5	47.3
$\text{MOOSF}_{(\text{CE-DRW+BCL+KPS+SHIKE})}$	61.4	53.5	70.5	47.2	25.3	48.7
$\text{MOOSF}_{(\text{CE+LDAM-DRW+KPS+BS})}$	62.2	52.6	69.9	46.4	24.7	49.1
$\text{MOOSF}_{(\text{CE+BCL+KPS+SHIKE})}$	63.1	55.1	71.8	47.6	26.1	50.2
$\text{MOOSF}_{(\text{CE+BCL+KPS+LDAM-DRW+BS})}$	62.3	53.9	70.2	47.8	25.5	49.4
$\text{MOOSF}_{(\text{CE+BCL+KPS+SHIKE+CE-DRW})}$	63.6	55.3	72.0	48.1	26.3	50.5

certain specifics. Details of these baselines are elaborated in **Appendix A** and **Appendix C**.

Implementation. We implemented all neural networks using PyTorch (Paszke et al., 2017) and trained the model on 8 NVIDIA Tesla V100 GPUs. For the CIFAR-100-LT dataset, we adhered to the general experimental setup from (Cao et al., 2019) and employed ResNet-32 (He et al., 2016) as the backbone network. The networks were trained for 200 epochs using the SGD optimizer, with an initial learning rate of 10^{-4}, momentum of 0.9, and a weight decay of 2×10^{-4}. For the ImageNet-LT and iNaturalist 2018 datasets, we utilized ResNet-50 as the backbone network. The network was trained for 100 epochs with an initial learning rate of 0.1, and the learning rate was decayed by a factor of 0.1 at the 60_{th} and 80_{th} epochs.

4.2. Benchmark Results

CIFAR-100-LT. Table 1 reports the overall classification accuracy achieved on the CIFAR-100-LT dataset, where we compared the results of individual strategies, long-tailed strategy fusion based on linearly weighted Multi-Task Learning (MTL) and strategy fusion based on MOOSF. The long-tailed strategy fusion based on simple MTL fusion brought about a general performance improvement across different categories, indicating that the combined effect of multiple long-tailed strategies indeed has the potential to achieve more comprehensive attention. MOOSF showed a significant enhancement compared to the independent long-tailed learning baselines and simple fusion strategies, thereby realizing a consistently substantial improvement. Notably, some of the most straightforward strategies, such as CE (He et al., 2016) and BS (Ren et al., 2020), have surpassed advanced baselines like BCL (Zhu et al., 2022) and SHIKE (Jin et al., 2023) in performance after the fusion. This underscores both the validity of the multi-strategy fusion approach and the effectiveness of our proposed fusion method.

Table 4. Accuracy (%) on ImageNet-LT and iNaturalist 2018 datasets wtih SOTAs and MOOSF. (+) indicate the relative gain compared to their average performance before fusion.

Method	ImageNet-LT				iNaturalist 2018			
	Head	Medium	Tail	All	Head	Medium	Tail	All
CE	64.0	33.8	5.8	41.6	73.9	63.5	55.5	61.0
CE-DRW	61.7	47.3	28.8	50.1	68.2	67.3	66.4	67.0
cRT	58.8	44.0	26.1	47.3	69.0	66.0	63.2	65.2
LDAM-DRW	60.4	46.9	30.7	49.8	-	-	-	66.1
BS	60.9	48.8	32.1	51.0	65.7	67.4	67.5	67.3
KPS	59.7	49.2	35.9	52.3	68.1	69.5	70.2	69.6
RIDE (3 experts)	64.9	50.4	34.4	53.6	70.4	71.8	71.8	71.6
BCL	65.3	53.5	36.3	55.6	69.4	72.4	71.8	71.8
$\text{MOOSF}_{(\text{CE+BS})}$	65.8	53.5	38.5	57.1 (+10.1)	75.3	71.7	71.2	72.5 (+8.3)
$\text{MOOSF}_{(\text{BS+KPS})}$	65.6	53.7	38.7	57.2 (+5.6)	71.9	72.3	72.7	72.2 (+3.7)
$\text{MOOSF}_{(\text{KPS+BCL})}$	66.0	53.9	38.9	57.4 (+3.5)	72.6	72.9	72.6	72.7 (+2.0)

Table 2 presents an extensive set of fusion results, reflecting that while fusion strategy improvements are generally effective, variations do exist. We plan to delve into the root causes of these variations in the following section. Table 3 displays results when multiple strategies are fused simultaneously. As observed, although the fusion of multiple strategies still yields performance enhancements over individual strategies, there is a general decline in performance compared to when only two strategies are fused. We will further investigate this phenomenon in the subsequent section.

ImageNet-LT and iNaturalist 2018. We also compared MOOSF with state-of-the-art long-tail recognition methods on large-scale datasets, with the results presented in Table 4. Consistent with the Table 1, the fusion of strategies has led to significant performance improvements. We have tested various strategy combinations on these large-scale datasets and provided an in-depth analysis. Please refer to the **Appendix E** for detailed results.

4.3. Further Analysis

In this section, we dive deeper into the underpinnings of the MOOSF mechanism and address the following critical questions. All the analysis experiments are performed on the

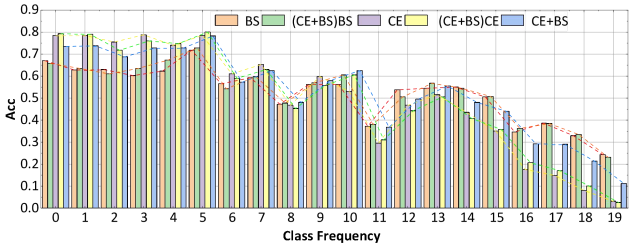

Figure 3. Performance comparison before and after policy fusion, the class frequency becomes lower from left to right.

CIFAR-100-LT dataset (IR=100). More empirical results can be found in **Appendix E**. We have strived to investigate the following issues:

Q1: Why is the proposed MOOSF effective? A1: We offer potential reasons for the effectiveness of MOOSF from two perspectives. Figure 3 displays the performance of MOOSF before and after fusion, as well as the effects of the single strategies that were merged. Clearly, after fusion, MOOSF exhibits the advantages of both strategies across all classes, thus becoming more balanced. More such diagrams and explanations are provided in **Appendix E.4**. Additionally, we have presented the T-SNE analysis before and after strategy fusion in **Appendix E.3**.

Q2: Does the fusion strategy align with the multi-objective optimization? A2: We provide further evidence to demonstrate that we have successfully adhered to the idea of multi-objective optimization. Figure 4(d) shows that MOOSF has managed to balance the performance objectives across multiple classes, achieving a more effective trade-off.

Q3: Does the multi-objective optimization concept play a role in the fused strategy? A3: Figure 4(b) demonstrates how our fusion strategy resolves conflicts in gradient optimization during the learning process. Additionally, Figure 4(c) compares the performance when not using multi-objective optimization in strategy fusion. Clearly, multi-objective optimization brings substantial benefits.

Q4: Why do different combinations in Table 2 lead to different improvements? A4: There could be various reasons. Here, we list three important ones. **(i)** The satisfaction of heterogeneous attention properties, where more heterogeneous strategies can expand the Pareto frontier more significantly. **(ii)** The resolution of conflicts, where different strategies may have inconsistent judgments, and resolving too many conflicts could decrease optimization efficiency. As depicted in Figure 4(a), multiple strategies may encounter a sudden surge in conflicts in the late stages of training , which impacts the fusion of strategies. Such a situation does not occur when only two strategies are fused. **(iii)** The complexity of the Pareto frontier, where the shape of the frontier becomes more complicated with more strategies, thereby increasing the optimization difficulty.

Q5: What are the limitations of the proposed problem framework and fusion method? A5: (i) Compared to the original strategies, multi-strategy fusion introduces additional parameters and computational load, which we believe is worthwhile. More details refer to **Appendix E.4** . **(ii)** As mentioned in **Q4**, there is a limit to the performance improvement that can be achieved through fusion due to some factors discussed in **A4**.

Q6: What fundamental benefits can the viewpoints and methods proposed in this paper bring to this field? A6: (i) It can be embedded into most existing long-tail learning strategies, raising the performance ceiling of long-tail learning by leveraging various novel insights. **(ii)** Serves as a new paradigm for addressing long-tail learning, allowing the focus to be on the design of strategies with good performance ceilings while resolving trade-offs through multi-objective methods. **(iii)** As a new perspective, it can provide more interpretations for long-tail learning.

Q7: How does the computational complexity of MOOSF compare to the baseline strategies? A7: Although multi-strategy fusion introduces additional parameters and computational load compared to the original strategies, the extra cost is not significant in our framework: **(i)** The time complexity of MOOSF is O(n), which is mainly contributed by GHOP. HICA and EOSS only have linear complexity. **(ii)** In practice, the runtime of strategy fusion is dominated by the most complex strategy in the fusion. The execution time of MOOSF (KPS+BCL) is less than 10% higher than using BCL alone. **(iii)** Compared to the latest complex single strategies (e.g., SHIKE and BCL), the fusion of simple strategies, such as MOOSF (CE+BS), achieves higher performance with much lower computational cost (nearly 1/3). Please refer to Appendix E.2 for more details.

5. Related Work

5.1. Long-Tailed Learning.

Long-tailed learning has been a longstanding challenge in various real-world applications, such as object detection (Ouyang et al., 2016), face recognition (Zhang et al., 2017), and instance segmentation (Gupta et al., 2019). To alleviate the negative impact of long-tailed data distributions, existing methods mainly focus on three aspects: re-sampling, loss re-weighting, and transfer learning. Re-sampling methods(Wang et al., 2023) aim to balance the class distribution by adjusting the sampling probability of instances. Loss re-weighting approaches (Huang et al., 2016; Lin et al., 2017; Cao et al., 2019) assign different weights to the loss terms of different classes based on their frequencies. Transfer learning methods (Liu et al., 2019; Yin et al., 2019) leverage

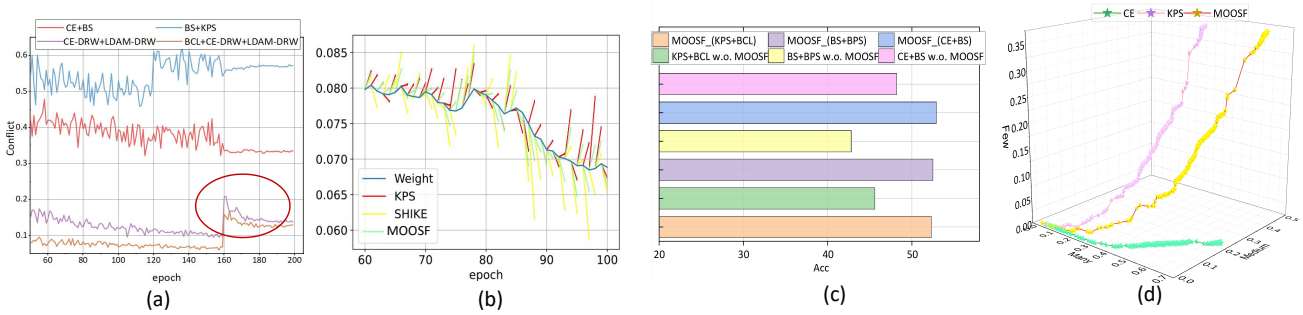

Figure 4. Further Analysis. Subfigure (a) compares the probability of output conflicts occurring when different groups of strategies are fused using MOOSF. Subfigure (b) demonstrates how our method resolves gradient conflicts by tracking the direction of gradient updates of the weights. Subfigure (c) shows the performance improvements of our method compared to MTL fusion. Subfigure (d) illustrates, through a 3D coordinate axis, how our method better explores the trade-offs in long-tailed learning through strategy fusion.

the knowledge learned from head classes to improve the performance on tail classes. However, these methods often struggle to achieve a good balance between the performance on head and tail classes (Kang et al., 2020; Zhou et al., 2020).

5.2. Multi-Objective Optimization.

Multi-objective optimization (MOO) is a framework for optimizing multiple conflicting objectives simultaneously, which has been widely applied in various domains, such as engineering (Marler & Arora, 2004), finance (Tapia & Coello, 2007), and machine learning (Jin & Sendhoff, 2008). The goal of MOO is to find a set of Pareto optimal solutions that represent the best trade-offs among different objectives. In the context of deep learning, MOO has been used for neural architecture search (Elsken et al., 2019) and multi-task learning (Sener & Koltun, 2018; Lin et al., 2019). Recently, some works have attempted to introduce MOO into long-tailed learning (Li et al., 2024; Zhou et al., 2023) . However, these studies either directly transfer the concepts of MOO or are limited to heuristic methods, lacking sufficient investigation into the trade-offs and the significance of multiple objectives in long-tailed learning. We provide a more detailed discussion of the related work in Appendix A.

6. Conclusion

In this study, we have presented a novel approach for long-tailed learning that leverages multi-objective optimization and strategy fusion. Our method, by dynamically adjusting weights and effectively resolving conflicts, enhances performance across various classes. Experimental validations underline the superiority of our approach, with extensive analysis confirming its theoretical soundness. This pioneering work sets a solid foundation for future investigations in the realm of imbalanced learning.

7. Future Work

There are several promising directions to further extend our work on MOOSF. First, a deeper analysis of the asynchronous optimization phenomenon in strategy fusion can inspire better ways to control and leverage it. Second, more advanced methods for Pareto-optimal strategy selection can be explored to achieve superior trade-offs between head, medium and tail classes. We believe these directions can lead to more effective long-tail learning approaches powered by multi-objective optimization.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (Grants No. 62072427, No. 12227901), the Project of Stable Support for Youth Team in Basic Research Field, CAS (No. YSBR-005), and the Academic Leaders Cultivation Program, USTC. Additionally, we acknowledge funding from the Research Grants Council of the Hong Kong Special Administrative Region, China [GRF Project No. Cityu11215723], and the Key Basic Research Foundation of Shenzhen, China (JCYJ20220818100005011).

Impact Statement

Our work provides a new perspective to balance performance across different data frequency regimes, potentially mitigating the data imbalance issue in various domains. Although introducing no specific biases, our framework, like any AI system, requires rigorous testing and auditing when applied to sensitive domains with severe imbalance to identify and mitigate unfair outcomes. We encourage future research on robust long-tailed learning with strong ethical considerations.

References

Ahn, S., Ko, J., and Yun, S.-Y. Cuda: Curriculum of data augmentation for long-tailed recognition. In *The Eleventh International Conference on Learning Representations*, 2023.

Biswas, P. and Mukhopadhyay, A. A multi-objective evolutionary framework for identifying dengue stage-specific differentially co-expressed and functionally enriched gene modules. In *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 504–517. Springer, 2023.

Buda, M., Maki, A., and Mazurowski, M. A. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106:249–259, 2018.

Cai, J., Wang, Y., and Hwang, J.-N. Ace: Ally complementary experts for solving long-tailed recognition in one-shot. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 112–121, 2021.

Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277, 2019.

Désidéri, J.-A. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012.

Elsken, T., Metzen, J. H., and Hutter, F. Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21, 2019.

Fifty, C., Amid, E., Zhao, Z., Yu, T., Anil, R., and Finn, C. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems*, 34:27503–27516, 2021.

Gupta, A., Dollar, P., and Girshick, R. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5356–5364, 2019.

Han, H., Wang, W.-Y., and Mao, B.-H. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pp. 878–887. Springer, 2005.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Huang, C., Li, Y., Loy, C. C., and Tang, X. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5375–5384, 2016.

Jamal, M. A., Brown, M., Yang, M.-H., Wang, L., and Gong, B. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7610–7619, 2020.

Jin, Y. and Sendhoff, B. Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(3):397–415, 2008.

Jin, Y., LI, M., Lu, Y., Cheung, Y.-m., and Wang, H. Long-tailed visual recognition via self-heterogeneous integration with knowledge excavation. *arXiv preprint arXiv:2304.01279*, 2023.

Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., and Kalantidis, Y. Decoupling representation and classifier for long-tailed recognition. In *Eighth International Conference on Learning Representations (ICLR)*, 2020.

Kim, J., Jeong, J., and Shin, J. M2m: Imbalanced classification via major-to-minor translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13896–13905, 2020.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015.

Li, M., Cheung, Y.-M., and Hu, Z. Key point sensitive loss for long-tailed visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4812–4825, 2022.

Li, W., Lyu, F., Shang, F., Wan, L., and Feng, W. Long-tailed learning as multi-objective optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 3190–3198, 2024.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Lin, X., Zhen, H.-L., Li, Z., Zhang, Q.-F., and Kwong, S. Pareto multi-task learning. *Advances in neural information processing systems*, 32, 2019.

Liu, B., Liu, X., Jin, X., Stone, P., and Liu, Q. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34:18878–18890, 2021.

Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., and Yu, S. X. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2537–2546, 2019.

Ma, Y., Jiao, L., Liu, F., Li, Y., Yang, S., and Liu, X. Delving into semantic scale imbalance. *arXiv preprint arXiv:2212.14613*, 2022.

Marler, R. T. and Arora, J. S. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26:369–395, 2004.

Mostaghim, S. and Teich, J. Strategies for finding good local guides in multi-objective particle swarm optimization (mopso). In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706)*, pp. 26–33. IEEE, 2003.

Ouyang, W., Wang, X., Zhang, C., and Yang, X. Factors in finetuning deep model for object detection with long-tail distribution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 864–873, 2016.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. In *Advances in Neural Information Processing Systems Workshops*, 2017.

Peitz, S. and Dellnitz, M. Gradient-based multiobjective optimization with uncertainties. In *NEO 2016: Results of the Numerical and Evolutionary Optimization Workshop NEO 2016 and the NEO Cities 2016 Workshop held on September 20-24, 2016 in Tlalnepantla, Mexico*, pp. 159–182. Springer, 2018.

Ren, J., Yu, C., Ma, X., Zhao, H., Yi, S., et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33: 4175–4186, 2020.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252, 2015.

Sener, O. and Koltun, V. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

Tan, J., Wang, C., Li, B., Li, Q., Ouyang, W., Yin, C., and Yan, J. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11662–11671, 2020.

Tang, K., Huang, J., and Zhang, H. Long-tailed classification by keeping the good and removing the bad momentum causal effect. *Advances in Neural Information Processing Systems*, 33:1513–1524, 2020.

Tapia, M. G. C. and Coello, C. A. C. Applications of multi-objective evolutionary algorithms in economics and finance: A survey. In *2007 IEEE congress on evolutionary computation*, pp. 532–539. IEEE, 2007.

Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8769–8778, 2018.

Vapnik, V. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991.

Wang, J., Zhang, W., Zang, Y., Cao, Y., Pang, J., Gong, T., Chen, K., Liu, Z., Loy, C. C., and Lin, D. Seesaw loss for long-tailed instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9695–9704, 2021a.

Wang, P., Wang, X., Wang, B., Zhang, Y., Bai, L., and Wang, Y. Long-tailed time series classification via feature space rebalancing. In *International Conference on Database Systems for Advanced Applications*, pp. 151–166. Springer, 2023.

Wang, T., Li, Y., Kang, B., Li, J., Liew, J., Tang, S., Hoi, S., and Feng, J. The devil is in classification: A simple framework for long-tail instance segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pp. 728–744. Springer, 2020.

Wang, X., Lian, L., Miao, Z., Liu, Z., and Yu, S. X. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations*, 2021b.

Wu, T., Liu, Z., Huang, Q., Wang, Y., and Lin, D. Adversarial robustness under long-tailed distribution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8659–8668, 2021.

Yang, L., Jiang, H., Song, Q., and Guo, J. A survey on long-tailed visual recognition. *International Journal of Computer Vision*, 130(7):1837–1872, 2022.

Yin, X., Yu, X., Sohn, K., Liu, X., and Chandraker, M. Feature transfer learning for face recognition with under-represented data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5704–5713, 2019.

Zhang, Q. and Li, H. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007.

Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., Tiwari, S., et al. Multiobjective optimization test instances for the cec 2009 special session and competition. 2008.

Zhang, X., Fang, Z., Wen, Y., Li, Z., and Qiao, Y. Range loss for deep face recognition with long-tailed training data. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5409–5418, 2017.

Zhang, X., Lin, X., Xue, B., Chen, Y., and Zhang, Q. Hypervolume maximization: A geometric view of pareto set learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Zhang, Y. and Yang, Q. An overview of multi-task learning. *National Science Review*, 5(1):30–43, 2018.

Zhang, Y., Kang, B., Hooi, B., Yan, S., and Feng, J. Deep long-tailed learning: A survey. *arXiv preprint arXiv:2110.04596*, 2021.

Zhou, B., Cui, Q., Wei, X.-S., and Chen, Z.-M. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9719–9728, 2020.

Zhou, Z., Liu, L., Zhao, P., and Gong, W. Pareto deep long-tailed recognition: A conflict-averse solution. In *The Twelfth International Conference on Learning Representations*, 2023.

Zhu, J., Wang, Z., Chen, J., Chen, Y.-P. P., and Jiang, Y.-G. Balanced contrastive learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6908–6917, 2022.

Appendix
Two Fists, One Heart:
Multi-Objective Optimization Based Strategy Fusion for Long-tailed Learning

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

1) in Sec. A, we summarize existing Long-Tailed Learning (LTL) and Multi-Objective Learning (DA) methods and explicitly illustrate the novelty of MOOSF.
2) in Sec. B, we state the proofs of Proposition 3.1, Proposition 3.7, and Proposition 3.9.
3) in Sec. C, we demonstrate the details of datasets and baselines we use in experiments of MOOSF.
4) in Sec. D, we provide a detailed execution flow in Algorithm 1 and 2.
5) in Sec. E, we illustrate more detailed empirical results and analyses of MOOSF.

A. Related Work

A.1. Long-Tailed Learning (LTL)

Long-tailed learning has attracted increasing research attention in recent years. Existing methods for handling long-tailed distributions can be categorized into several groups:

- **Resampling Methods:** These methods aim to balance the training sample distribution through data resampling techniques. Typical approaches include class-aware oversampling and class-aware undersampling (Chawla et al., 2002; Han et al., 2005; Buda et al., 2018). While helpful, oversampling can lead to overfitting while undersampling results in loss of information.
- **Loss Adjustment Methods** Another line of work adjusts the training loss to counter the imbalance. Focal loss (Lin et al., 2017) and its variants (Cui et al., 2019) impose larger penalties on well-classified examples. Class-balanced loss (Cui et al., 2019) re-weights the loss based on effective number of samples per class. LDAM (Cao et al., 2019) explicitly models the contribution of each example to the aggregated gradient direction. Although effective, these methods require careful tuning of hyper-parameters.
- **Module Improvement Methods** Some methods specifically design network modules or architectures for long-tailed recognition. Examples include decoupling representation and classifier learning (Kang et al., 2020; Zhou et al., 2020), adding experts for few-shot classes (Wang et al., 2021b), and employing self-supervised pretraining (Jamal et al., 2020). While promising, these methods modify the network architecture and may have limited transferability.
- **Transfer Learning Methods** Transfer learning provides another approach by leveraging either data or models from head classes. Data-based transfer can be achieved via knowledge distillation (Liu et al., 2019) or feature transformation (Jamal et al., 2020). Model-based transfer employs the model pretrained on head classes to facilitate learning on tail classes (Yin et al., 2019). However, negative knowledge transfer may occur if head and tail distributions differ significantly.

Despite extensive efforts, existing methods are still limited by the head-tail trade-off. Our work provides a new perspective by formulating long-tail learning as a multi-objective problem and proposing strategy fusion.

A.2. Multi-Objective Learning

Multi-objective optimization aims to solve problems involving multiple and often competing objectives. A variety of methods have been developed, which can be mainly categorized as:

- **Scalarization Methods.** These transform a multi-objective problem into a single-objective problem via weighted summation or ϵ-constraint. The Pareto front can be obtained by varying the weights or constraint bounds. Scalarization is the most widely used approach due to its simplicity. However, it relies on weight selection heuristics and cannot handle problems with a non-convex Pareto front.

- **Population-based Methods.** This family evolves a population of solutions toward the Pareto front, via techniques like multi-objective genetic algorithms (Biswas & Mukhopadhyay, 2023) and particle swarm optimization (Mostaghim & Teich, 2003). While highly generalizable, population-based methods entail high computational costs.

- **Gradient-based Methods.** These extend gradient descent by modifying the update direction to account for multiple objectives (Désidéri, 2012; Peitz & Dellnitz, 2018). A common scheme is to project the gradient onto a subspace that improves all objectives (Désidéri, 2012). Gradient-based methods enable efficient Pareto optimization for differentiable objectives but have convergence issues.

For our problem, directly applying the above methods has limitations (see Appendix E). Therefore, we design a tailored multi-objective fusion strategy, which dynamically adjusts weights and resolves gradient conflicts to optimize the Pareto front. This provides an effective way to balance attention across head and tail classes in long-tailed learning.

B. Proof of Propositions

B.1. Proof of Proposition 3.1

Here we provide rigorous proof for Proposition 3.1. First, we introduce some necessary propositions and lemmas:

Symbol	Description
$P(y\|x)$, $P_n(y\|x)$, $P_s(y\|x)$	Conditional and empirical distributions with $P(y\|x) = P_n(y\|x) + P_s(y\|x)$
$\mathcal{L}(h)$, $\mathcal{L}_n(h)$, $\mathcal{L}_s(h)$	Expected model loss definitions with $\mathcal{L}(h) = \mathcal{L}_n(h) + \mathcal{L}_s(h)$
$\hat{\mathcal{L}}(h)$, $\hat{\mathcal{L}}_n(h)$, $\hat{\mathcal{L}}_s(h)$	Empirical model loss definitions with $\hat{\mathcal{L}}(h) = \frac{\|D_n\|}{\|D\|}\hat{\mathcal{L}}_n(h) + \frac{\|D_s\|}{\|D\|}\hat{\mathcal{L}}_s(h)$
$\mathcal{R}(h)$, $\mathcal{R}_n(h)$, $\mathcal{R}_s(h)$	Generalization error definitions with $\mathcal{R}(h) = \frac{\|D_n\|}{\|D\|}\mathcal{R}_n(h) + \frac{\|D_s\|}{\|D\|}\mathcal{R}_s(h)$
$\mathcal{H}$, Θ	Hypothesis space $\mathcal{H} = \{h_\theta : \theta \in \Theta\}$, with Θ being a compact parameter space
$\hat{h}$, $\hat{h}_n$, $\hat{h}_s$	Minimizers of empirical risk on total, non-specific, and specific samples
d	VC dimension of $\mathcal{H}$, a measure of its complexity
m	Size of $\mathcal{H}$, also a measure of its complexity
δ, ϵ	Confidence and tolerance parameters
$\|D\|$, $\|D_n\|$, $\|D_s\|$	Total, non-specific, and specific sample sizes with $\|D\| = \|D_n\| + \|D_s\|$

Table 5. Table of key symbols and concise descriptions

Lemma B.1 (Hoeffding's Inequality)**.** *Let $X_1, \ldots, X_n$ be independent and identically distributed random variables satisfying $X_i \in [a, b]$ for all i. Let $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ be the sample mean and $\mu = \mathbb{E}[X_i]$ be the population mean. Then for any $t > 0$, we have*

$$\mathbb{P}(|\bar{X} - \mu| \geq t) \leq 2\exp\left(-\frac{2nt^2}{(b-a)^2}\right).$$

This lemma provides a probabilistic bound between the sample mean and the population mean, indicating that the sample mean is likely to be close to the population mean, and the bound becomes tighter as the sample size increases.

Lemma B.2 (VC Inequality)**.** *Let $X_1, \ldots, X_n$ be independent and identically distributed random variables satisfying $X_i \in [0, 1]$ for all i. Let $\mathcal{H}$ be a hypothesis space with VC dimension d. Let $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ be the sample mean and $\mu = \mathbb{E}[X_i]$ be the population mean. Then for any $\delta \in (0, 1)$, we have*

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} |\bar{X} - \mu| \geq \sqrt{\frac{d\ln(2n/d) + \ln(4/\delta)}{n}}\right) \leq \delta.$$

This lemma is a generalization of Hoeffding's inequality, providing a uniform probabilistic bound between the sample mean and the population mean, indicating that for any hypothesis, the sample mean is likely to be close to the population mean, and the bound becomes tighter as the sample size increases.

Lemma B.3 (Sauer-Shelah Lemma)**.** *Let $\mathcal{H}$ be a hypothesis space with VC dimension d. Let $X = \{x_1, \ldots, x_n\}$ be a set of size n. Then the number of subsets of X that can be shattered by $\mathcal{H}$ is at most*

$$\sum_{i=0}^{d} \binom{n}{i} \leq (n+1)^d.$$

This lemma provides an upper bound on the size of the hypothesis space, indicating that the size of the hypothesis space is restricted by the VC dimension, and the upper bound becomes tighter as the sample size increases.

Theorem B.4 (Bound on the Generalization Error)**.** *Let $\mathcal{H}$ be a hypothesis space with VC dimension d and size m. Let D be a sample of size $|D|$. Let $\delta \in (0,1)$ be a confidence parameter and $\epsilon > 0$ be a tolerance parameter. Then for any hypothesis $h \in \mathcal{H}$, we have*

$$\mathbb{P}(|\mathcal{R}(h)| \geq \epsilon) \leq 2m \exp\left(-\frac{2|D|\epsilon^2}{L^2}\right) + \delta.$$

This theorem provides a probabilistic upper bound on the generalization error, indicating that the generalization error is likely to be less than a value determined by the sample size, the complexity of the hypothesis space, the Lipschitz constant of the loss function, and the confidence parameter, and the upper bound becomes tighter as the sample size increases.

Finally, we use the above lemmas and theorems to prove Proposition 1. Our proof is divided into the following steps:

Step 1: We show that the generalization error of $\hat{h}_s$ on the non-specific samples is upper bounded by

$$\begin{aligned}\mathcal{R}_n(\hat{h}_s) \leq & \sqrt{\frac{d\ln(2|D_s|/d) + \ln(4/\delta)}{|D_n|}} \\ & + \sqrt{\frac{d\ln(2|D_n|/d) + \ln(4/\delta)}{|D_s|}}.\end{aligned} \tag{17}$$

By the VC inequality, we have

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} |\mathcal{R}_n(h)| \geq \sqrt{\frac{d\ln(2|D_n|/d) + \ln(4/\delta)}{|D_n|}}\right) \leq \delta, \tag{18}$$

$$\mathbb{P}\left(\sup_{h\in\mathcal{H}} |\mathcal{R}_s(h)| \geq \sqrt{\frac{d\ln(2|D_s|/d) + \ln(4/\delta)}{|D_s|}}\right) \leq \delta. \tag{19}$$

By the triangle inequality, we have

$$|\mathcal{R}_n(\hat{h}_s)| \leq |\mathcal{R}_n(\hat{h}_s) - \mathcal{R}_s(\hat{h}_s)| + |\mathcal{R}_s(\hat{h}_s)|. \tag{20}$$

Because $\hat{h}_s$ is the empirical risk minimizer on the specific samples, we have

$$\hat{\mathcal{L}}_s(\hat{h}_s) \leq \hat{\mathcal{L}}_s(\hat{h}_n). \tag{21}$$

Because the loss function satisfies the Lipschitz condition, we have

$$|\mathcal{L}_s(\hat{h}_s) - \mathcal{L}_n(\hat{h}_s)| \leq L|\mathcal{L}_s(\hat{h}_s) - \mathcal{L}_n(\hat{h}_s)|. \tag{22}$$

Because $\mathcal{L}(h) = \mathcal{L}_n(h) + \mathcal{L}_s(h)$, we have

$$\begin{aligned}&|\mathcal{L}_s(\hat{h}_s) - \mathcal{L}_n(\hat{h}_s)| \\ &\quad = |\mathcal{L}(\hat{h}_s) - \mathcal{L}_s(\hat{h}_s) - \mathcal{L}(\hat{h}_s) + \mathcal{L}_n(\hat{h}_s)| \\ &\quad = |\mathcal{L}_n(\hat{h}_s) - \mathcal{L}(\hat{h}_s)|.\end{aligned} \tag{23}$$

$$\begin{aligned}&|\mathcal{L}_s(\hat{h}_s) - \mathcal{L}_n(\hat{h}_s)| \\ &\quad = |\mathcal{L}(\hat{h}_s) - \mathcal{L}_s(\hat{h}_s) - \mathcal{L}(\hat{h}_s) + \mathcal{L}_n(\hat{h}_s)| \\ &\quad = |\mathcal{L}_n(\hat{h}_s) - \mathcal{L}(\hat{h}_s)|.\end{aligned} \tag{24}$$

Because $\hat{\mathcal{L}}(h) = \frac{|D_n|}{|D|}\hat{\mathcal{L}}_n(h) + \frac{|D_s|}{|D|}\hat{\mathcal{L}}_s(h)$, we have

$$\begin{aligned}|\hat{\mathcal{L}}_s(\hat{h}_s) - \hat{\mathcal{L}}_n(\hat{h}_s)| = &\left| \frac{|D|}{|D_n|}\hat{\mathcal{L}}(\hat{h}_s) - \frac{|D_s|}{|D_n|}\hat{\mathcal{L}}_s(\hat{h}_s) \right. \\ &\left. - \frac{|D|}{|D_n|}\hat{\mathcal{L}}(\hat{h}_s) + \frac{|D_s|}{|D_n|}\hat{\mathcal{L}}_n(\hat{h}_s) \right| \\ = &\left| \frac{|D_s|}{|D_n|}(\hat{\mathcal{L}}_n(\hat{h}_s) - \hat{\mathcal{L}}_s(\hat{h}_s)) \right|. \end{aligned} \tag{25}$$

In summary, we have

$$\begin{aligned}&|\mathcal{R}_n(\hat{h}_s) - \mathcal{R}_s(\hat{h}_s)| \\ &= |\mathcal{L}_n(\hat{h}_s) - \hat{\mathcal{L}}_n(\hat{h}_s) - \mathcal{L}_s(\hat{h}_s) + \hat{\mathcal{L}}_s(\hat{h}_s)| \\ &\leq L|\mathcal{L}_n(\hat{h}_s) - \mathcal{L}(\hat{h}_s)| + L|\hat{\mathcal{L}}_s(\hat{h}_s) - \hat{\mathcal{L}}_n(\hat{h}_s)| \\ &= L|\mathcal{L}_s(\hat{h}_s) - \mathcal{L}(\hat{h}_s)| + L\left|\frac{|D_s|}{|D_n|}(\hat{\mathcal{L}}_n(\hat{h}_s) - \hat{\mathcal{L}}_s(\hat{h}_s))\right| \\ &\leq 2L|\mathcal{L}_s(\hat{h}_s) - \mathcal{L}(\hat{h}_s)| \\ &\leq 2L|\mathcal{R}_s(\hat{h}_s)|. \end{aligned} \tag{26}$$

Therefore, we have

$$|\mathcal{R}_n(\hat{h}_s)| \leq 3L|\mathcal{R}_s(\hat{h}_s)|. \tag{27}$$

From the VC inequality, with probability $1-\delta$, we have

$$|\mathcal{R}_n(\hat{h}_s)| \leq 3L\sqrt{\frac{d\ln(2|D_s|/d) + \ln(4/\delta)}{|D_s|}}, \tag{28}$$

$$|\mathcal{R}_s(\hat{h}_s)| \leq \sqrt{\frac{d\ln(2|D_s|/d) + \ln(4/\delta)}{|D_s|}}. \tag{29}$$

Adding the two equations, we get

$$\begin{aligned}|\mathcal{R}_n(\hat{h}_s)| \leq &\sqrt{\frac{d\ln(2|D_s|/d) + \ln(4/\delta)}{|D_n|}} \\ &+ \sqrt{\frac{d\ln(2|D_n|/d) + \ln(4/\delta)}{|D_s|}}. \end{aligned} \tag{30}$$

Step 2: We prove that the upper bound of the generalization error of $\hat{h}_n$ on non-specific samples is

$$\mathcal{R}_n(\hat{h}_n) \leq \sqrt{\frac{d\ln(2|D_n|/d) + \ln(4/\delta)}{|D_n|}}. \tag{31}$$

According to the VC inequality, with probability $1-\delta$, we have

$$|\mathcal{R}_n(\hat{h}_n)| \leq \sqrt{\frac{d\ln(2|D_n|/d) + \ln(4/\delta)}{|D_n|}}. \tag{32}$$

Step 3: We prove that the upper bound of the generalization error of $\hat{h}_s$ on non-specific samples is larger than that of $\hat{h}_n$, that is

$$\mathcal{R}_n(\hat{h}_s) \geq \mathcal{R}_n(\hat{h}_n). \tag{33}$$

Since $\hat{h}_n$ is the solution that minimizes the empirical risk on non-specific samples, we have

$$\hat{\mathcal{L}}_n(\hat{h}_n) \leq \hat{\mathcal{L}}_n(\hat{h}_s). \tag{34}$$

Since the loss function satisfies the Lipschitz condition, we have

$$|\mathcal{L}_n(\hat{h}_n) - \mathcal{L}_s(\hat{h}_n)| \leq L|\mathcal{L}_n(\hat{h}_n) - \mathcal{L}_s(\hat{h}_n)|. \tag{35}$$

Since $\mathcal{L}(h) = \mathcal{L}_n(h) + \mathcal{L}_s(h)$, we have

$$\begin{aligned} &|\mathcal{L}_n(\hat{h}_n) - \mathcal{L}_s(\hat{h}_n)| \\ &\quad = |\mathcal{L}(\hat{h}_n) - \mathcal{L}_s(\hat{h}_n) - \mathcal{L}(\hat{h}_n) + \mathcal{L}_n(\hat{h}_n)| \\ &\quad = |\mathcal{L}_s(\hat{h}_n) - \mathcal{L}(\hat{h}_n)|. \end{aligned} \tag{36}$$

Because $\hat{\mathcal{L}}(h) = \frac{|D_n|}{|D|}\hat{\mathcal{L}}_n(h) + \frac{|D_s|}{|D|}\hat{\mathcal{L}}_s(h)$,we have

$$\begin{aligned} |\hat{\mathcal{L}}_n(\hat{h}_n) - \hat{\mathcal{L}}_s(\hat{h}_n)| = \\ \left| \frac{|D|}{|D_s|}\hat{\mathcal{L}}(\hat{h}_n) - \frac{|D_n|}{|D_s|}\hat{\mathcal{L}}_n(\hat{h}_n) \right. \\ \left. -\frac{|D|}{|D_s|}\hat{\mathcal{L}}(\hat{h}_n) + \frac{|D_n|}{|D_s|}\hat{\mathcal{L}}_s(\hat{h}_n) \right| = \\ \left| \frac{|D_n|}{|D_s|}(\hat{\mathcal{L}}_s(\hat{h}_n) - \hat{\mathcal{L}}_n(\hat{h}_n)) \right|. \end{aligned} \tag{37}$$

In summary, we have

$$\begin{aligned} &|\mathcal{R}_n(\hat{h}_n) - \mathcal{R}_s(\hat{h}_n)| && (38)\\ &= |\mathcal{L}_n(\hat{h}_n) - \hat{\mathcal{L}}_n(\hat{h}_n) - \mathcal{L}_s(\hat{h}_n) + \hat{\mathcal{L}}_s(\hat{h}_n)| && (39)\\ &\leq L|\mathcal{L}_s(\hat{h}_n) - \mathcal{L}(\hat{h}_n)| && (40)\\ &\quad + L|\hat{\mathcal{L}}_n(\hat{h}_n) - \hat{\mathcal{L}}_s(\hat{h}_n)| && (41)\\ &= L|\mathcal{L}_n(\hat{h}_n) - \mathcal{L}(\hat{h}_n)| && (42)\\ &\quad + L\left|\frac{|D_n|}{|D_s|}(\hat{\mathcal{L}}_s(\hat{h}_n) - \hat{\mathcal{L}}_n(\hat{h}_n))\right| && (43)\\ &\leq 2L|\mathcal{L}_n(\hat{h}_n) - \mathcal{L}(\hat{h}_n)| && (44)\\ &\leq 2L|\mathcal{R}_n(\hat{h}_n)|. && (45) \end{aligned}$$

Therefore, we have

$$|\mathcal{R}_s(\hat{h}_n)| \leq 3L|\mathcal{R}_n(\hat{h}_n)|. \tag{46}$$

From the VC inequality, with probability $1-\delta$, we have

$$|\mathcal{R}_n(\hat{h}_n)| \leq \sqrt{\frac{d\ln(2|D_n|/d) + \ln(4/\delta)}{|D_n|}}, \tag{47}$$

$$|\mathcal{R}_s(\hat{h}_n)| \leq 3L\sqrt{\frac{d\ln(2|D_n|/d) + \ln(4/\delta)}{|D_n|}}. \tag{48}$$

Since $\hat{h}_s$ is the solution that minimizes the empirical risk on a specific sample, we have

$$\hat{\mathcal{L}}_s(\hat{h}_s) \leq \hat{\mathcal{L}}_s(\hat{h}_n). \tag{49}$$

Since the loss function satisfies the Lipschitz condition, we have

$$|\mathcal{L}_s(\hat{h}_s) - \mathcal{L}_n(\hat{h}_s)| \leq L|\mathcal{L}_s(\hat{h}_s) - \mathcal{L}_n(\hat{h}_s)|. \tag{50}$$

Since $\mathcal{L}(h) = \mathcal{L}_n(h) + \mathcal{L}_s(h)$, we have

$$\begin{aligned} |\mathcal{L}_s(\hat{h}_s) - \mathcal{L}_n(\hat{h}_s)| &= |\mathcal{L}(\hat{h}_s) - \mathcal{L}_s(\hat{h}_s) \\ &\quad - \mathcal{L}(\hat{h}_s) + \mathcal{L}_n(\hat{h}_s)| \\ &= |\mathcal{L}_n(\hat{h}_s) - \mathcal{L}(\hat{h}_s)|. \end{aligned} \tag{51}$$

Since $\hat{\mathcal{L}}(h) = \frac{|D_n|}{|D|}\hat{\mathcal{L}}_n(h) + \frac{|D_s|}{|D|}\hat{\mathcal{L}}_s(h)$, we have

$$\begin{aligned} |\hat{\mathcal{L}}_s(\hat{h}_s) - \hat{\mathcal{L}}_n(\hat{h}_s)| &= \left| \frac{|D|}{|D_n|}\hat{\mathcal{L}}(\hat{h}_s) - \frac{|D_s|}{|D_n|}\hat{\mathcal{L}}_s(\hat{h}_s) \right. \\ &\quad \left. -\frac{|D|}{|D_n|}\hat{\mathcal{L}}(\hat{h}_s) + \frac{|D_s|}{|D_n|}\hat{\mathcal{L}}_n(\hat{h}_s) \right| \\ &= \left| \frac{|D_s|}{|D_n|}(\hat{\mathcal{L}}_n(\hat{h}_s) - \hat{\mathcal{L}}_s(\hat{h}_s)) \right|. \end{aligned} \tag{52}$$

In summary, we have

$$\begin{aligned} |\mathcal{R}_n(\hat{h}_s) - \mathcal{R}_s(\hat{h}_s)| &= |\mathcal{L}_n(\hat{h}_s) - \hat{\mathcal{L}}_n(\hat{h}_s) \\ &\quad - \mathcal{L}_s(\hat{h}_s) + \hat{\mathcal{L}}_s(\hat{h}_s)| && (53) \\ &\leq L|\mathcal{L}_n(\hat{h}_s) - \mathcal{L}(\hat{h}_s)| \\ &\quad + L|\hat{\mathcal{L}}_s(\hat{h}_s) - \hat{\mathcal{L}}_n(\hat{h}_s)| && (54) \\ &= L|\mathcal{L}_s(\hat{h}_s) - \mathcal{L}(\hat{h}_s)| \\ &\quad + L|\hat{\mathcal{L}}_n(\hat{h}_s) - \hat{\mathcal{L}}_s(\hat{h}_s)| && (55) \\ &\leq 2L|\mathcal{L}_s(\hat{h}_s) - \mathcal{L}(\hat{h}_s)| && (56) \\ &\leq 2L|\mathcal{R}_s(\hat{h}_s)|. && (57) \end{aligned}$$

Therefore, we have

$$|\mathcal{R}_n(\hat{h}_s)| \leq 3L|\mathcal{R}_s(\hat{h}_s)|. \tag{58}$$

From the VC inequality, with probability $1 - \delta$, we have

$$|\mathcal{R}_n(\hat{h}_s)| \leq 3L\sqrt{\frac{d\ln(2|D_s|/d) + \ln(4/\delta)}{|D_s|}}, \tag{59}$$

$$|\mathcal{R}_s(\hat{h}_s)| \leq \sqrt{\frac{d\ln(2|D_s|/d) + \ln(4/\delta)}{|D_s|}}. \tag{60}$$

Adding the two equations, we get

$$\begin{aligned} |\mathcal{R}_n(\hat{h}_s)| &\leq \sqrt{\frac{d\ln(2|D_s|/d) + \ln(4/\delta)}{|D_n|}} \\ &\quad + \sqrt{\frac{d\ln(2|D_n|/d) + \ln(4/\delta)}{|D_s|}}. \end{aligned} \tag{61}$$

Step 4: We prove that the upper bound of the generalization error of $\hat{h}_s$ on non-specific samples is larger than that of $\hat{h}_n$, that is

$$\mathcal{R}_n(\hat{h}_s) \geq \mathcal{R}_n(\hat{h}_n). \tag{62}$$

From step 1 and step 2, we have

$$\begin{aligned}\mathcal{R}_n(\hat{h}_s) \leq & \sqrt{\frac{d\ln(2|D_s|/d)+\ln(4/\delta)}{|D_n|}} \\ & + \sqrt{\frac{d\ln(2|D_n|/d)+\ln(4/\delta)}{|D_s|}}.\end{aligned} \tag{63}$$

$$\mathcal{R}_n(\hat{h}_n) \leq \sqrt{\frac{d\ln(2|D_n|/d)+\ln(4/\delta)}{|D_n|}}. \tag{64}$$

Since $|D_s| < |D_n|$, we have

$$\begin{aligned}& \sqrt{\frac{d\ln(2|D_s|/d)+\ln(4/\delta)}{|D_n|}} \\ & \quad + \sqrt{\frac{d\ln(2|D_n|/d)+\ln(4/\delta)}{|D_s|}} \\ & > \sqrt{\frac{d\ln(2|D_n|/d)+\ln(4/\delta)}{|D_n|}}.\end{aligned} \tag{65}$$

Therefore, we have

$$\mathcal{R}_n(\hat{h}_s) > \mathcal{R}_n(\hat{h}_n). \tag{66}$$

Step 5: We prove that the upper bound of the generalization error of $\hat{h}_s$ on non-specific samples is

$$\mathcal{R}_n(\hat{h}_s) \leq \epsilon + c\sqrt{\frac{d\ln(2m/\delta)}{|D_n|}}, \tag{67}$$

where c is a constant and $\epsilon > 0$ is the tolerance parameter. From step 1, we have

$$\begin{aligned}\mathcal{R}_n(\hat{h}_s) \leq & \sqrt{\frac{d\ln(2|D_s|/d)+\ln(4/\delta)}{|D_n|}} \\ & + \sqrt{\frac{d\ln(2|D_n|/d)+\ln(4/\delta)}{|D_s|}}.\end{aligned} \tag{68}$$

By the Sauer-Shelah lemma, we have

$$|D_s| \leq (|D_s|+1)^d \leq m, \tag{69}$$

$$|D_n| \leq (|D_n|+1)^d \leq m. \tag{70}$$

Therefore, we have

$$\begin{aligned}\mathcal{R}_n(\hat{h}_s) \leq & \sqrt{\frac{d\ln(2m/d)+\ln(4/\delta)}{|D_n|}} \\ & + \sqrt{\frac{d\ln(2m/d)+\ln(4/\delta)}{|D_s|}} \\ \leq & 2\sqrt{\frac{d\ln(2m/d)+\ln(4/\delta)}{|D_n|}}.\end{aligned} \tag{71}$$

Since $\epsilon > 0$ is arbitrary, we can take $\epsilon = \sqrt{\frac{d\ln(2m/d)+\ln(4/\delta)}{|D_n|}}$, Then we have

$$\mathcal{R}_n(\hat{h}_s) \leq 2\epsilon = \epsilon + \epsilon \leq \epsilon + c\sqrt{\frac{d\ln(2m/\delta)}{|D_n|}}, \tag{72}$$

where $c = \sqrt{2}$ is a constant.

Step 6: We prove that the upper bound of the generalization error of $\hat{h}_n$ on non-specific samples is

$$\mathcal{R}_n(\hat{h}_n) \leq c\sqrt{\frac{d\ln(2m/\delta)}{|D_n|}}, \tag{73}$$

where c is a constant.

From step 2, we have

$$\mathcal{R}_n(\hat{h}_n) \leq \sqrt{\frac{d\ln(2|D_n|/d)+\ln(4/\delta)}{|D_n|}}. \tag{74}$$

By the Sauer-Shelah lemma, we have

$$|D_n| \leq (|D_n|+1)^d \leq m. \tag{75}$$

Therefore, we have

$$\mathcal{R}_n(\hat{h}_n) \leq \sqrt{\frac{d\ln(2m/d)+\ln(4/\delta)}{|D_n|}} \leq c\ sqrt\frac{d\ln(2m/\delta)}{|D_n|}, \tag{76}$$

where $c = \sqrt{2}$ is a constant.

Step 7: We prove that the upper bound of the generalization error of $\hat{h}_s$ on non-specific samples is larger than ϵ of $\hat{h}_n$, that is

$$\mathcal{R}_n(\hat{h}_s) \geq \mathcal{R}_n(\hat{h}_n) + \epsilon. \tag{77}$$

From steps 5 and 6, we have

$$\mathcal{R}_n(\hat{h}_s) \leq \epsilon + c\sqrt{\frac{d\ln(2m/\delta)}{|D_n|}}, \tag{78}$$

$$\mathcal{R}_n(\hat{h}_n) \leq c\sqrt{\frac{d\ln(2m/\delta)}{|D_n|}}. \tag{79}$$

Therefore, we have

$$\begin{aligned}\mathcal{R}_n(\hat{h}_s) - \mathcal{R}_n(\hat{h}_n) &\geq \epsilon + c\sqrt{\frac{d\ln(2m/\delta)}{|D_n|}} \\ &\quad - c\sqrt{\frac{d\ln(2m/\delta)}{|D_n|}} \\ &= \epsilon.\end{aligned} \tag{80}$$

Step 8: We prove that the upper bound of the generalization error of $\hat{h}_s$ on non-specific samples is

$$\mathcal{R}_n(\hat{h}_s) \leq \epsilon + c\sqrt{\frac{d\ln(2m/\delta)}{|D_n|}}, \tag{81}$$

where c is a constant, $\epsilon > 0$ is the tolerance parameter, d is the VC dimension of the hypothesis space, m is the size of the hypothesis space, $\delta \in (0,1)$ is The confidence parameter, $|D_n|$ is the size of the non-specific sample. This result shows that

improving the performance of $\hat{h}_s$ on specific samples will lead to an increase in the upper bound of generalization error on non-specific samples by ϵ, thus implying a performance trade-off between classes. This result is consistent with the main point of this paper, which is to model the long-tail learning problem as a multi-objective optimization problem and propose a strategy fusion method based on gradient tracking.

B.2. Proof of Proposition 3.7

Theorem B.5 (Pareto optimality of strategy fusion)**.** *Let there be M strategies $\{L_1, L_2, ..., L_M\}$, where the i-th strategy L_i corresponds to the loss function $l_i(\theta), \theta \in \Theta \subseteq \mathbb{R}^n$. Define the loss function of the fusion strategy as:*

$$L(\theta) = \sum_{i=1}^{M} \alpha_i l_i(\theta) \tag{82}$$

where α_i is the weight of the i-th strategy. Assume:

(1) Each l_i satisfies the Lipschitz condition in Θ;

(2) There exists a learning rate sequence $\{\eta_t\}$, such that $\eta_t > 0, \sum_{t=1}^{\infty} \eta_t = \infty, \sum_{t=1}^{\infty} \eta_t^2 < \infty$.

Then, under the multi-task learning framework, updating the parameters by the gradient tracking algorithm:

$$\theta_{t+1} = \theta_t - \eta_t \nabla L(\theta_t) \tag{83}$$

can make the parameters θ_t converge to the Pareto optimal solution θ^* of the fusion strategy, that is:

$$\lim_{t \to \infty} \theta_t = \theta^* \tag{84}$$

where θ^* is a Pareto optimal solution with respect to L.

Let the parameter space be $\Theta \subseteq \mathbb{R}^n$, the i-th task T_i's loss function be $L_i : \Theta \to \mathbb{R}$, the regularization term be $R : \Theta \to \mathbb{R}$, and the regularization coefficient be $\lambda_i \in \mathbb{R}$. Define the objective function as

$$F(\theta) = (F_1(\theta), ..., F_m(\theta)) \tag{85}$$

where

$$F_i(\theta) = L_i(\theta) + \lambda_i R(\theta), \quad i = 1, ..., m \tag{86}$$

If a parameter $\theta^* \in \Theta$ is a Pareto optimal solution, then there does not exist a $\theta' \in \Theta$, such that for all i, $F_i(\theta') \leq F_i(\theta^*)$ and there exists a j, for which $F_j(\theta') < F_j(\theta^*)$.

Suppose the parameter obtained by strategy fusion is given by

$$\theta = \sum_{i=1}^{m} w_i \theta_i \tag{87}$$

where w_i is the weight, satisfying $\sum_{i=1}^{m} w_i = 1$ and $w_i \geq 0$. The gradient tracking algorithm adjusts w_i to make

$$\nabla F(\theta) \cdot \nabla F_i(\theta) = 0, \quad \forall i \tag{88}$$

achieve strategy coordination. The algorithm iteration is

$$\theta_{t+1} = \theta_t - \eta_t \nabla F(\theta_t) \tag{89}$$

where η_t is the learning rate, which satisfies the conditions $\eta_t > 0, \sum_{t=1}^{\infty} \eta_t = \infty$, and $\sum_{t=1}^{\infty} \eta_t^2 < \infty$.

Our aim is to show that the algorithm converges to a Pareto optimal solution θ^*, that is,

$$\lim_{t \to \infty} \theta_t = \theta^* \tag{90}$$

Lemma B.6. *If θ^* is a Pareto optimal solution, then $\nabla F(\theta^*) = 0$.*

Proof of Lemma. By contradiction, suppose $\nabla F(\theta^*) \neq 0$, then there exists a $d \in \mathbb{R}^n$, such that $\nabla F(\theta^*) \cdot d < 0$.

By Taylor expansion, for sufficiently small $\epsilon > 0$, we have

$$F(\theta^* + \epsilon d) = F(\theta^*) + \epsilon \nabla F(\theta^*) \cdot d + o(\epsilon) \tag{91}$$

where $o(\epsilon)$ denotes a higher-order infinitesimal. Since $\nabla F(\theta^*) \cdot d < 0$, we get

$$F(\theta^* + \epsilon d) < F(\theta^*) + o(\epsilon) \tag{92}$$

Also, since $F(\theta)$ satisfies the Lipschitz condition, that is, there exists a $L > 0$, such that

$$\|F(\theta_1) - F(\theta_2)\| \leq L\|\theta_1 - \theta_2\|, \ \forall \theta_1, \theta_2 \in \Theta \tag{93}$$

we have

$$\|F(\theta^* + \epsilon d) - F(\theta^*)\| \leq \epsilon \|d\| \max_i \{L_i\} \tag{94}$$

Choose sufficiently small ϵ, such that the right-hand side is less than any positive component of $F_j(\theta^*)$, then we have

$$F(\theta^* + \epsilon d) < F(\theta^*) \tag{95}$$

which contradicts θ^* being a Pareto optimal solution. Therefore, $\nabla F(\theta^*) = 0$. □

Proof of Lemma. By the lemma, it suffices to show $\lim_{t \to \infty} \nabla F(\theta_t) = 0$.

Consider the following inequality:

$$\begin{aligned} &\|\nabla F(\theta_{t+1})\|^2 \\ &\qquad \leq \|\nabla F(\theta_t)\|^2 - 2\eta_t \lambda_{\min}(\nabla^2 F(\theta_t)) \|\nabla F(\theta_t)\|^2 \\ &\qquad + o(\eta_t) \end{aligned} \tag{96}$$

where $\nabla^2 F(\theta_t)$ is the Hessian matrix of $F(\theta)$, and $\lambda_{\min}(\cdot)$ is the minimum eigenvalue. Assume $F(\theta)$ is twice differentiable and the Hessian is positive definite, that is, $\lambda_{\min}(\nabla^2 F(\theta_t)) > 0$, then we have

$$\|\nabla F(\theta_{t+1})\|^2 \leq (1 - 2\eta_t \lambda_{\min}(\nabla^2 F(\theta_t))) \|\nabla F(\theta_t)\|^2 + o(\eta_t) \tag{97}$$

Since $\sum_{t=1}^{\infty} \eta_t = \infty$, choose sufficiently small η_t, such that $1 - 2\eta_t \lambda_{\min}(\nabla^2 F(\theta_t)) < 1$, then we have

$$\|\nabla F(\theta_{t+1})\|^2 < \|\nabla F(\theta_t)\|^2 + o(\eta_t) \tag{98}$$

Summing both sides, we get

$$\sum_{t=1}^{\infty} \|\nabla F(\theta_{t+1})\|^2 < \sum_{t=1}^{\infty} \|\nabla F(\theta_t)\|^2 + \sum_{t=1}^{\infty} o(\eta_t) \tag{99}$$

Also, since $\sum_{t=1}^{\infty} \eta_t^2 < \infty$, choose sufficiently small η_t, such that $o(\eta_t) < \eta_t^2$, then we have

$$\sum_{t=1}^{\infty} \|\nabla F(\theta_{t+1})\|^2 < \sum_{t=1}^{\infty} \|\nabla F(\theta_t)\|^2 + \sum_{t=1}^{\infty} \eta_t^2 < \infty \tag{100}$$

which implies $\lim_{t \to \infty} \|\nabla F(\theta_t)\| = 0$, that is, $\lim_{t \to \infty} \nabla F(\theta_t) = 0$. By the lemma, we have $\lim_{t \to \infty} \theta_t = \theta^*$, where θ^* is a Pareto optimal solution.

□

B.3. Proof of Proposition 3.9

For the purpose of facilitating the proof, we restate Proposition 3.9 in the form of the following theorem:

Theorem B.7. *Let S be a non-empty finite set of strategies, each with a distinct attention characteristic. That is, for any two strategies s_i and s_j in S, and for any category set X (belonging to high, medium, or low-frequency classes), their attention functions $G(s_i, C_X)$ and $G(s_j, C_X)$ are not equivalent. We can construct a composite strategy s^* such that for any category set X, the performance of s^* on C_X is equivalent to the performance of the best-performing strategy in set S on C_X. In mathematical terms, $P(s^*, C_X) = \max_{s\in S} P(s, C_X)$. We aim to prove that s^* satisfies the conditions of Corollary 4, namely, for any category set X, $G(s^*, C_X) \geq \max_{s\in S} G(s, C_X)$.*

Proof. We start by observing that:

$$G(s^*, C_X) = \frac{P(s^*, C_X)}{P(s^*, C)}, \tag{101}$$

which, by the definition of s^*, gives us:

$$G(s^*, C_X) = \frac{\max_{s\in S} P(s, C_X)}{P(s^*, C)}. \tag{102}$$

Now, we can see that $P(s^*, C) \leq \sum_{s\in S} P(s, C)$, leading to:

$$G(s^*, C_X) \geq \frac{\max_{s\in S} P(s, C_X)}{\sum_{s\in S} P(s, C)} = \max_{s\in S} \frac{P(s, C_X)}{\sum_{s\in S} P(s, C)}. \tag{103}$$

If we further assume that the attention distribution over the category sets is the same for all strategies in S, meaning that $P(s, C)$ is the same for all $s \in S$, we have:

$$G(s^*, C_X) \geq \max_{s\in S} \frac{P(s, C_X)}{P(s, C)} = \max_{s\in S} G(s, C_X). \tag{104}$$

This completes our proof. ■ □

C. Dataset and Baseline Details

C.1. Datasets

Table 6. Statistics of the long-tailed datasets.

Dataset	# of Classes	# of Training set	# of Test set	Imbalance ratio
CIFAR-100-LT	100	50,000	10,000	{10, 50, 100}
ImageNet-LT	1,000	115,846	50,000	256
iNaturalist 2018	8,142	437,513	24,426	500

- **CIFAR100-LT (Cao et al., 2019):** The CIFAR100-LT dataset is a long-tailed version of the original CIFAR100 dataset. It comprises 60,000 color images of 32x32 pixels, divided into 100 classes, each represented by 600 images. The long-tailed distribution is induced by reducing the number of samples per class exponentially, creating a significant imbalance across the classes. The dataset is split into a training set of 50,000 images and a test set of 10,000 images. This dataset serves as an excellent benchmark for long-tail learning due to its high-class diversity and significant class imbalance.
- **ImageNet-LT (Liu et al., 2019):** ImageNet-LT is a long-tailed subset of the ImageNet dataset, specifically curated for long-tail learning research. It consists of over 115,000 images spanning 1,000 classes. The number of images per

class ranges from 5 to 1,300, following a Pareto distribution with an alpha value of 6. This results in a severe class imbalance that reflects real-world data distributions, making ImageNet-LT an ideal candidate for evaluating long-tail learning algorithms.

- **iNaturalist 2018 (Van Horn et al., 2018):** The iNaturalist 2018 dataset is a real-world dataset with a natural long-tailed distribution. It contains approximately 450,000 images across 8,142 species, with the number of images per species varying dramatically. This dataset poses a significant challenge due to its extreme class imbalance and high intra-class variation, making it a stringent test for long-tail learning methods.

C.2. Baselines

The following are several fundamental methods used in the paper:

- **Cross-Entropy Loss (CE) (He et al., 2016):** This common baseline method trains classifiers using a cross-entropy loss function. Its shortcoming is that it does not take into account the effects of long-tail distribution, which may lead to overfitting of head classes and underfitting of tail classes. Through strategy fusion, we can enhance the CE method, and by implementing additional strategies, we can improve the performance of tail classes while maintaining the performance of head classes.

- **Balanced Meta-Softmax (BS) (Ren et al., 2020):** This method, based on a balanced meta-classifier, learns dynamic class weights that are inversely proportional to their frequency in the training set, achieving class balance. We incorporate additional strategies into the BS method to balance the performance of head and tail classes while considering class differences.

- **Routing Diverse Distribution-aware Experts (RIDE) (Wang et al., 2021b):** This method employs a routing network to distribute input images to multiple expert classifiers, each focusing on different class distributions. We incorporate additional strategies into the RIDE method to balance the performance of various expert classifiers while optimizing the efficiency and stability of the routing network.

- **Balanced Contrastive Learning (BCL) (Zhu et al., 2022):** This method utilizes a contrastive learning framework to learn robust feature representations. We incorporate additional strategies into the BCL method to balance the effect and cost of contrastive learning while considering class similarity.

- **Cross-Entropy Loss with Dynamic Reweighting (CE-DRW) (Cao et al., 2019):** This method adjusts the cross-entropy loss function through dynamic class weights. We incorporate additional strategies into the CE-DRW method to balance the performance of head and tail classes while considering class differences.

- **Label-Distribution-Aware Margin Loss with Dynamic Reweighting (LDAM-DRW) (Cao et al., 2019):** This method employs a label-distribution-aware margin loss function, increasing the gap between different classes, and adjusts the loss function with dynamic class weights. We incorporate additional strategies into the LDAM-DRW method to balance the utility and cost of the margin loss function while considering class differences.

- **Key Point Sensitive Loss (KPS) (Li et al., 2022):** This method employs a keypoint-sensitive loss function to increase the weight of keypoint features in tail classes. We incorporate additional strategies into the KPS method to balance the utility and cost of keypoint features while considering class similarity.

- **Self-Heterogeneous Integration with Knowledge Excavation (SHIKE) (Jin et al., 2023):** This long-tail visual recognition method uses adaptive heterogeneous integration. It excavates from various knowledge sources (such as class hierarchy, class similarity, and class features) to construct multiple sub-classifiers, employing an adaptive weight distribution mechanism for fusion. We incorporate other strategies into the SHIKE method to balance the influences of different knowledge sources and optimize multiple objectives.

In summary, our method can dynamically and adaptively integrate the advantages of these baseline methods, taking into consideration their heterogeneous attention properties, providing a comprehensive and flexible solution for long-tail problems. When handling key factors in long-tail problems, such as balancing between head and tail classes, class differences, and class similarity, Strategy Fusion demonstrates excellent performance.

D. Pseudo Code

Here are pseudo codes explaining core aspects of the method proposed in the paper, and the main symbols are list in Table 7:

Algorithm 1 Proposed Method for Long-Tail Learning

Input: Training data with long-tail distribution
Output: Classifier model with balanced performance
Initialize model with shared feature extractor F
Initialize strategy modules $L = \{L_1, L_2, ..., L_N\}$
for each epoch **do**
  for each batch **do**
    $f = F(x)$ {Shared feature extraction}
    for L_i in L **do**
      $l_i = L_i(f, y)$ {Individual strategy losses}
    end for
    A_{ij} = ComputeInfluence(L_i, C_j) {Calculate the influence of L_i on class C_j}
    β_i = CosineSimilarity(L_i, AvgAccuracy)
    α_i = Softmax(β) {HICA}
    $L_{fused} = \sum_i \alpha_i l_i$
    $g_i = \nabla_\theta l_i, \forall i$
    g_i = OrthogonalProj(g_i, g_j), $\forall i, j$ {GHOP}
    $\theta = \theta - \eta \nabla_\theta L_{fused}$
  end for
  $i^* = \arg\max_i A_{ij}, \forall C_j$ {EOSS}
  $L_{pred} = L_{i^*}$
end for

Algorithm 2 Evolving Optimal Strategy Selection

Require: $L = \{L_1, L_2, ..., L_N\}$: the strategy set
Require: $C = \{C_1, C_2, ..., C_M\}$: the class set
Require: A_{ij}: accuracy of strategy L_i on class C_j
Store $A \in \mathbb{R}^{N \times M}$
for each training iteration **do**
  Compute $i^* = \arg\max_i A_{ij}$ for each C_j {Best strategy}
  $L_{pred}(x) = L_{i^*}(x)$ if $y = C_j$
  A_{ij} = UpdateInfluence(A_{ij}, latest predictions) {Specify the update method here}
end for

E. More Empirical Results

E.1. Results on ImageNet-LT and iNaturalist 2018 Datasets

We also evaluate the effect of MOOSF on large datasets. The specific results are shown in Table 8. We find that MOOSF still achieves superior results on large datasets.

E.2. Complexity Analysis

Time Complexity: The multi-task learning framework itself does not add additional time complexity. The HICA module, which calculates the class accuracy for each strategy, has a time complexity of $O(MN)$, where M is the number of strategies and N is the number of classes. The GHOP module, which calculates the orthogonal projection between gradients to coordinate their directions, leads to a time complexity of $O(M^2D)$, where D is the number of parameters. The EOSS module, used for strategy selection for each class, has a complexity of $O(MN)$. The gradient backpropagation process is equivalent to the original model, which is $O(D)$.

Table 7. List of Main Symbols

Symbol	Meaning
D	Training set
(x_i, y_i)	A sample in the training set, where x_i is the feature and y_i is the class
R^d	d-dimensional real space
C	Set of classes
$c_1, ..., c_K$	K distinct classes
n_k	Number of samples in class c_k
$k - \alpha$	Represents a long-tail distribution
f_θ	Classifier, where θ is the parameter
$\ell(f_\theta(x_i), y_i)$	Loss function
$\Omega(\theta)$	Function introducing class balance constraint

Table 8. Accuracy (%) on ImageNet-LT and iNaturalist 2018 datasets wtih MOOSF. We report the average results of three random trials.

Method	ImageNet-LT				iNaturalist 2018			
	Head	Medium	Tail	All	Head	Medium	Tail	All
MOOSF$_{(CE+LDAM-DRW)}$	66.0	49.2	31.7	51.8	75.1	72.3	71.9	72.6
MOOSF$_{(CE+KPS)}$	66.3	49.8	33.2	52.7	75.4	72.6	72.2	73.8
MOOSF$_{(BS+CE-DRW)}$	61.8	51.3	35.7	53.2	70.2	71.1	71.6	71.5
MOOSF$_{(KPS+LDAM-DRW)}$	60.3	50.4	34.8	52.1	69.4	71.7	71.8	71.4
MOOSF$_{(SHIKE+BCL)}$	66.9	52.6	36.9	55.3	71.2	72.3	72.4	72.4
MOOSF$_{(BS+BCL+LDAM-DRW)}$	64.8	50.5	34.1	53.3	69.1	71.3	71.8	71.2
MOOSF$_{(CE+BS+LDAM-DRW)}$	65.3	51.0	34.8	53.8	69.8	72.0	72.1	71.8
MOOSF$_{(CE+LDAM-DRW+KPS+BCL)}$	64.9	50.6	34.4	53.5	69.3	71.6	71.9	71.4
MOOSF$_{(CE+BCL+KPS+SHIKE+CE-DRW)}$	66.1	51.8	36.0	54.7	71.0	72.4	72.5	72.5

Therefore, the overall time complexity is $O(MN) + O(M^2D) + O(MN) + O(D) = O(M^2D)$. Compared with single strategy training, the main addition comes from the $O(M^2)$ term from GHOP. However, in practice, due to GPU parallelization, the increase in M does not lead to a quadratic increase in the actual running time. By appropriately setting the number of strategies M, the increase in computation time can be kept within an acceptable range, as confirmed by the experiments in the paper.

Experimental Analysis: In experiments on the CIFAR-100 dataset, we recorded the computational complexity and training time of various long-tail learning strategies and their combinations. As shown in Table 9, **macs** represents the computational cost of the model (in millions of operations), and **Time** represents the training time (in minutes). γ_1 is the ratio of the amount of parameters trained together to the sum of the amounts of parameters trained separately, while γ_2 is the ratio of the combined training time to the sum of the separate training times.

The results indicate that the computational cost and time of a single strategy are roughly equivalent. After combining strategies, although the amount of parameters decreases (γ_1 decreases), the training time does not decrease linearly ($\gamma_2 > \gamma_1$). This is because coordinating the gradient directions introduces additional computational overhead, even though the number of parameters is reduced through multi-task learning. The ratio of γ_1 to γ_2 reflects the computational efficiency, with dual strategy fusion being the most efficient. As the number of fused strategies increases, computational efficiency decreases (the ratio γ_1/γ_2 decreases). This aligns with the observation in the paper that too many strategy fusions can lead to conflicts. The quantitative results confirm the conclusion in the text that the efficiency of multi-objective strategy fusion decreases as the number of fusions increases.

E.3. T-SNE Analysis

Figure 5 presents the t-SNE analysis of the post-fusion representations. Almost all classes show significantly increased clustering separability compared to before fusion. This is robust evidence that our method has successfully achieved the objective Proposition 4.

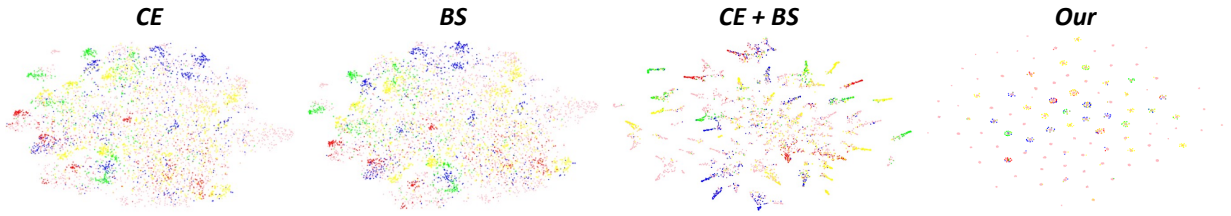

Figure 5. T-SNE comparison analysis. We conducted a comparison of the classification results on CIFAR-100-LT when using Cross-Entropy (CE) and Balanced-Softmax (BS) independently, as well as the results obtained through simple Multi-Task Learning (MTL) fusion and our proposed fusion method.

E.4. Further Analysis

In this section, we provide a more microscopic view of why our strategy fusion is effective. Due to formatting constraints, we use a series of two-dimensional line graphs to illustrate this process. Figures 6, 7, 8, 9, and 10 show the changes in the weights of the last layer of the encoder after a single gradient update using individual strategies and the fusion strategy at epochs 25, 50, 75, 125, and 150, respectively.

As can be seen, at different training stages, for weights at different positions, the fusion strategy brings more stable and centered updates. This indicates that our fusion strategy achieves stability and balance in the microscopic neuron update process, which ultimately leads to the improvements in balancing demonstrated at a macroscopic level in the main text.

Table 9. Indicators of computational complexity and training time of different long-tail learning strategies and their combinations on the CIFAR-100 dataset

Task	**macs/M**	**Time/min**	γ_1	γ_2	γ_1/γ_2
CE	6.554	18	1.0	1.0	1.0
BS	6.554	17	1.0	1.0	1.0
KPS	6.554	18	1.0	1.0	1.0
CE-DRW	6.554	18	1.0	1.0	1.0
LDAM-DRW	6.554	19	1.0	1.0	1.0
BCL	6.554	36	1.0	1.0	1.0
$\text{MOOSF}_{(\text{CE+BS})}$	13.108	26	0.5	0.743	0.673
$\text{MOOSF}_{(\text{CE+KPS})}$	13.108	28	0.5	0.778	0.643
$\text{MOOSF}_{(\text{KPS+BCL})}$	13.108	39	0.5	0.722	0.692
$\text{MOOSF}_{(\text{CE+BS+CE-DRW})}$	19.662	34	0.333	0.642	0.52
$\text{MOOSF}_{(\text{CE+BS+LDAM-DRW})}$	19.662	34	0.333	0.63	0.53
$\text{MOOSF}_{(\text{CE+BCL+CE-DRW})}$	19.662	65	0.333	0.903	0.369
$\text{MOOSF}_{(\text{CE+BS+KPS+CE-DRW})}$	26.216	36	0.25	0.507	0.493
$\text{MOOSF}_{(\text{CE+BS+KPS+LDAM-DRW})}$	26.216	32	0.25	0.444	0.563

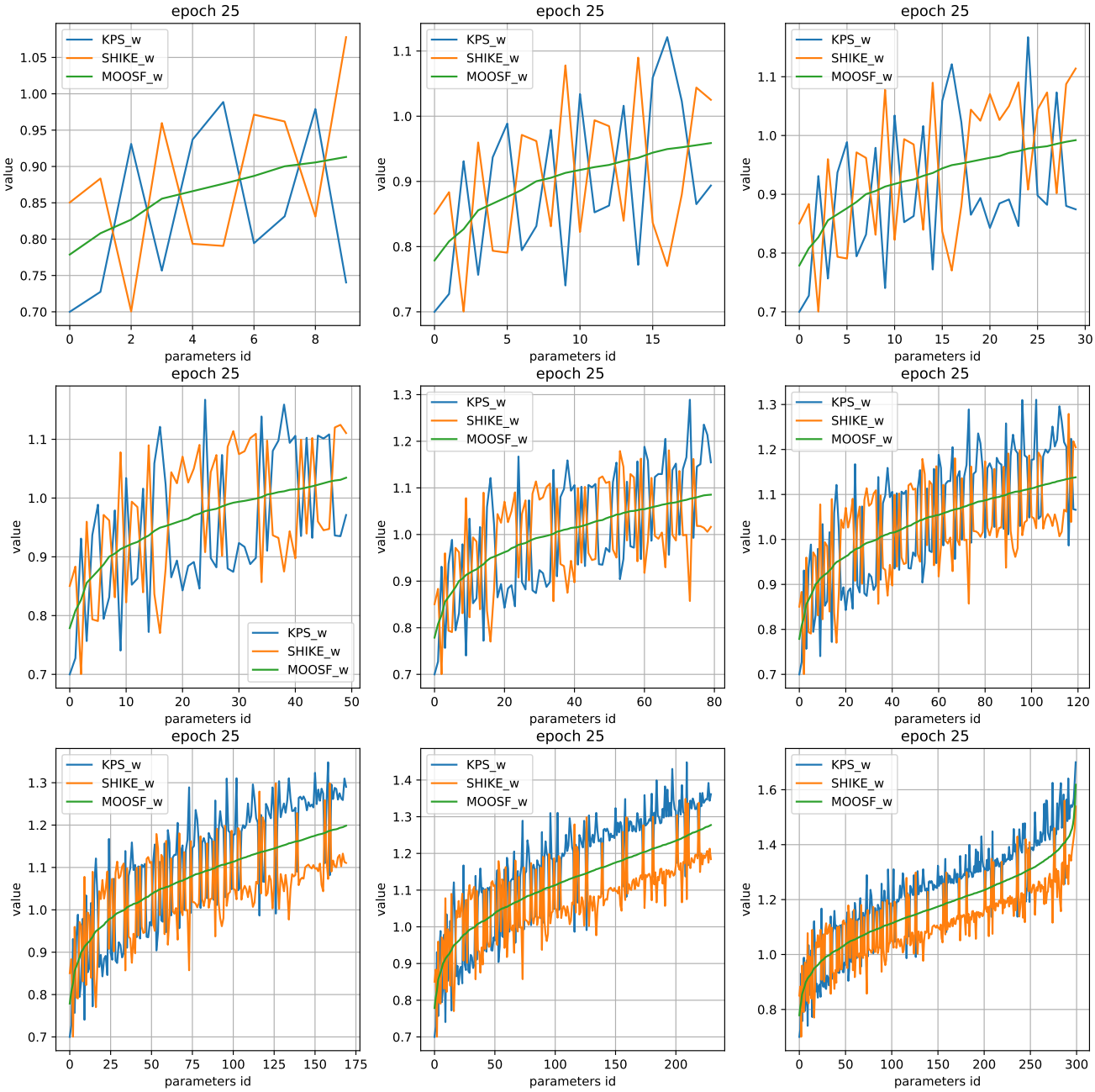

Figure 6. We extracted the weight matrix from the last convolutional layer of the encoder at the 25th epoch. The parameter id indicates the sequence number of the weight parameter, and the vertical axis represents the gradient value of the corresponding parameter during the backpropagation process at that moment. Clearly, our method has achieved favorable results in resolving gradient conflicts and stabilizing the optimization process.

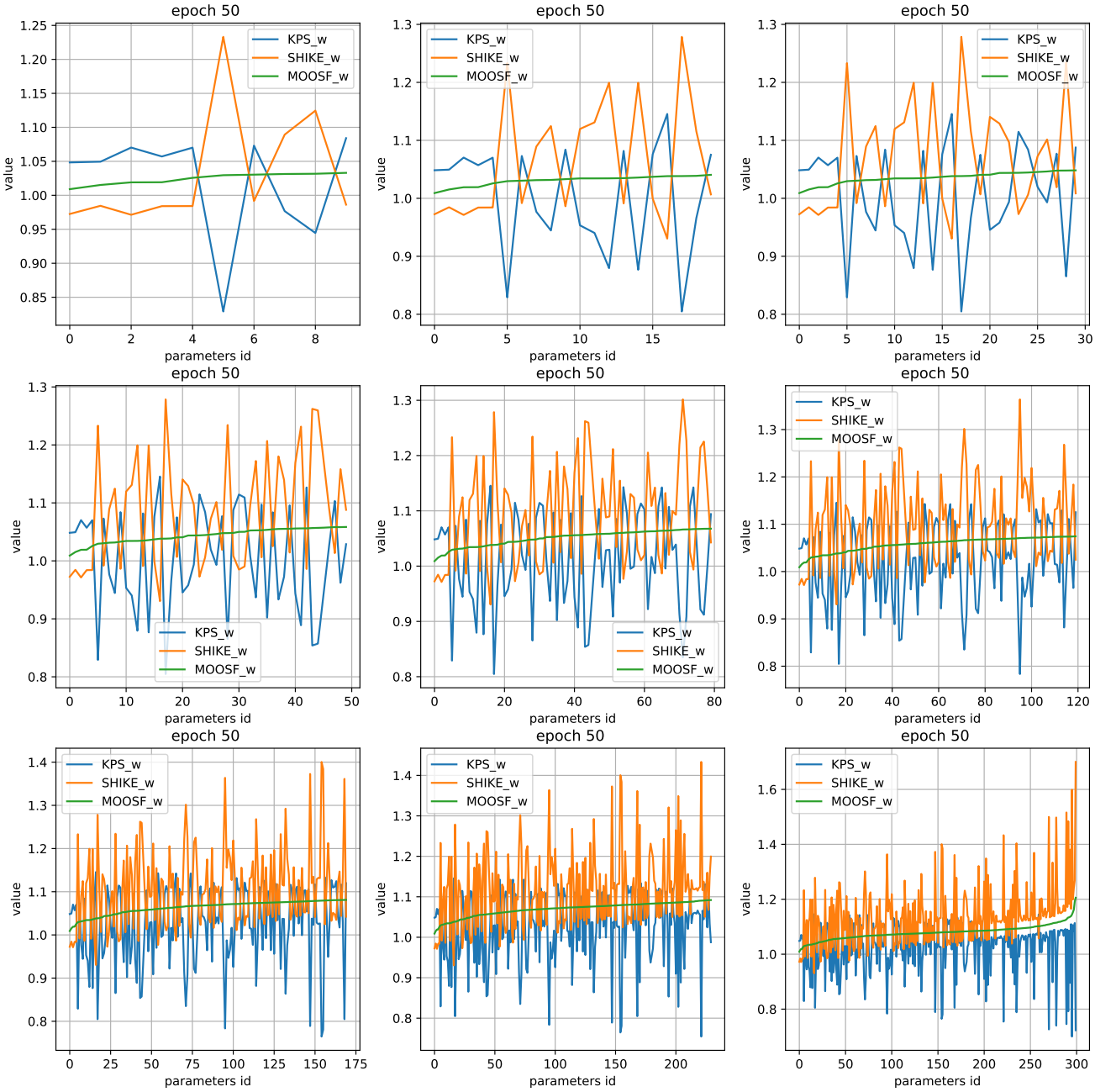

Figure 7. The change in the gradient values of the weight parameters of the last convolutional layer of the encoder at the 50th epoch demonstrates that our method has been effective in resolving gradient conflicts and achieving stable optimization. The other settings are the same as in Figure 6.

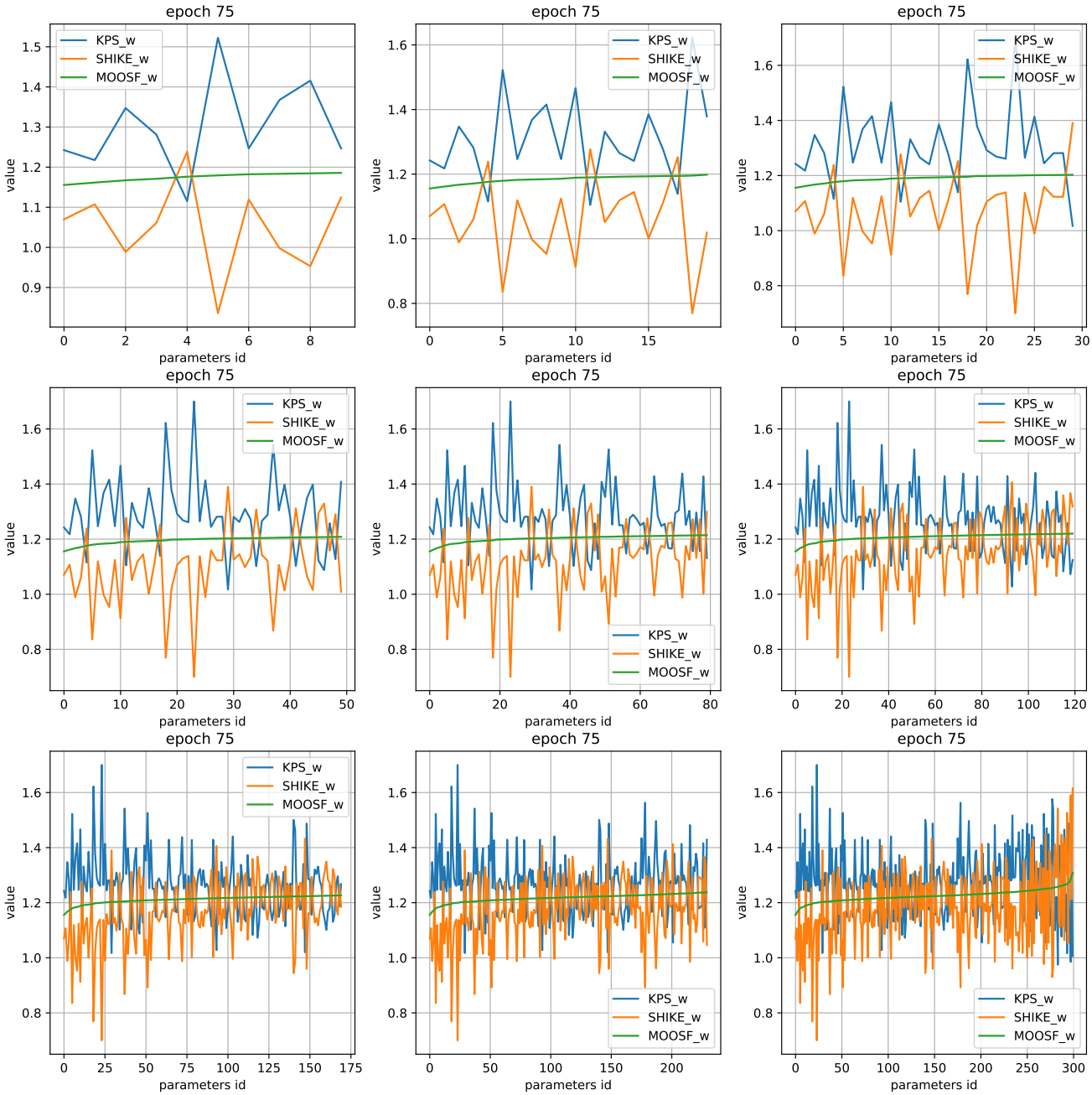

Figure 8. The process of change in the gradient values of the weight parameters for the last convolutional layer of the encoder at the 75th epoch is depicted. The other settings are identical to those in Figure 6.

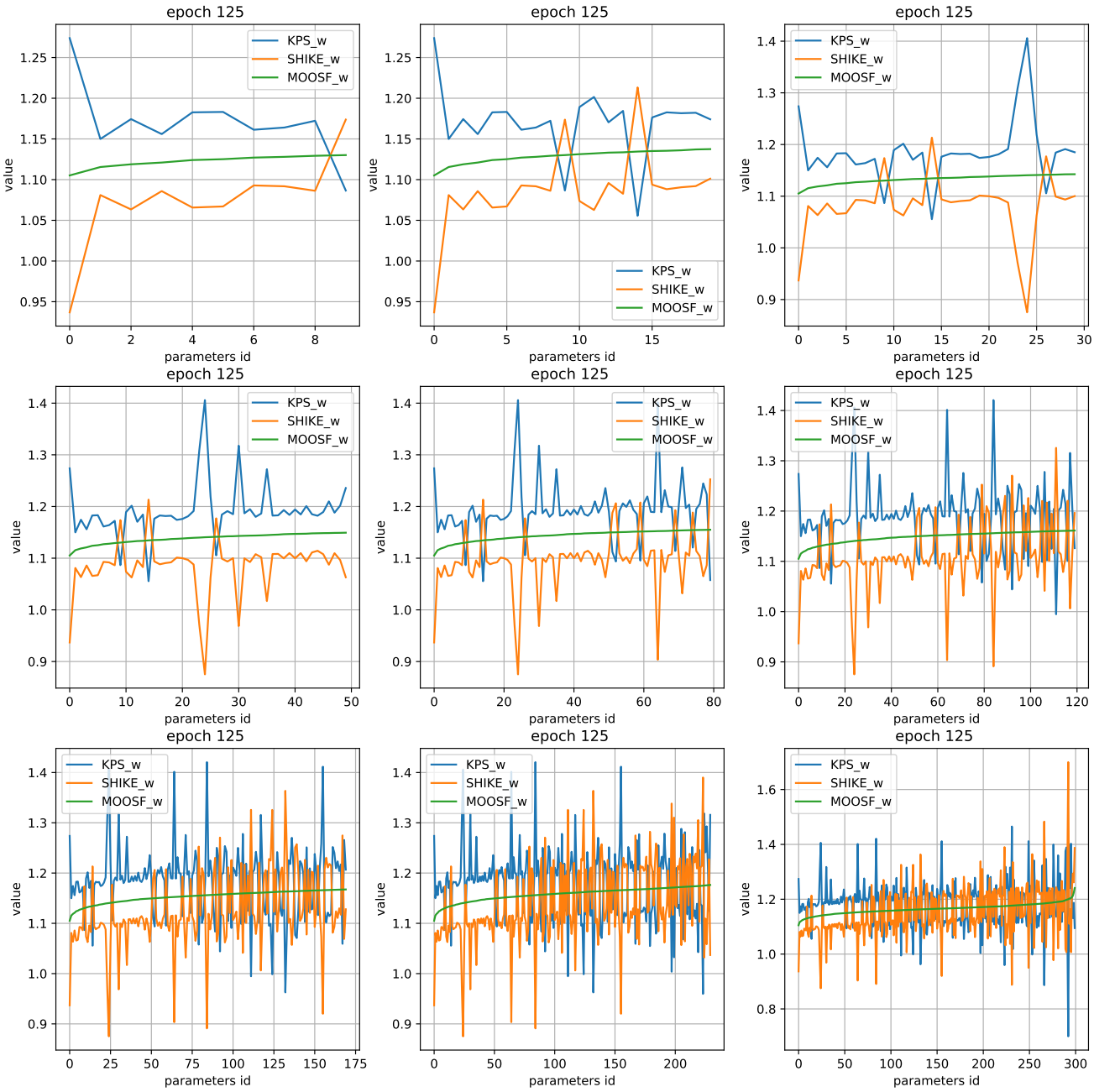

Figure 9. The process of change in the gradient values of the weight parameters for the last convolutional layer of the encoder at the 125th epoch is depicted. The other settings are identical to those in Figure 6.

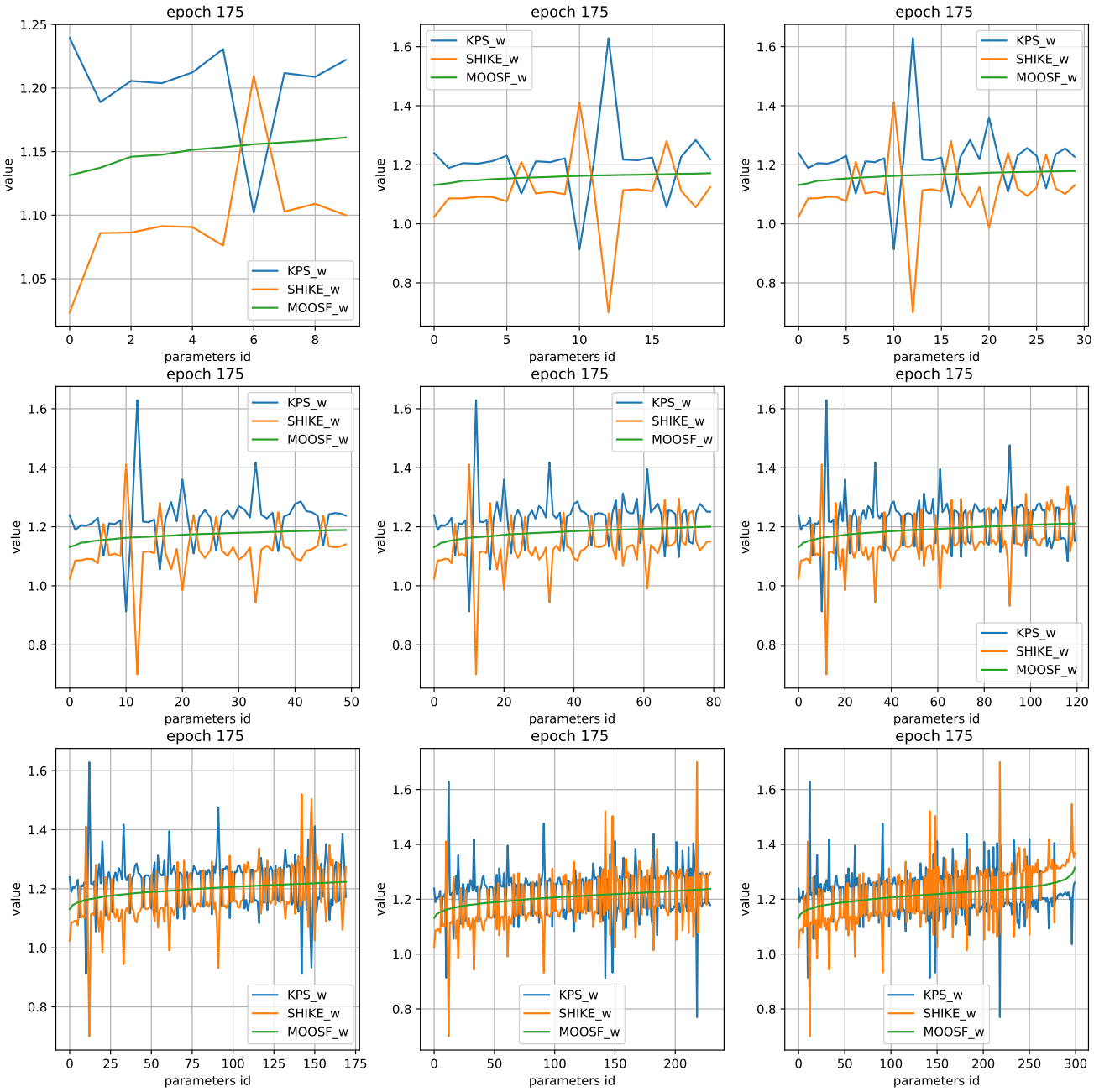

Figure 10. The process of change in the gradient values of the weight parameters for the last convolutional layer of the encoder at the 175th epoch is depicted. The other settings are identical to those in Figure 6.