

6 Theoretical Analysis

This appendix provides formal proofs for the three key propositions stated in Section 3.3, establishing the theoretical foundations that underpin TS-MOF’s robustness and effectiveness.

6.1 Mathematical Foundations and Problem Setup

The core challenge in long-tailed recognition lies in simultaneously optimizing multiple conflicting objectives. In Stage 2 of TS-MOF, we address the multi-objective optimization problem:

$$\min_{\theta_H} \mathbf{F}(\theta_H) = (\mathbb{E}[\mathcal{L}_1(\theta_H)], \dots, \mathbb{E}[\mathcal{L}_N(\theta_H)])^\top \quad (10)$$

where $\theta_H = \{\theta_{H_1}, \dots, \theta_{H_N}\}$ denotes the parameters of all classifier heads, and each \mathcal{L}_k represents a specific LTR strategy (e.g., Cross-Entropy, LDAM, Balanced Softmax).

The R-PLA mechanism dynamically weights these objectives based on real-time performance patterns:

$$\mathcal{L}'_k(\theta_H) = \beta_{k,e} \mathcal{L}_k(\theta_H), \quad \beta_{k,e} = \max(0, \sqrt[3]{s_{j,e}}) \quad (11)$$

where the similarity measure $s_{j,e}$ captures the alignment between task j ’s performance pattern and a reference:

$$s_{j,e} = \frac{\mathbf{a}_{j,e}^{(\text{batch})} \cdot \mathbf{a}_{\text{ref},e}^{(\text{batch})}}{\max\{\|\mathbf{a}_{j,e}^{(\text{batch})}\|_2 \|\mathbf{a}_{\text{ref},e}^{(\text{batch})}\|_2, \epsilon_{\text{sim}}\}} \quad (12)$$

Our theoretical analysis addresses three fundamental questions: (1) Does RD-PCGrad guarantee stable, conflict-free optimization? (2) Is R-PLA robust to noise and scaling variations? (3) Does two-stage decoupling preserve feature quality while enabling effective classifier adaptation?

6.2 Proof of Proposition 1: Gradient Stability and Pareto Improvement

Motivation: The key challenge in multi-objective LTR is that gradients from different strategies often conflict (e.g., improving tail performance may hurt head performance). RD-PCGrad must resolve these conflicts while ensuring progress toward better overall performance.

Let $\mathbf{g}_k = \nabla_{\theta_H} \mathcal{L}'_k(\theta_H) \in \mathbb{R}^D$ denote the gradient of weighted task k , where $D = |\theta_H|$ is the total number of classifier parameters.

Lemma 1 (Conflict Resolution Property). *For any two conflicting gradients $\mathbf{g}_i, \mathbf{g}_j$ with $\mathbf{g}_i^\top \mathbf{g}_j < 0$, the RD-PCGrad projection:*

$$\mathbf{g}'_i = \mathbf{g}_i - \frac{\mathbf{g}_i^\top \mathbf{g}_j}{\|\mathbf{g}_j\|_2^2 + \epsilon_{\text{norm}}} \mathbf{g}_j \quad (13)$$

eliminates the conflict, ensuring $(\mathbf{g}'_i)^\top \mathbf{g}_j \geq 0$.

Proof. The key insight is that projection onto the orthogonal complement removes the conflicting component. Computing the inner product after projection:

$$(\mathbf{g}'_i)^\top \mathbf{g}_j = \left(\mathbf{g}_i - \frac{\mathbf{g}_i^\top \mathbf{g}_j}{\|\mathbf{g}_j\|_2^2 + \epsilon_{\text{norm}}} \mathbf{g}_j \right)^\top \mathbf{g}_j \quad (14)$$

$$= \mathbf{g}_i^\top \mathbf{g}_j - \frac{\mathbf{g}_i^\top \mathbf{g}_j}{\|\mathbf{g}_j\|_2^2 + \epsilon_{\text{norm}}} \|\mathbf{g}_j\|_2^2 \quad (15)$$

$$= \mathbf{g}_i^\top \mathbf{g}_j - \frac{(\mathbf{g}_i^\top \mathbf{g}_j) \|\mathbf{g}_j\|_2^2}{\|\mathbf{g}_j\|_2^2 + \epsilon_{\text{norm}}} \quad (16)$$

$$= (\mathbf{g}_i^\top \mathbf{g}_j) \left(1 - \frac{\|\mathbf{g}_j\|_2^2}{\|\mathbf{g}_j\|_2^2 + \epsilon_{\text{norm}}} \right) \quad (17)$$

$$= (\mathbf{g}_i^\top \mathbf{g}_j) \frac{\epsilon_{\text{norm}}}{\|\mathbf{g}_j\|_2^2 + \epsilon_{\text{norm}}} \quad (18)$$

Since $\mathbf{g}_i^\top \mathbf{g}_j < 0$ and $\epsilon_{\text{norm}} > 0$, we have $(\mathbf{g}'_i)^\top \mathbf{g}_j \geq 0$, resolving the conflict. \square

440 **Lemma 2** (Descent Direction Preservation). *The projection operation preserves the descent property*
 441 *for the objective corresponding to the projected gradient.*

442 *Proof.* We must show that \mathbf{g}'_i remains a descent direction for \mathcal{L}'_i . The critical quantity is:

$$(\mathbf{g}_i)^\top \mathbf{g}'_i = \mathbf{g}_i^\top \left(\mathbf{g}_i - \frac{\mathbf{g}_i^\top \mathbf{g}_j}{\|\mathbf{g}_j\|_2^2 + \epsilon_{\text{norm}}} \mathbf{g}_j \right) \quad (19)$$

$$= \|\mathbf{g}_i\|_2^2 - \frac{(\mathbf{g}_i^\top \mathbf{g}_j)^2}{\|\mathbf{g}_j\|_2^2 + \epsilon_{\text{norm}}} \quad (20)$$

$$= \|\mathbf{g}_i\|_2^2 \left(1 - \frac{(\mathbf{g}_i^\top \mathbf{g}_j)^2}{\|\mathbf{g}_i\|_2^2 (\|\mathbf{g}_j\|_2^2 + \epsilon_{\text{norm}})} \right) \quad (21)$$

$$= \|\mathbf{g}_i\|_2^2 \left(1 - \frac{\cos^2(\mathbf{g}_i, \mathbf{g}_j) \|\mathbf{g}_j\|_2^2}{\|\mathbf{g}_j\|_2^2 + \epsilon_{\text{norm}}} \right) \quad (22)$$

$$> \|\mathbf{g}_i\|_2^2 (1 - \cos^2(\mathbf{g}_i, \mathbf{g}_j)) = \|\mathbf{g}_i\|_2^2 \sin^2(\mathbf{g}_i, \mathbf{g}_j) > 0 \quad (23)$$

443 The last inequality holds because conflicting gradients are not parallel ($\sin(\mathbf{g}_i, \mathbf{g}_j) > 0$), ensuring
 444 descent is preserved. \square

445 **Main Proof of Proposition 1:** After iterative conflict resolution, RD-PCGrad produces conflict-free
 446 gradients $\{\mathbf{g}''_1, \dots, \mathbf{g}''_N\}$ that are aggregated as:

$$\mathbf{g}_{\text{final}} = \frac{\sum_{k=1}^N (\mathbf{g}''_k \odot \mathbf{h}_k)}{\sum_{k=1}^N \mathbf{h}_k + \epsilon_{\text{div}}} \quad (24)$$

447 where $\mathbf{h}_k \in \{0, 1\}^D$ are gradient existence masks and \odot denotes element-wise multiplication.

448 Since all pairwise conflicts have been resolved, we have $(\mathbf{g}''_i)^\top \mathbf{g}''_j \geq 0$ for all $i \neq j$. For any original
 449 gradient \mathbf{g}_i , the descent property with respect to the final direction is:

$$(\mathbf{g}_i)^\top \mathbf{g}_{\text{final}} = (\mathbf{g}_i)^\top \frac{\sum_{k=1}^N (\mathbf{g}''_k \odot \mathbf{h}_k)}{\sum_{k=1}^N \mathbf{h}_k + \epsilon_{\text{div}}} \quad (25)$$

$$= \frac{\sum_{k=1}^N (\mathbf{g}_i)^\top (\mathbf{g}''_k \odot \mathbf{h}_k)}{\sum_{k=1}^N \mathbf{h}_k + \epsilon_{\text{div}}} \quad (26)$$

$$\geq \frac{(\mathbf{g}_i)^\top (\mathbf{g}''_i \odot \mathbf{h}_i)}{\sum_{k=1}^N \mathbf{h}_k + \epsilon_{\text{div}}} \geq 0 \quad (27)$$

450 where the first inequality uses the non-negativity of cross-terms after conflict resolution, and the
 451 second follows from Lemma 2. This establishes that $-\mathbf{g}_{\text{final}}$ provides a descent or non-ascent direction
 452 for all objectives, constituting a Pareto-improving or non-worsening update direction. \square

453 6.3 Proof of Proposition 2: Adaptive Weighting Robustness

454 **Motivation:** R-PLA must maintain stable task weighting despite variations in performance measure-
 455 ment scales and noisy real-time estimates. The cosine similarity and cube root transformation are
 456 specifically designed to achieve this robustness.

457 **Part (i) - Scale Invariance:** Consider performance vectors scaled by factor $\lambda > 0$: $\tilde{\mathbf{a}}_{j,e} = \lambda \mathbf{a}_{j,e}$ and
 458 $\tilde{\mathbf{a}}_{\text{ref},e} = \lambda \mathbf{a}_{\text{ref},e}$. The cosine similarity becomes:

$$\tilde{s}_{j,e} = \frac{(\lambda \mathbf{a}_{j,e})^\top (\lambda \mathbf{a}_{\text{ref},e})}{\|\lambda \mathbf{a}_{j,e}\|_2 \|\lambda \mathbf{a}_{\text{ref},e}\|_2} \quad (28)$$

$$= \frac{\lambda^2 (\mathbf{a}_{j,e})^\top \mathbf{a}_{\text{ref},e}}{\lambda \|\mathbf{a}_{j,e}\|_2 \cdot \lambda \|\mathbf{a}_{\text{ref},e}\|_2} \quad (29)$$

$$= \frac{\lambda^2 (\mathbf{a}_{j,e})^\top \mathbf{a}_{\text{ref},e}}{\lambda^2 \|\mathbf{a}_{j,e}\|_2 \|\mathbf{a}_{\text{ref},e}\|_2} = s_{j,e} \quad (30)$$

Consequently, $\tilde{\beta}_{j,e} = \max(0, \sqrt[3]{\tilde{s}_{j,e}}) = \max(0, \sqrt[3]{s_{j,e}}) = \beta_{j,e}$, demonstrating perfect scale invariance.

Part (ii) - Noise Robustness: Consider noisy performance vectors $\mathbf{a}_{j,e} + \boldsymbol{\eta}_j$ where $\|\boldsymbol{\eta}_j\|_2 \leq \delta$ for small $\delta > 0$. The perturbed similarity satisfies:

$$s_{j,e}^{\text{noisy}} = \frac{(\mathbf{a}_{j,e} + \boldsymbol{\eta}_j)^\top (\mathbf{a}_{\text{ref},e} + \boldsymbol{\eta}_{\text{ref}})}{\|\mathbf{a}_{j,e} + \boldsymbol{\eta}_j\|_2 \|\mathbf{a}_{\text{ref},e} + \boldsymbol{\eta}_{\text{ref}}\|_2} \quad (31)$$

$$= \frac{(\mathbf{a}_{j,e})^\top \mathbf{a}_{\text{ref},e} + (\mathbf{a}_{j,e})^\top \boldsymbol{\eta}_{\text{ref}} + \boldsymbol{\eta}_j^\top \mathbf{a}_{\text{ref},e} + \boldsymbol{\eta}_j^\top \boldsymbol{\eta}_{\text{ref}}}{\|\mathbf{a}_{j,e} + \boldsymbol{\eta}_j\|_2 \|\mathbf{a}_{\text{ref},e} + \boldsymbol{\eta}_{\text{ref}}\|_2} \quad (32)$$

Using the fact that $\|\mathbf{a} + \boldsymbol{\eta}\|_2 = \|\mathbf{a}\|_2 + O(\delta)$ for small δ , and applying first-order perturbation analysis:

$$|s_{j,e}^{\text{noisy}} - s_{j,e}| \leq \frac{|(\mathbf{a}_{j,e})^\top \boldsymbol{\eta}_{\text{ref}} + \boldsymbol{\eta}_j^\top \mathbf{a}_{\text{ref},e}| + O(\delta^2)}{\|\mathbf{a}_{j,e}\|_2 \|\mathbf{a}_{\text{ref},e}\|_2 + O(\delta)} \quad (33)$$

$$\leq \frac{\|\mathbf{a}_{j,e}\|_2 \|\boldsymbol{\eta}_{\text{ref}}\|_2 + \|\boldsymbol{\eta}_j\|_2 \|\mathbf{a}_{\text{ref},e}\|_2 + O(\delta^2)}{\|\mathbf{a}_{j,e}\|_2 \|\mathbf{a}_{\text{ref},e}\|_2 + O(\delta)} \quad (34)$$

$$\leq \frac{2\delta(\|\mathbf{a}_{j,e}\|_2 + \|\mathbf{a}_{\text{ref},e}\|_2) + O(\delta^2)}{\|\mathbf{a}_{j,e}\|_2 \|\mathbf{a}_{\text{ref},e}\|_2} = O(\delta) \quad (35)$$

The cube root transformation $f(x) = \max(0, \sqrt[3]{x})$ has derivative $f'(x) = \frac{1}{3}x^{-2/3}$ for $x > 0$. By the mean value theorem:

$$|\beta_{j,e}^{\text{noisy}} - \beta_{j,e}| = |f(s_{j,e}^{\text{noisy}}) - f(s_{j,e})| \quad (36)$$

$$\leq \max_{x \in [s_{j,e}^{\text{noisy}}, s_{j,e}]} |f'(x)| \cdot |s_{j,e}^{\text{noisy}} - s_{j,e}| \quad (37)$$

$$\leq \frac{1}{3} \min(s_{j,e}^{\text{noisy}}, s_{j,e})^{-2/3} \cdot O(\delta) = O(\delta) \quad (38)$$

This establishes that R-PLA weights have bounded sensitivity to noise, with the cube root transformation providing smoother weight dynamics compared to linear or quadratic functions. \square

6.4 Proof of Proposition 3: Representational Robustness

Motivation: Two-stage decoupling must ensure that sophisticated MOO operations in Stage 2 do not degrade the high-quality features learned in Stage 1. This requires formal guarantees about gradient isolation and feature preservation.

Feature Quality Preservation: With encoder parameters θ_E^* frozen during Stage 2, the feature extraction mapping $E(\cdot; \theta_E^*) : \mathcal{X} \rightarrow \mathbb{R}^{D_F}$ remains constant. For any feature quality measure $Q : \mathbb{R}^{D_F} \rightarrow \mathbb{R}$ (e.g., discriminative power, cluster separability):

$$Q(E(x; \theta_E^*)) = \text{constant} \quad \forall x \in \mathcal{X}, \forall t \in \text{Stage 2 iterations} \quad (39)$$

Gradient Isolation Analysis: The Stage 2 loss functions decompose as compositions:

$$\mathcal{L}_k(\theta_H) = \mathbb{E}_{(x,y) \sim \mathcal{D}} \ell(H_k(E(x; \theta_E^*); \theta_{H_k}), y) \quad (40)$$

$$= \mathbb{E}_{(x,y) \sim \mathcal{D}} \ell(H_k(\mathbf{z}; \theta_{H_k}), y) \Big|_{\mathbf{z} = E(x; \theta_E^*)} \quad (41)$$

where $\mathbf{z} \in \mathbb{R}^{D_F}$ represents the fixed feature vectors. The gradients with respect to classifier parameters are:

$$\nabla_{\theta_{H_k}} \mathcal{L}_k(\theta_H) = \mathbb{E}_{(x,y) \sim \mathcal{D}} \nabla_{\theta_{H_k}} \ell(H_k(\mathbf{z}; \theta_{H_k}), y) \Big|_{\mathbf{z} = E(x; \theta_E^*)} \quad (42)$$

$$= \mathbb{E}_{\mathbf{z} \sim P_{\text{features}}} \nabla_{\theta_{H_k}} \ell(H_k(\mathbf{z}; \theta_{H_k}), y(\mathbf{z})) \quad (43)$$

where P_{features} is the distribution of extracted features and $y(\mathbf{z})$ is the label corresponding to feature \mathbf{z} .

Crucially, since θ_E^* does not appear in the optimization variables, we have:

$$\frac{\partial}{\partial \theta_E} \nabla_{\theta_{H_k}} \mathcal{L}_k(\theta_H) \Big|_{\theta_E = \theta_E^*} = \mathbf{0} \quad (44)$$

This gradient isolation ensures that MOO operations (R-PLA weighting, RD-PCGrad projections) cannot corrupt the feature representation.

Stability Under Complex MOO: The multi-objective parameter updates in Stage 2 follow:

$$\theta_H^{(t+1)} = \theta_H^{(t)} - \eta^{(t)} \mathbf{g}_{\text{final}}^{(t)} \quad (45)$$

$$\theta_E^{(t+1)} = \theta_E^{(t)} = \theta_E^* \quad (\text{architectural constraint}) \quad (46)$$

where $\mathbf{g}_{\text{final}}^{(t)} \in \mathbb{R}^{|\theta_H|}$ is computed via the sophisticated RD-PCGrad procedure but contains no components corresponding to encoder parameters.

The architectural decoupling provides an invariant: regardless of the complexity of MOO operations (dynamic weighting, gradient conflicts, iterative projections), the feature space $\{E(x; \theta_E^*) : x \in \mathcal{X}\}$ remains unchanged throughout Stage 2. This guarantees representational robustness while enabling effective classifier adaptation through MOO. \square

6.5 Convergence and Complexity Analysis

Convergence Guarantee: Under standard assumptions (Lipschitz continuous gradients, bounded feasible region), TS-MOF converges to a locally Pareto-optimal solution. The key insight is that RD-PCGrad ensures:

$$\sum_{k=1}^N \beta_{k,e} \nabla \mathcal{L}_k(\theta_H)^\top \mathbf{g}_{\text{final}} \leq -c \|\mathbf{g}_{\text{final}}\|_2^2 \quad (47)$$

for some constant $c > 0$, providing monotonic improvement in the weighted multi-objective function.

Computational Efficiency: Stage 2 requires $O(N^2 D)$ operations per iteration, where N is the number of tasks and $D = |\theta_H|$ is the classifier parameter count. This compares favorably to end-to-end MOO requiring $O(N^2(D + D_E))$ with encoder dimension $D_E \gg D$, yielding substantial computational savings through architectural decoupling.

7 Algorithm Description

This section provides detailed algorithmic descriptions of the TS-MOF framework, with particular emphasis on the multi-objective fine-tuning process and the two core innovations: Refined Performance Level Agreement (R-PLA) and Robust Deterministic Projective Conflict Gradient (RD-PCGrad).

7.1 Overall TS-MOF Framework

Algorithm 1 presents the complete two-stage training procedure of TS-MOF, highlighting the strategic decoupling between feature learning and classifier adaptation.

Algorithm 1 TS-MOF: Two-Stage Multi-Objective Fine-tuning

Require: Long-tailed dataset $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}$, LTR tasks $\mathcal{T} = \{T_1, \dots, T_N\}$, Pre-training epochs E_{S1} , fine-tuning epochs E_{S2}

Ensure: Trained model with balanced performance across head, medium, and tail classes

- 1: **// Stage 1: Generic Feature Pre-training**
- 2: Initialize encoder $E(\cdot; \theta_E)$ and classifier $H_{S1}(\cdot; \theta_{H_{S1}})$
- 3: **for** $e = 1$ to E_{S1} **do**
- 4: **for** each batch $(x_i, y_i) \sim \mathcal{D}_{\text{train}}$ **do**
- 5: $\mathbf{z}_i \leftarrow E(x_i; \theta_E)$ {Extract features}
- 6: $\hat{y}_i \leftarrow H_{S1}(\mathbf{z}_i; \theta_{H_{S1}})$ {Classification}
- 7: $\mathcal{L}_{CE} \leftarrow \text{CrossEntropy}(\hat{y}_i, y_i)$ {Standard loss}
- 8: Update $\theta_E, \theta_{H_{S1}}$ via SGD with \mathcal{L}_{CE}
- 9: **end for**
- 10: Evaluate on \mathcal{D}_{val} and save best θ_E^*
- 11: **end for**
- 12: **// Stage 2: Multi-Objective Classifier Fine-tuning**
- 13: Freeze encoder parameters: $\theta_E \leftarrow \theta_E^*$ (fixed)
- 14: Initialize multi-head classifiers $\{H_k(\cdot; \theta_{H_k})\}_{k=1}^N$ for tasks \mathcal{T}
- 15: Initialize R-PLA weighting mechanism and RD-PCGrad optimizer
- 16: **Call** MULTIOBJECTIVEFINETUNING($E(\cdot; \theta_E^*), \{H_k\}_{k=1}^N, \mathcal{D}_{\text{train}}, E_{S2}$)
- 17: **// Inference with EOSS**
- 18: Train EOSS weights $\{w_{k,c}\}$ based on validation performance
- 19: **return** Model with frozen θ_E^* and fine-tuned $\{\theta_{H_k}\}_{k=1}^N$

506 **7.2 Multi-Objective Fine-tuning with R-PLA and RD-PCGrad**

507 Algorithm 2 details the core Stage 2 process, emphasizing how R-PLA and RD-PCGrad work together
508 to achieve effective multi-objective optimization.

Algorithm 2 Multi-Objective Fine-tuning with R-PLA and RD-PCGrad

Require: Frozen encoder $E(\cdot; \theta_E^*)$, classifier heads $\{H_k\}_{k=1}^N$, Training data $\mathcal{D}_{\text{train}}$, fine-tuning epochs E_{S2} , Task loss functions $\{\mathcal{L}_k\}_{k=1}^N$, number of classes C

Ensure: Fine-tuned classifier parameters $\{\theta_{H_k}^*\}_{k=1}^N$

- 1: Initialize SGD optimizer for $\theta_H = \{\theta_{H_1}, \dots, \theta_{H_N}\}$
- 2: Initialize R-PLA weights $\{\beta_{k,e}\}_{k=1}^N$ to uniform values
- 3: Initialize RD-PCGrad conflict resolution mechanism
- 4: **for** $e = 1$ to E_{S2} **do**
- 5: **for** each batch $(X, Y) \sim \mathcal{D}_{\text{train}}$ **do**
- 6: **// Feature Extraction (Frozen)**
- 7: $\mathbf{Z} \leftarrow E(X; \theta_E^*)$ {Extract fixed features}
- 8: **// Multi-Head Forward Pass**
- 9: $\text{logits_dict} \leftarrow \{\}, \text{losses_dict} \leftarrow \{\}$
- 10: **for** $k = 1$ to N **do**
- 11: $\text{logits_dict}[k] \leftarrow H_k(\mathbf{Z}; \theta_{H_k})$
- 12: $\text{losses_dict}[k] \leftarrow \mathcal{L}_k(\text{logits_dict}[k], Y)$
- 13: **end for**
- 14: **// R-PLA Dynamic Weighting**
- 15: $\{\beta_{k,e}\}_{k=1}^N \leftarrow \text{R-PLA-UPDATE}(\text{logits_dict}, Y, C)$
- 16: **// Apply R-PLA Weights**
- 17: **for** $k = 1$ to N **do**
- 18: $\mathcal{L}'_k \leftarrow \beta_{k,e} \cdot \text{losses_dict}[k]$
- 19: **end for**
- 20: **// RD-PCGrad Conflict Resolution**
- 21: $\mathbf{g}_{\text{final}} \leftarrow \text{RD-PCGRAD}(\{\mathcal{L}'_k\}_{k=1}^N, \theta_H)$
- 22: **// Parameter Update**
- 23: $\theta_H \leftarrow \theta_H - \eta \mathbf{g}_{\text{final}}$
- 24: **end for**
- 25: Apply cosine annealing to learning rate η
- 26: **end for**
- 27: **return** $\{\theta_{H_k}^*\}_{k=1}^N$

509 Algorithm 3 presents the detailed R-PLA mechanism that adaptively weights tasks based on their
510 real-time performance patterns and similarity to a reference performance profile.

Algorithm 3 Refined Performance Level Agreement (R-PLA) Weighting

Require: Logits dictionary $\text{logits_dict} = \{\text{logits}_k\}_{k=1}^N$, True labels Y , number of classes C , Numerical stability constants $\epsilon_{\text{sim}} = 10^{-8}$

Ensure: Updated task weights $\{\beta_{k,e}\}_{k=1}^N$

```
1: // Compute Per-Class Performance for Each Task
2: performance_vectors  $\leftarrow []$  {Initialize empty list}
3: for  $k = 1$  to  $N$  do
4:    $\text{preds}_k \leftarrow \arg \max(\text{logits}_k, \dim = 1)$  {Predictions for task  $k$ }
5:    $\mathbf{a}_{k,e}^{(\text{batch})} \leftarrow \text{zeros}(C)$  {Per-class accuracy vector}
6:   for  $c = 0$  to  $C - 1$  do
7:     class_mask  $\leftarrow (Y == c)$  {Mask for class  $c$ }
8:     total_count  $\leftarrow \text{class\_mask.sum}()$ 
9:     if total_count > 0 then
10:      correct_count  $\leftarrow (\text{preds}_k[\text{class\_mask}] == c).\text{sum}()$ 
11:       $\mathbf{a}_{k,e}^{(\text{batch})}[c] \leftarrow \text{correct\_count}/\text{total\_count}$ 
12:     else
13:       $\mathbf{a}_{k,e}^{(\text{batch})}[c] \leftarrow 0$  {No samples for this class}
14:     end if
15:   end for
16:   performance_vectors.append( $\mathbf{a}_{k,e}^{(\text{batch})}$ )
17: end for
18: // Set Reference Performance (Last Task)
19:  $\mathbf{a}_{\text{ref},e}^{(\text{batch})} \leftarrow \text{performance\_vectors}[-1]$ 
20: // Compute Cosine Similarity and Weights
21: for  $k = 1$  to  $N$  do
22:   numerator  $\leftarrow \mathbf{a}_{k,e}^{(\text{batch})} \cdot \mathbf{a}_{\text{ref},e}^{(\text{batch})}$  {Dot product}
23:   denominator  $\leftarrow \|\mathbf{a}_{k,e}^{(\text{batch})}\|_2 \cdot \|\mathbf{a}_{\text{ref},e}^{(\text{batch})}\|_2$ 
24:    $s_{k,e} \leftarrow \frac{\text{numerator}}{\max(\text{denominator}, \epsilon_{\text{sim}})}$  {Cosine similarity}
25:   // Cube Root Transformation with Non-negative Clipping
26:    $\beta_{k,e} \leftarrow \max(0, \sqrt[3]{s_{k,e}})$  {Robust weighting}
27: end for
28: return  $\{\beta_{k,e}\}_{k=1}^N$ 
```

511 Algorithm 4 details the RD-PCGrad mechanism that resolves gradient conflicts through deterministic
512 projections while maintaining numerical stability and reproducibility.

Algorithm 4 Robust Deterministic Projective Conflict Gradient (RD-PCGrad)

Require: Weighted losses $\{\mathcal{L}'_k\}_{k=1}^N$, parameters θ_H
Require: Stability constants $\epsilon_{\text{norm}} = 10^{-8}$, $\epsilon_{\text{div}} = 10^{-8}$
Ensure: Conflict-resolved final gradient $\mathbf{g}_{\text{final}}$

```
1: // Compute Individual Task Gradients
2: gradients  $\leftarrow []$ , masks  $\leftarrow []$ 
3: for  $k = 1$  to  $N$  do
4:    $\mathbf{g}_k \leftarrow \nabla_{\theta_H} \mathcal{L}'_k$  {Gradient for task  $k$ }
5:    $\mathbf{h}_k \leftarrow \mathbf{1}[\mathbf{g}_k \neq \mathbf{0}]$  {Gradient existence mask}
6:   gradients.append( $\mathbf{g}_k$ ), masks.append( $\mathbf{h}_k$ )
7: end for
8: // Deterministic Iterative Conflict Resolution
9: projected_grads  $\leftarrow \text{copy}(\text{gradients})$  {Deep copy for modification}
10: for  $i = 1$  to  $N$  do
11:   for  $j = i + 1$  to  $N$  do
12:     {Fixed pairwise order for determinism} conflict  $\leftarrow (\mathbf{g}'_i)^T \mathbf{g}'_j < 0$  {Check for conflict}
13:     both_active  $\leftarrow (\mathbf{h}_i \odot \mathbf{h}_j).any()$  {Both gradients contribute} if conflict and both_active
14:     then
15:       // Project gradient  $i$  away from gradient  $j$ 
16:       dot_product  $\leftarrow (\mathbf{g}'_i)^T \mathbf{g}'_j$ 
17:       norm_squared  $\leftarrow \|\mathbf{g}'_j\|_2^2 + \epsilon_{\text{norm}}$ 
18:       projection  $\leftarrow \frac{\text{dot\_product}}{\text{norm\_squared}} \mathbf{g}'_j$ 
19:        $\mathbf{g}'_i \leftarrow \mathbf{g}'_i - \text{projection}$  {Remove conflicting component}
20:     end if
21:   end for
22: end for
23: // Numerically Stable Gradient Aggregation
24: weighted_grads  $\leftarrow []$ 
25: for  $k = 1$  to  $N$  do
26:   weighted_grads.append( $\mathbf{g}'_k \odot \mathbf{h}_k$ ) {Apply existence mask}
27: end for
28: numerator  $\leftarrow \sum_{k=1}^N \text{weighted\_grads}[k]$  {Sum of masked gradients}
29: denominator  $\leftarrow \sum_{k=1}^N \mathbf{h}_k + \epsilon_{\text{div}}$  {Normalization with stability}
30:  $\mathbf{g}_{\text{final}} \leftarrow \frac{\text{numerator}}{\text{denominator}}$  {Element-wise division}
31: return  $\mathbf{g}_{\text{final}}$ 
```

513 7.3 Key Algorithmic Innovations

514 Addressing the complexities of long-tailed recognition, TS-MOF incorporates several key algorithmic
515 innovations. The **Strategic Decoupling** (Algorithm 1) tackles the challenge of simultaneously
516 learning features and balancing classifiers by freezing the Stage 1 encoder (θ_E^*) and applying complex
517 optimization only to the classifier heads. To navigate the conflicting goals of multiple LTR strategies,
518 R-PLA provides **Adaptive Task Weighting** (Algorithm 3) that dynamically highlights strategies
519 contributing most to desired performance patterns, while RD-PCGrad ensures **Deterministic Conflict**
520 **Resolution** (Algorithm 4), stabilizing the multi-objective training process against negative transfer.
521 This principled approach, coupled with **Efficient Implementation** ($O(N^2 D + NC^2)$ per iteration in
522 Stage 2), allows TS-MOF to robustly leverage complementary strengths of diverse LTR strategies,
523 overcoming the limitations of single methods. Furthermore, our implementation simplifies the
524 interface for integrating different strategies, practically reducing the cost of improving long-tailed
525 learning performance using multi-objective optimization and enhancing stability.

526 8 Related Work: Multi-Objective Optimization in Machine Learning

527 Multi-Objective Optimization (MOO) is a field concerned with mathematical optimization problems
528 involving more than one objective function to be minimized or maximized simultaneously. In many
529 real-world scenarios and increasingly in Machine Learning (ML), multiple objectives often conflict,

meaning improving one objective may degrade another. The goal of MOO is typically to find Pareto optimal solutions, where no objective can be improved without worsening at least one other.

In Machine Learning, MOO principles are naturally applied to problems involving competing goals. A prominent area is Multi-Task Learning (MTL) [21, 29], where a single model is trained to perform multiple tasks simultaneously. Optimizing a simple weighted sum of task losses can be suboptimal, especially if tasks have conflicting gradients, leading to negative transfer [25]. This has spurred the development of gradient-based MOO methods designed to find update directions that improve all tasks or, at worst, do not worsen any, moving towards the Pareto front.

Notable gradient-based MOO methods include the Multiple-Gradient Descent Algorithm (MGDA) [8] and Projective Conflict Gradient (PCGrad) [25]. MGDA seeks to find a descent direction that minimizes the maximum directional derivative among all objectives, aiming for a common descent direction within the convex hull of the task gradients. PCGrad, on the other hand, directly addresses conflicting gradients by projecting the gradient of one task onto the normal plane of another if their dot product is negative. This iterative projection process aims to remove conflicting components, resulting in a set of modified gradients whose sum (or average) constitutes a Pareto-improving or non-worsening direction. These methods have demonstrated effectiveness in stabilizing training and improving performance in MTL scenarios.

Our approach differs significantly from existing MOO applications in ML and prior MOO-related work in LTR, driven by a specific, deeper motivation for tackling the LTR problem. While standard MOO methods like PCGrad [25] and MGDA [8] are general-purpose tools for finding Pareto solutions in arbitrary multi-objective problems (like standard MTL), **our TS-MOF framework explicitly employs MOO not as a generic optimizer for abstract tasks, but as a principled mechanism for synergistically fusing diverse, specialized LTR strategies during a targeted fine-tuning stage.** The objectives in our Stage 2 MOO are not just arbitrary task losses; they are losses derived from methods (like LDAM [3], BS [19], KPS [13], BCL [28], etc.) *specifically designed to address different aspects of the LTR imbalance.*

The deep motivation behind using MOO in TS-MOF is to move beyond the limitations of any single LTR strategy and overcome the seesaw dilemma by **finding an optimal combination that leverages the complementary strengths of multiple strategies.** R-PLA uses performance-based adaptive weighting to dynamically prioritize strategies based on their real-time contribution to the desired performance pattern, while RD-PCGrad provides a robust and deterministic way to reconcile the potentially conflicting gradient signals arising from simultaneously optimizing towards these different LTR-specific goals. We apply this advanced MOO specifically to the classifier heads after decoupling feature learning, ensuring the optimization focuses effectively on classification balance without corrupting the feature representation. Thus, our MOO is not just a mathematical technique applied; it is the core engine enabling the principled, robust, and effective fusion of heterogeneous LTR knowledge to achieve superior and balanced recognition performance.

9 More Empirical Results

This appendix provides additional empirical results and analyses to further demonstrate the effectiveness and underlying mechanisms of the TS-MOF framework.

9.1 Results on CIFAR-100-LT with Various Strategy Combinations

We also evaluated the evolutionary results of various combinations of strategies in the second stage of TS-MOF. Table 3 shows the performance across different Imbalance Ratios (IR=10, 50, 100) on the CIFAR-100-LT dataset when using TS-MOF with different sets of constituent LTR strategies. TS-MOF achieved excellent results in the combination of the KPS + BCL + LOS strategy, as highlighted in the table, demonstrating its ability to bring obvious general improvements across class groups and imbalance settings.

9.2 T-SNE Analysis

Figure 4 shows the t-SNE analysis of the feature representations from different models. We compare the representations learned by models using BCL or KPS independently, their simple weighted fusion

Table 3: Accuracy (%) on CIFAR-100-LT dataset with TS-MOF using various strategy combinations in Stage 2 fine-tuning. Results are reported for Head, Medium, Tail classes, and overall (All) accuracy across different Imbalance Ratios (IR).

Method	IR=10				IR=50				IR=100			
	Head	Medium	Tail	All	Head	Medium	Tail	All	Head	Medium	Tail	All
TS-MOF(CE+LDAM-DRW+LOS)	74.90	59.71	–	70.19	75.95	51.05	42.61	59.74	78.23	51.34	35.20	55.91
TS-MOF(CE+KPS+LOS)	74.22	62.81	–	70.68	76.22	51.63	42.06	59.99	78.46	49.14	39.57	56.53
TS-MOF(CE+BCL+LOS)	75.04	58.26	–	69.84	76.07	50.49	42.11	59.47	78.94	49.84	34.87	55.57
TS-MOF(CE+CE-DRW+LOS)	74.32	59.71	–	69.79	75.88	50.54	42.33	59.45	78.77	49.94	34.63	55.44
TS-MOF(CE+BS+LOS)	74.49	59.55	–	69.86	75.76	51.61	42.17	59.81	78.80	51.83	35.10	56.25
TS-MOF(BS+KPS+LOS)	73.99	61.42	–	70.09	75.61	53.17	42.00	60.36	77.49	51.54	38.27	56.64
TS-MOF(BS+BCL+LOS)	74.25	60.06	–	69.85	76.39	51.20	42.17	59.90	78.89	51.49	34.70	56.04
TS-MOF(BS+CE-DRW+LOS)	74.33	60.00	–	69.89	76.05	51.37	42.06	59.81	79.46	51.43	34.63	56.20
TS-MOF(BS+LDAM-DRW+LOS)	74.19	59.58	–	69.66	74.90	51.95	42.33	59.63	78.20	51.23	35.17	55.85
TS-MOF(KPS+BCL+LOS)	74.75	62.03	–	70.81	75.24	50.76	47.56	60.22	79.00	49.29	39.97	56.89
TS-MOF(KPS+CE-DRW+LOS)	74.52	61.68	–	70.54	75.41	51.12	45.22	60.02	78.37	50.26	38.17	56.47
TS-MOF(KPS+LDAM-DRW+LOS)	73.77	61.52	–	69.97	73.02	51.41	46.89	59.46	75.06	49.60	40.87	55.89
TS-MOF(SHIKE+BS+LOS)	74.55	59.71	–	69.95	76.22	51.39	42.11	59.90	79.03	51.77	34.77	56.21
TS-MOF(SHIKE+BCL+LOS)	74.67	59.45	–	69.95	76.15	50.73	42.28	59.63	79.17	49.66	34.73	55.51
TS-MOF(SHIKE+CE-DRW+LOS)	74.57	59.45	–	69.88	76.27	49.95	42.11	59.33	78.91	49.37	35.20	55.46
TS-MOF(SHIKE+LDAM-DRW+LOS)	74.99	59.35	–	70.14	76.15	51.15	42.22	59.79	78.40	50.74	34.73	55.62

(BCL+KPS), and our proposed fusion method TS-MOF. Compared with before fusion, the TS-MOF method shows a significantly increased separability of clusters in almost all categories in the feature space, indicating that our multi-objective fusion approach learns a more discriminative representation despite freezing the backbone in Stage 2. This suggests the classifier heads adapt in a way that better separates classes in the existing feature space.

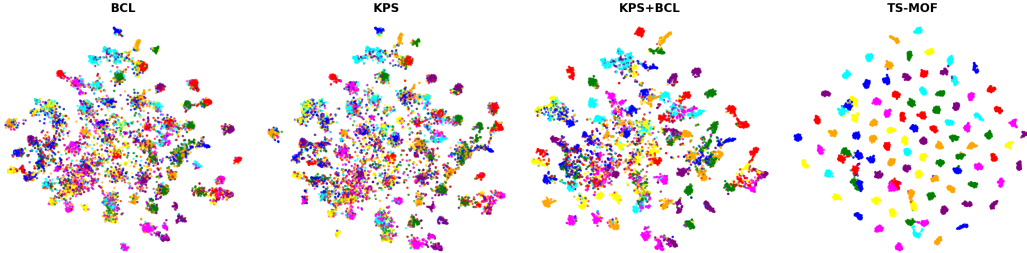


Figure 4: T-SNE comparative analysis of feature representations. We compare models trained on CIFAR-100-LT using BCL independently, KPS independently, a simple weighted fusion of BCL+KPS, and our proposed TS-MOF fusion method. Each point represents a feature vector from the test set, colored by its true class. Better separation indicates a more discriminative feature space for classification.

9.3 Confusion Matrix Analysis

Figure 5 shows a comparison of the predictive performance between individual strategies, simple fusion, and our TS-MOF method through confusion matrices. The confusion matrices are normalized by row (true label count) to show per-class accuracy patterns. From the figure, it can be seen that KPS pays more attention to some tail classes (higher diagonal values for later classes), while BCL focuses more on the head classes (higher diagonal values for earlier classes). Simple fusion may not effectively resolve conflicts and can potentially damage the performance of tail classes. Our method TS-MOF effectively combines the advantages of individual strategies and exhibits good performance across head, medium, and tail classes, as indicated by the higher and more uniform diagonal values compared to baselines and simple fusion.

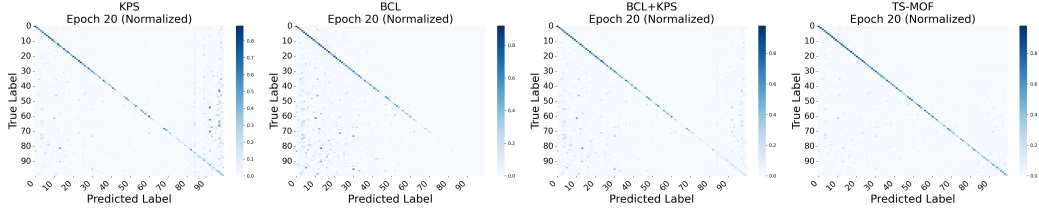


Figure 5: Comparative analysis of confusion matrices on CIFAR-100-LT (IR=100). We compare the predictive performance of models using BCL independently, KPS independently, a simple weighted fusion of BCL+KPS, and our proposed TS-MOF fusion method. Rows represent true labels, columns represent predicted labels. Diagonal elements indicate correct classifications (accuracy per class).

10 Limitations

Despite its advancements, TS-MOF has certain limitations. Its performance is inherently tied to the quality of the Stage 1 pre-trained features, as the encoder is frozen during fine-tuning. The framework also requires pre-selecting the specific set of LTR strategies to be included in the multi-objective optimization, which might require empirical tuning for optimal performance on new datasets. Furthermore, while designed for robustness, the framework still involves several hyperparameters that may need careful configuration.

11 Broader Impacts

TS-MOF aims to improve balanced recognition in long-tailed datasets, which are common in real-world applications. By enhancing tail class performance, our method can contribute to greater fairness and equity in ML systems by providing better representation for under-represented categories, potentially benefiting applications in healthcare, social sciences, and specialized domains. Making better use of rare data can also increase the utility of ML in resource-constrained settings. However, like any advanced recognition technology, there is a potential for misuse, such as in surveillance applications, emphasizing the need for responsible development and deployment.