U-MOF: Uncertainty-Guided Parameter-Efficient Multi-Objective Fine-Tuning for Long-Tailed Recognition

Di Wu[1,†] **Yuan Dong**[1,†] **Zhe Zhao**[1] **Liheng Yu**[1]
Xiaofeng Cao[3] **Pengkun Wang**[1,2,*] **Yang Wang**[1,2,*]
[1]University of Science and Technology of China
[2]Suzhou Institute for Advanced Research
[3]Tongji University
[†]Di Wu and Yuan Dong contributed equally and share first authorship.
[*]Corresponding authors

Abstract

Recent long-tailed recognition methods increasingly adopt two-stage fine-tuning, where a generic representation is first learned and imbalance-aware adaptation is then performed in a lightweight second stage. However, existing second-stage designs often allocate objective emphasis using delayed empirical summaries, such as historical class-wise performance or validation feedback, and commonly restrict adaptation to classifier heads. These choices may make routing unreliable when empirical feedback is noisy or skewed, and may also limit the feature plasticity required for rare categories. In this paper, we propose U-MOF, an uncertainty-guided parameter-efficient multi-objective fine-tuning framework for long-tailed recognition. U-MOF introduces lightweight residual adapters into a shared-backbone expert committee, uses entropy-based committee statistics as diagnostic routing proxies to modulate tail-oriented and smoothing-oriented objectives, and applies conflict-aware orthogonal projection to coordinate heterogeneous objective gradients. Experiments on CIFAR-100-LT, ImageNet-LT, and iNaturalist 2018 show that U-MOF improves rare-category and overall performance in controlled stage-two comparisons and remains competitive with strong published long-tailed recognition baselines, while preserving the efficiency advantages of decoupled adaptation. These results indicate that reliable stage-two long-tailed fine-tuning benefits not only from selecting appropriate objectives, but also from diagnosing when each objective should dominate and from retaining limited feature-level plasticity. An anonymized implementation is included in the **Supplementary Material**.

1 Introduction

Long-tailed recognition remains a central challenge in visual learning. Real-world visual data are rarely distributed uniformly across classes. Instead, a small number of head categories often dominate the sample space, while many tail categories are supported by only limited observations. This imbalance causes standard empirical risk minimization to allocate disproportionate representational capacity to majority classes, yielding strong head-category accuracy but insufficient discrimination for rare categories. A large body of work has therefore explored re-sampling, re-weighting, decoupled training, tailored objectives, and multi-expert architectures to alleviate head dominance and improve tail recognition [1–11]. Despite substantial progress, the tension between preserving head-category performance and recovering tail-category discrimination remains difficult to resolve within a single training objective.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

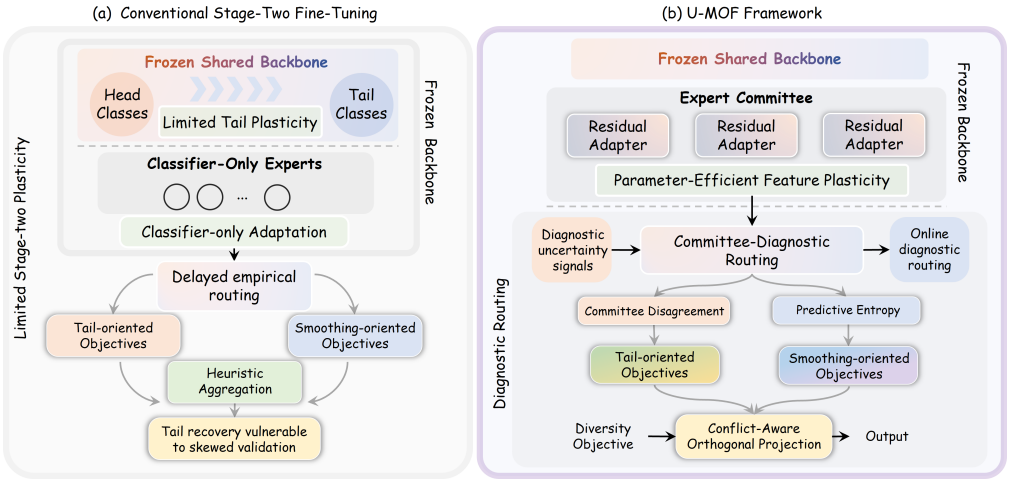


Figure 1: Conceptual comparison between conventional stage-two fine-tuning and the proposed U-MOF framework. (a) Conventional stage-two designs often combine classifier-only adaptation with routing signals derived from delayed empirical feedback. Such feedback may become unreliable when validation statistics or historical estimates are noisy or skewed. (b) U-MOF preserves the shared frozen-backbone setting, but introduces residual adapters for parameter-efficient feature plasticity, committee-derived diagnostics for objective routing, and conflict-aware orthogonal projection for coordinating classification and diversity objectives.

A promising direction is to view long-tailed learning as a coordination problem among multiple complementary objectives. Objectives that emphasize tail mining, hard-sample learning, class balancing, or prediction smoothing often improve different regions of the label distribution, but they may also interfere when optimized jointly. Recent multi-objective formulations address this issue by coordinating heterogeneous objectives within a unified optimization process [12–16]. Among these strategies, two-stage fine-tuning has emerged as a practical paradigm for long-tailed recognition [2, 17–22]. It first learns a generic representation and then performs imbalance-aware adaptation in a lightweight second stage. This decoupling reduces the instability and computational burden of end-to-end multi-objective optimization, making multi-strategy collaboration easier to deploy.

However, we argue that *current stage-two designs still rely on two limiting assumptions* that have not been sufficiently examined. The first assumption is that stage-two routing can be reliably driven by delayed empirical summaries. Many second-stage methods determine objective emphasis from signals such as historical class-wise accuracy, validation-driven feedback, or accumulated loss statistics [17, 4, 23, 24]. These signals are useful indicators of where the model currently performs poorly, but they provide limited information about why a class or an instance is difficult. In particular, they may conflate model-side disagreement with prediction ambiguity, and their reliability can degrade when validation data are skewed or when class-wise estimates are noisy. In long-tailed settings, this issue directly affects how optimization capacity is allocated across objectives.

The second assumption is that classifier-only adaptation is sufficient once a generic backbone has been pretrained. Freezing the shared encoder is attractive because it preserves representational stability and enables efficient second-stage training. Nevertheless, restricting adaptation entirely to the classifier head can create a feature-level bottleneck for rare categories. When tail recovery requires even modest feature refinement, classifier-level updates alone may be insufficient to form discriminative boundaries for underrepresented classes. In other words, the efficiency of frozen-backbone fine-tuning is often obtained by implicitly assuming that stage two no longer needs feature-level movement. We find this assumption overly restrictive for long-tailed recognition, where the central challenge is precisely to redistribute limited adaptive capacity toward data-scarce categories.

These observations suggest that stage-two fine-tuning should be strengthened from two complementary perspectives: routing quality and feature plasticity. Rather than viewing the second stage as a lightweight controller that merely reweights predefined objectives on top of fixed representations, we treat it as a targeted adaptation process. A practical stage-two optimizer should satisfy two

requirements. First, its routing signal should remain informative even when delayed empirical feedback is noisy, skewed, or incomplete. Second, it should retain enough representational flexibility to improve tail discrimination while preserving the efficiency and stability that make two-stage fine-tuning attractive.

Motivated by this view, we propose **U-MOF**, an uncertainty-guided parameter-efficient multi-objective fine-tuning framework for long-tailed recognition. U-MOF comprises three coordinated components. First, we introduce lightweight residual adapters into a shared-backbone expert committee. These adapters restore limited yet effective feature-level plasticity without fully unfreezing the pretrained backbone, thereby expanding the second-stage adaptive space beyond classifier parameters while maintaining a small computational footprint. Second, we replace delayed empirical routing with online diagnostic routing derived from committee behavior. Specifically, entropy-based committee statistics are used as operational routing proxies rather than calibrated Bayesian uncertainty estimates, allowing the optimizer to modulate the emphasis between tail-oriented and smoothing-oriented objectives. We therefore do not aim to solve uncertainty calibration; instead, we use committee statistics as lightweight online diagnostics for stage-two objective allocation. Third, because dynamic weighting alone cannot remove destructive interference in parameter space, we apply conflict-aware orthogonal projection to coordinate heterogeneous objective gradients during stage-two adaptation.

U-MOF studies role-structured coordination under restricted stage-two adaptation, not new individual adapters, uncertainty estimators, or generic multi-objective solvers. The residual paths expose a limited feature-update space, committee diagnostics allocate objective emphasis, and classification-first projection controls auxiliary diversity pressure. Adapters are not intrinsically tail-selective; their frequency-dependent benefit comes from the objectives and coordinated updates. Introducing them after backbone freezing preserves the original representation-learning stage and separates generic learning from imbalance-aware adaptation (Appendix E.12).

Experiments on CIFAR-100-LT [25, 1], ImageNet-LT [26], and iNaturalist 2018 [27] show that U-MOF improves rare-category and overall performance in controlled stage-two comparisons and remains competitive with strong published long-tailed recognition baselines. Additional analyses further show that the proposed design preserves the practical efficiency of decoupled adaptation, remains stable under skewed validation protocols, and is supported by strict component-removal and routing-negative-control comparisons. These results suggest that the effectiveness of stage-two long-tailed fine-tuning depends not only on the selected objective family, but also on how routing signals are defined and how much feature-level plasticity is retained during adaptation.

Our main contributions are summarized as follows:

❶ **Problem diagnosis.** We identify two under-examined limitations of existing stage-two long-tailed fine-tuning: the reliance on delayed empirical routing signals and the representational bottleneck induced by classifier-only adaptation.

❷ **Framework design.** We propose **U-MOF**, a parameter-efficient stage-two optimization framework that couples residual expert adapters, entropy-based committee diagnostics, and conflict-aware orthogonal projection to coordinate heterogeneous long-tailed objectives under a shared frozen-backbone setting.

❸ **Experimental evidence.** We evaluate **U-MOF** on three standard long-tailed benchmarks and controlled stage-two comparisons, showing improved rare-category and overall performance while preserving the efficiency advantages of decoupled adaptation.

2 Related Work

Long-Tailed Recognition and Two-Stage Fine-Tuning.

Long-tailed recognition has been studied through re-sampling, loss re-balancing, classifier adjustment, decoupled training, representation refinement, and multi-expert learning [1–5, 7–10, 28]. Recent work further explores prototype-anchored manifold completion to recover underrepresented tail representations [29]. Decoupled and two-stage methods first learn a generic representation and then perform imbalance-aware adaptation [2, 17–22]. Related research on heterogeneous learning investigates cross-silo feature alignment for imbalanced federated data [30], non-interfering attention masking for federated prompt refinement [31], and geometry-aware exemplar replay scheduling under continual dynamic heterogeneity [32]. Although these studies address different learning settings,

they highlight complementary challenges involving distribution heterogeneity and representational interference. U-MOF follows the decoupled paradigm but strengthens the second stage itself, replacing classifier-only adaptation and delayed empirical feedback with lightweight feature plasticity and diagnostic objective routing.

Diagnostic Uncertainty Proxies for Adaptive Optimization. Predictive uncertainty has been used to characterize confidence, ambiguity, and reliability [33–35]. In long-tailed recognition, difficult samples are heterogeneous: some reflect unstable tail decisions, while others reflect ambiguous or noisy evidence. Instead of routing objectives with delayed accuracy, loss, or validation summaries, U-MOF uses entropy-based committee statistics as online diagnostic proxies. Because all experts share the frozen backbone and differ only through lightweight adapters and heads, these statistics are not calibrated Bayesian uncertainty estimates; they are internal optimization diagnostics for modulating tail-oriented and smoothing-oriented objectives.

Multi-Objective Optimization and Gradient Coordination. Multi-objective optimization and multi-task learning show that naively aggregating heterogeneous objectives can produce unstable trade-offs when gradients conflict [12–16]. This issue is particularly relevant to long-tailed recognition, where objectives for tail recovery, head preservation, smoothing, and expert diversity may prefer different descent directions. Recent long-tailed methods have begun to incorporate conflict-aware or routing-based optimization into second-stage adaptation [23, 24, 17]. U-MOF is related to these methods but differs in focus: rather than proposing a generic multi-objective solver, it integrates diagnostic routing, lightweight feature adaptation, and classification-first projection into a practical stage-two optimization framework. A more detailed comparison is provided in Appendix A.

3 Method

3.1 Stage-Two Formulation

Consider a long-tailed classification dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathcal{X}$, $y_i \in \{1, \dots, C\}$, and class frequencies satisfy $n_1 \geq n_2 \geq \cdots \geq n_C$. U-MOF follows a two-stage protocol. In stage one, a generic backbone f_θ is trained on the original long-tailed training set. In stage two, the backbone is frozen and only lightweight expert-specific parameters are updated. This preserves the efficiency and stability of decoupled training while allowing the second stage to focus on imbalance-aware adaptation with limited yet targeted representational flexibility.

We view stage two not as a fixed *post hoc* classifier adjustment, but as a targeted optimization process that must decide how to route heterogeneous objectives and how much feature plasticity to retain. U-MOF addresses these questions with a lightweight expert committee whose adapters provide tail-sensitive adaptation paths and whose cross-expert behavior supplies online routing diagnostics. U-MOF does not redefine existing long-tailed objectives; it assigns their relative emphasis and coordinates their gradients under a shared frozen-backbone setting.

Given an input x, the frozen backbone produces a shared feature map

$$h = f_\theta(x). \tag{1}$$

U-MOF builds a committee with E experts. Each expert contains a residual adapter a_{ϕ_e} and a classifier head h_{ψ_e}. The trainable stage-two parameter set is

$$\Omega = \{(\phi_e, \psi_e)\}_{e=1}^{E}, \tag{2}$$

where ϕ_e and ψ_e denote the adapter and classifier-head parameters of expert e.

Rather than optimizing a single scalar loss, U-MOF operates on a family of stage-two objectives:

$$\mathcal{O} = \mathcal{O}^{\text{mine}} \cup \mathcal{O}^{\text{smooth}} \cup \mathcal{O}^{\text{div}}. \tag{3}$$

The mining group promotes tail recovery and hard-example adaptation, the smoothing group stabilizes uncertain predictions, and the diversity group prevents expert collapse. In the main experiments, we instantiate this family as

$$\mathcal{O}^{\text{mine}} = \{\mathcal{L}_{\text{KPS}}, \mathcal{L}_{\text{BCL}}\}, \quad \mathcal{O}^{\text{smooth}} = \{\mathcal{L}_{\text{LOS}}\}, \quad \mathcal{O}^{\text{div}} = \{\ell_{\text{feat}}^{\text{div}}, \ell_{\text{pred}}^{\text{div}}\}. \tag{4}$$

KPS and BCL serve as representative tail-oriented objectives, LOS serves as the smoothing objective, and the two diversity terms are defined in Appendix C.3. This assembly provides a fixed objective

pool for controlled comparisons. Unless otherwise specified, controlled comparisons use this same objective assembly, while robustness to alternative combinations is studied in Appendix E.1.

For a mini-batch $\mathcal{B}_t$, let $\ell_k^t(\Omega)$ denote the k-th objective and $g_k^t = \nabla_\Omega \ell_k^t$ its gradient over the trainable stage-two parameters. U-MOF constructs the final update direction from $\{g_k^t\}_{k\in\mathcal{O}}$ through three sequential components: residual expert adaptation, diagnostic objective routing, and conflict-aware gradient coordination before applying the parameter update.

3.2 Parameter-Efficient Expert Adaptation

A primary limitation of classifier-only stage-two adaptation is that it assumes the frozen feature space already contains sufficient discrimination for tail categories. This assumption can be restrictive when rare classes require modest feature refinement rather than only adjusted decision boundaries. Fully unfreezing the backbone would provide more plasticity, but it increases computation and can disrupt the representation learned in stage one. U-MOF addresses this trade-off by attaching lightweight residual adapters to the frozen feature map. The goal is not to relearn the backbone, but to provide a small expert-specific residual adjustment that can recover part of the feature plasticity lost by classifier-only fine-tuning.

For expert e, we first compute a pooled descriptor from the shared feature map:

$$v = \mathrm{GAP}(h). \tag{5}$$

A bottleneck MLP then produces a channel-wise gate:

$$g_e = \mathrm{Sigmoid}\left(W_e^{\mathrm{up}}\sigma\left(W_e^{\mathrm{down}}v\right)\right), \tag{6}$$

where W_e^{down} and W_e^{up} are the down-projection and up-projection matrices, and the bottleneck ratio is controlled by r. To avoid uncontrolled feature rescaling, we use a centered residual modulation:

$$h_e = h + \alpha_e h \odot \left(g_e - \frac{1}{2}\right), \tag{7}$$

where $\odot$ denotes channel-wise multiplication with broadcasting over spatial dimensions, and α_e is a residual scaling coefficient. This centered form is important because a direct residual form $h + h \odot g_e$ would rescale the frozen feature even when the gate is initialized around $1/2$. In practice, α_e is fixed to a small positive value or initialized to a small positive value when learnable. This keeps the adapter close to identity at the beginning of stage two while allowing gradients to flow through the residual branch, enabling stable but nonzero feature adaptation.

The adapted feature h_e is then pooled and passed to the expert classifier head:

$$\tilde{z}_e = \mathrm{GAP}(h_e), \qquad p_e(\cdot \mid x) = \mathrm{softmax}(h_{\psi_e}(\tilde{z}_e)). \tag{8}$$

Thus, each expert can perform limited feature-level modulation before classification, while the backbone remains shared and frozen. This makes U-MOF more expressive than classifier-only adaptation but substantially lighter than full multi-expert backbone replication. Detailed adapter parameterization, initialization, and parameter counts are provided in Appendix B.2 and Appendix F.2.

3.3 Diagnostic Committee-Guided Objective Routing

The second component of U-MOF addresses the routing problem in stage-two optimization. Existing methods often allocate objective emphasis using delayed empirical summaries, such as validation feedback, historical class-wise performance, or accumulated losses. These signals can indicate that the model is struggling, but they provide limited information about the source of difficulty behind the observed optimization difficulty. U-MOF instead uses online committee behavior as a diagnostic routing signal. This routing is computed from the current mini-batch and therefore avoids introducing an additional validation-driven objective allocation loop.

For an input x, the E experts produce predictive distributions $\{p_e(\cdot \mid x)\}_{e=1}^{E}$. We first compute the committee-averaged prediction:

$$\bar{p}(\cdot \mid x) = \frac{1}{E}\sum_{e=1}^{E} p_e(\cdot \mid x). \tag{9}$$

We then define two entropy-based committee statistics:

$$U^{\mathrm{ent}}(x) = \frac{1}{E}\sum_{e=1}^{E} H(p_e(\cdot \mid x)), \qquad U^{\mathrm{dis}}(x) = H(\bar{p}(\cdot \mid x)) - U^{\mathrm{ent}}(x). \tag{10}$$

Here, U^{ent} measures within-expert diffuseness and U^{dis} cross-expert disagreement. **Uncertainty-guided Proxy Loss Allocation (U-PLA)** uses them to adjust the relative emphasis on smoothing and mining, respectively; it is an allocation rule, not an additional loss. High entropy can also occur on hard or rare examples, so the assignment is not exclusive: all objective groups retain positive weights. The statistics are operational proxies, not calibrated Bayesian uncertainties; matched routing controls and a one-step intervention audit support their use (Appendices E.7–E.8).

For mini-batch $\mathcal{B}_t$, we compute normalized batch diagnostics:

$$\tilde{U}_t^{\mathrm{ent}} = \frac{1}{|\mathcal{B}_t| \log C}\sum_{x\in\mathcal{B}_t} U^{\mathrm{ent}}(x), \qquad \tilde{U}_t^{\mathrm{dis}} = \frac{1}{|\mathcal{B}_t| \log E}\sum_{x\in\mathcal{B}_t} U^{\mathrm{dis}}(x). \tag{11}$$

The normalization makes the routing scale more comparable across datasets with different numbers of classes and experts. U-PLA then maps these diagnostics to group-level scores:

$$\rho_t^{\mathrm{mine}} = \exp(\beta_m \tilde{U}_t^{\mathrm{dis}}), \quad \rho_t^{\mathrm{smooth}} = \exp(\beta_s \tilde{U}_t^{\mathrm{ent}}), \quad \rho_t^{\mathrm{div}} = \exp(\beta_d), \tag{12}$$

where β_m, β_s, and β_d control the sensitivity of the mining, smoothing, and diversity groups.

For objective $k \in \mathcal{O}^q$, the final routing coefficient is

$$\lambda_k^t = \frac{\rho_t^q}{\sum_{q'\in\{\mathrm{mine,smooth,div}\}} \rho_t^{q'}} \cdot \frac{\pi_k}{\sum_{j\in\mathcal{O}^q} \pi_j}, \tag{13}$$

where π_k is a fixed intra-group prior. The routed gradient is

$$\hat{g}_k^t = \lambda_k^t g_k^t. \tag{14}$$

By using online committee diagnostics, U-PLA avoids relying solely on delayed validation summaries and provides a lightweight mechanism for adapting objective emphasis during stage-two training. It determines how much each objective group should contribute, but not whether their gradient directions are compatible, while the next component determines how the routed gradients should be coordinated in parameter space. Additional routing details, including normalization properties and intra-group priors, are provided in Appendix C.2. Sensitivity analyses are reported in Appendix F.3. Appendix E.7 further includes routing controls, including a reverse diagnostic assignment that swaps disagreement and entropy, to test whether the proposed mapping is meaningful rather than arbitrary.

3.4 Conflict-Aware Stage-Two Optimization

Routing controls the scalar weight of each objective, but it does not remove destructive interference in parameter space. This issue is important in U-MOF because classification-related objectives and diversity-promoting regularizers play different roles. The former directly optimizes recognition performance, whereas the latter encourages expert complementarity. If directly summed, diversity gradients may partially oppose the classification descent direction.

To reduce such interference, U-MOF uses a conflict-aware projection primitive. Given two gradients u and v, we remove from u the component that conflicts with v when their inner product is negative:

$$\Pi(u; v) = u - \mathbf{1}[\langle u, v\rangle < 0]\, \frac{\langle u, v\rangle}{\|v\|_2^2 + \varepsilon} v. \tag{15}$$

When u and v are compatible, the operator leaves u unchanged; otherwise, it removes the conflicting component along v. Therefore, the projection changes only those gradient components that would directly oppose the reference direction.

Let

$$\mathcal{O}^{\mathrm{cls}} = \mathcal{O}^{\mathrm{mine}} \cup \mathcal{O}^{\mathrm{smooth}} \tag{16}$$

denote the classification-related objectives. U-MOF first coordinates their routed gradients to obtain an aggregate classification direction d^t_{cls}. Diversity gradients are then projected against this direction:

$$d_t = d^t_{\mathrm{cls}} + \sum_{k \in \mathcal{O}^{\mathrm{div}}} \Pi(\hat{g}^t_k; d^t_{\mathrm{cls}}), \qquad \Omega_{t+1} = \Omega_t - \eta d_t. \tag{17}$$

This classification-first ordering treats diversity as auxiliary pressure relative to the aggregate classification direction. The stabilized projection attenuates conflicting components; it does not guarantee improvement of every classification objective or of held-out accuracy. Appendix C.4 specifies the recursion, ordering, and numerical interpretation of ε, while Appendix C.3 defines the diversity terms. Comparisons with alternative solvers are provided in Appendix E.7.

Training and inference. During stage two, U-MOF computes committee diagnostics, routes objective gradients with U-PLA, and applies conflict-aware projection to update only Ω. These operations are training-time mechanisms for constructing the stage-two update direction. At inference time, routing and projection are not used. The final prediction is obtained by averaging the expert predictions:

$$p_{\text{U-MOF}}(\cdot \mid x) = \frac{1}{E} \sum_{e=1}^{E} p_e(\cdot \mid x). \tag{18}$$

Thus, U-MOF requires one frozen-backbone forward pass followed by lightweight adapter-head evaluations. Full pseudocode and complexity analysis are provided in Appendix D and Appendix D.3.

4 Experiments

We evaluate **U-MOF** on CIFAR-100-LT, ImageNet-LT, and iNaturalist 2018 through broad benchmark comparisons, controlled stage-two comparisons with a fixed scaffold, and diagnostic analyses of attribution, routing validity, and efficiency. Additional details are provided in Appendix E and Appendix F.

4.1 Experimental Setup

We use ResNet-34 for CIFAR-100-LT and ResNet-50 for ImageNet-LT and iNaturalist 2018. CIFAR-100-LT is evaluated under imbalance ratios IR $= 10$, 50, and 100. For all datasets, the backbone is trained in stage one and frozen in stage two; U-MOF updates only the residual adapters and classifier heads. We report overall top-1 accuracy and the standard *Head*, *Medium*, and *Tail* breakdown [2]. Unless otherwise specified, U-MOF and its controlled variants are reported as mean $\pm$ standard deviation over five independent runs, while baseline results without uncertainty ranges are quoted from the corresponding papers or evaluated once under the matched protocol. Benchmark tables serve as broad references; controlled attribution is provided in Sections 4.3–4.4 and in Appendices E.5, E.7, E.6, and F.1.

4.2 Broad Benchmark Comparison

Table 1 compares U-MOF with representative long-tailed recognition methods on CIFAR-100-LT. Because several entries are quoted from published results, this table is intended as a broad benchmark reference rather than a fully controlled attribution study. Under this setting, U-MOF achieves competitive overall performance and strong Medium/Tail accuracy under high imbalance.

Table 2 further shows competitive overall and Tail accuracy on ImageNet-LT and iNaturalist 2018, suggesting that U-MOF scales beyond CIFAR-style synthetic imbalance. Controlled evidence for the stage-two optimization engine is provided next.

4.3 Controlled Stage-Two Engine Comparison

This classification-first ordering treats diversity as auxiliary pressure relative to the aggregate classification direction. The stabilized projection attenuates conflicting components; it does not guarantee improvement of every classification objective or of held-out accuracy. Appendix C.4 specifies the recursion, ordering, and numerical interpretation of ε, while Appendix C.3 defines the diversity terms. Comparisons with alternative solvers are provided in Appendix E.7.

Table 1: Top-1 accuracy (%) on CIFAR-100-LT under three imbalance ratios. This table provides a broad comparison with long-tailed recognition methods. Under the reported setting, **U-MOF** achieves competitive overall performance and strong Medium/Tail accuracy under high imbalance.

Method	IR=10				IR=50				IR=100			
	Head	Medium	Tail	All	Head	Medium	Tail	All	Head	Medium	Tail	All
CE	63.2	40.3	–	56.5	63.9	36.2	15.2	43.8	65.6	36.2	8.2	38.1
CE-DRW [1]	62.5	48.6	–	58.2	60.6	39.0	22.9	45.0	63.4	41.2	15.7	41.4
LDAM-DRW [1]	62.7	46.1	–	57.5	63.0	41.2	25.1	47.2	62.8	42.6	21.1	43.2
Balanced Softmax [4]	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) [5]	66.4	49.4	–	61.1	65.7	47.7	31.8	52.2	65.7	48.6	25.0	47.5
BCL [36]	62.2	51.8	–	58.9	61.6	43.1	34.3	49.1	63.1	42.9	23.9	44.2
KPS [37]	61.7	58.7	–	59.5	51.6	49.5	52.4	50.5	41.9	39.5	48.7	42.2
SHIKE [38]	66.0	45.0	–	59.0	67.0	43.0	23.0	49.5	66.0	39.0	12.0	46.9
LOS [39]	71.9	62.3	–	69.0	72.4	51.4	40.2	58.0	70.3	52.3	36.6	53.9
TS-MOF [17]	74.7	62.0	–	70.8	75.2	50.7	47.5	60.2	79.0	49.2	39.9	56.8
U-MOF	73.2 ± 0.3	**66.6** ± 0.4	–	**71.1** ± 0.2	74.9 ± 0.2	**52.2** ± 0.3	**49.9** ± 0.4	**61.1** ± 0.2	76.8 ± 0.3	**51.4** ± 0.3	**41.9** ± 0.4	**57.5** ± 0.2

Table 2: Top-1 accuracy (%) on ImageNet-LT and iNaturalist 2018. This table provides a broad reference comparison on larger-scale long-tailed benchmarks. **U-MOF** achieves competitive overall performance and strong Tail accuracy under the reported setting.

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 [1]	61.7	47.3	28.8	50.1	68.2	67.3	66.4	67.0
cRT [2]	58.8	44.0	26.1	47.3	69.0	66.0	63.2	65.2
LDAM-DRW [1]	60.4	46.9	30.7	49.8	–	–	–	66.1
BS [4]	60.9	48.8	32.1	51.0	65.7	67.4	67.5	67.3
KPS [37]	59.7	49.2	35.9	52.3	68.1	69.5	70.2	69.6
RIDE (3 experts) [5]	64.9	50.4	34.4	53.6	69.5	71.0	70.4	70.6
BCL [36]	65.3	53.5	36.3	55.6	69.5	70.9	71.3	70.9
LOS [39]	63.2	50.7	42.3	54.4	69.2	70.7	71.3	70.8
TS-MOF [17]	65.9	52.5	42.6	56.3	72.8	73.6	70.5	72.3
U-MOF	**67.6** ± 0.2	51.3 ± 0.3	**51.5** ± 0.3	**57.6** ± 0.1	**76.4** ± 0.2	70.8 ± 0.2	**75.1** ± 0.3	**73.1** ± 0.1

4.4 Component Ablation

Table 4 shows a cumulative construction on ImageNet-LT. Residual feature adaptation improves Tail accuracy, diversity with projection further improves Tail and Overall performance, and adding U-PLA gives the strongest result. Since this table is cumulative, Appendix E.5 reports strict one-module removal controls.

Figure 2 visualizes the optimization behavior of U-MOF. Projection reduces gradient conflicts during stage-two training, while routing assigns different objective emphasis across frequency regimes. This supports the intended role of U-PLA and conflict-aware projection. Appendix E.7 further tests the diagnostic mapping through alternative routing policies and a reverse U-PLA negative control.

Additional routing and solver comparisons. We further compare U-MOF with alternative routing policies and multi-objective solvers under the same lightweight stage-two footprint. These results, reported in Appendix E.7, show that U-PLA improves over confidence scaling, pure entropy scaling, empirical PLA, and reverse U-PLA, while the proposed classification-first projection outperforms generic solvers such as MGDA, GradNorm, and base PCGrad. The reverse U-PLA control is particularly important because it swaps the diagnostic assignment between disagreement and entropy, testing whether the proposed routing interpretation is meaningful rather than arbitrary.

4.5 Computational Efficiency and Additional Analysis

U-MOF improves stage-two adaptation without replicating the backbone: Table 5 uses one frozen-backbone forward and lightweight adapter-head branches. Routing and projection are training-only; at inference, U-MOF averages expert adapter-head predictions. Detailed parameter, MAC, wall-clock, and memory measurements are provided in Appendix D.3 and Appendix F.1.

Additional analyses are provided in Appendices E and F. These include matched closest-baseline comparisons, strict one-factor ablations, reverse-routing and alternative-solver comparisons, inference-rule

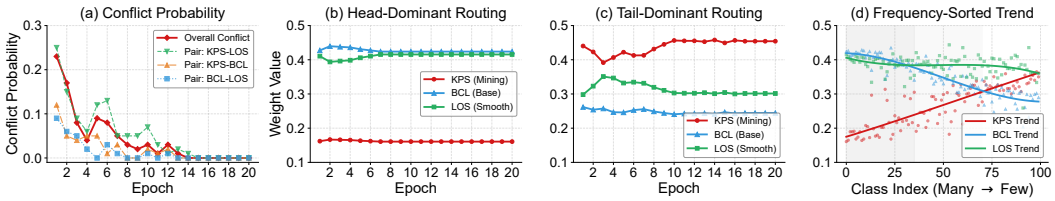


Figure 2: Optimization dynamics of U-MOF on CIFAR-100-LT with IR $= 100$. (a) Conflict-aware projection reduces overall and pairwise gradient conflicts during stage-two optimization. (b) and (c) show representative routing-weight dynamics for head-dominated and tail-dominated batches. (d) shows the final frequency-sorted routing trend across evaluation batches. These curves visualize batch-level routing coefficients rather than per-class trainable weights.

Table 3: Controlled stage-two engine comparison on CIFAR-100-LT with IR $= 100$. All methods use the same canonical objective pool $\{\mathcal{L}_{\mathrm{KPS}}, \mathcal{L}_{\mathrm{BCL}}, \mathcal{L}_{\mathrm{LOS}}, \ell_{\mathrm{feat}}^{\mathrm{div}}, \ell_{\mathrm{pred}}^{\mathrm{div}}\}$ under the same frozen-backbone stage-two scaffold. U-MOF is reported as mean $\pm$ standard deviation over five independent runs, while the other controlled baselines are reported under the matched protocol.

Stage-Two Engine	Head	Medium	Tail	Overall
Direct Linear Combination	77.5 $\pm$0.3	45.6 $\pm$0.4	16.5 $\pm$0.3	47.9 $\pm$0.5
TS-MOF Engine [17]	**79.0** $\pm$0.2	49.2 $\pm$0.3	39.9 $\pm$0.2	56.8 $\pm$0.3
U-MOF Engine (Ours)	76.8 $\pm$0.3	**51.4** $\pm$0.3	**41.9** $\pm$0.4	**57.5** $\pm$0.2

analysis, expert-number sensitivity, adapter-design ablations, measured complexity, and robustness to skewed validation splits. In particular, strict component-removal controls isolate the contributions of feature adaptation, diagnostic routing, and classification-first projection, while alternative routing policies examine whether committee diagnostics provide meaningful optimization signals. Expert-number and adapter-design studies further characterize the balance between recognition gains and lightweight adaptation cost. Together, these controls show that the gain is not explained by a different objective pool, a heavier expert scaffold, inference-time routing, or validation-split tuning.

Overall, the experiments provide three complementary levels of evidence: benchmark-level competitiveness, controlled improvement over prior stage-two coordination, and component-level support for lightweight feature adaptation, diagnostic routing, and conflict-aware projection. The combined findings highlight the importance of coordinating objective emphasis and limited feature plasticity within a parameter-efficient stage-two optimization framework.

5 Conclusion

We presented U-MOF, a parameter-efficient stage-two framework for long-tailed recognition. By combining residual adapters, committee diagnostics, and classification-first gradient projection, U-MOF improves objective routing and restores limited feature plasticity under a frozen-backbone setting. Experiments on CIFAR-100-LT, ImageNet-LT, and iNaturalist 2018 show competitive benchmark performance and improved rare-category recognition in controlled comparisons. Overall, U-MOF suggests that effective stage-two long-tailed adaptation depends not only on which objectives are optimized, but also on how objective emphasis is diagnosed, how gradient conflicts are coordinated, and how much lightweight feature plasticity is retained after backbone pretraining.

Acknowledgments

The authors gratefully acknowledge the support from the National Natural Science Foundation of China (NSFC) under Grant Nos. 62402472, 12227901, and 62476109. This work was also supported by the Natural Science Foundation of Jiangsu Province (No. BK20240461), the Project of Stable Support for Youth Team in Basic Research Field, CAS (No. YSBR-005), and the Academic Leaders

Table 4: Cumulative component ablation on ImageNet-LT. Starting from a classifier-only multi-objective stage-two reference configuration, we progressively add parameter-efficient adapters, diversity-aware projection, and U-PLA routing. This table should be interpreted as a progressive construction of the full framework rather than as a strict one-factor-at-a-time ablation.

Configuration	Adapter	Div. Obj.	Projection	U-PLA	Head	Medium	Tail	Overall
Classifier-Only Multi-Obj. Baseline	×	×	×	×	66.0 ± 0.4	35.0 ± 0.2	6.8 ± 0.6	43.1 ± 0.3
+ Feature Tuning	✓	×	×	×	66.5 ± 0.3	42.1 ± 0.1	18.3 ± 0.5	48.2 ± 0.2
+ Diversity and Projection	✓	✓	✓	×	67.4 ± 0.2	48.5 ± 0.2	35.6 ± 0.4	54.8 ± 0.2
Full U-MOF	✓	✓	✓	✓	**67.6** ± 0.2	**51.3** ± 0.3	**51.5** ± 0.3	**57.6** ± 0.1

Table 5: Stage-two efficiency comparison on CIFAR-100-LT with ResNet-34 and $E = 3$. U-MOF uses one shared frozen-backbone forward and lightweight adapter-head branches. The table summarizes parameter and inference-side stage-two footprint; training-time multi-objective gradient overhead is discussed in Appendix F.1.

Method	Backbone Forwards	Optimized Stage-Two Params	Extra Stage-Two MACs / Ratio	Inference Rule
Classifier-Only Stage Two	1	Heads only	Low	Single head
Full Multi-Expert	E	Expert backbones + heads	High	Expert ensemble
TS-MOF Engine (matched scaffold)	1	Matched	Matched	Stage-two output
U-MOF	1	252.204K	251.904K / 0.0217%	Committee average

Cultivation Program at USTC. The AI-driven experiments, simulations, and model training were performed on the robotic AI-Scientist platform of the Chinese Academy of Sciences.

Competing Interests. The authors declare that they have no competing interests.

References

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

[2] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

[3] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. 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*, pages 9719–9728, 2020.

[4] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020.

[5] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella X Yu. Long-tailed recognition by routing diverse distribution-aware experts. *arXiv preprint arXiv:2010.01809*, 2020.

[6] Shaden Alshammari, Yu-Xiong Wang, Deva Ramanan, and Shu Kong. Long-tailed recognition via weight balancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6897–6907, 2022.

[7] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *ArXiv*, abs/2007.07314, 2020.

[8] Tianhong Li, Peng Cao, Yuan Yuan, Lijie Fan, Yuzhe Yang, Rogerio Feris, Piotr Indyk, and Dina Katabi. Targeted supervised contrastive learning for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[9] Mengke Li, Zhikai Hu, Yang Lu, Weichao Lan, Yiu-Ming Cheung, and Hui Huang. Feature fusion from head to tail for long-tailed visual recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[10] Qihao Zhao, Yalun Dai, Hao Li, Wei Hu, Fan Zhang, and Jun Liu. Ltgc: Long-tail recognition via leveraging llms-driven generated content. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[11] Jiaan Luo, Feng Hong, Qiang Hu, Xiaofeng Cao, Feng Liu, and Jiangchao Yao. Long-tailed recognition with model rebalancing. *arXiv preprint arXiv:2510.08177*, 2025.

[12] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

[13] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR, 2018.

[14] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33:5824–5836, 2020.

[15] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in neural information processing systems*, 34:18878–18890, 2021.

[16] Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. *arXiv preprint arXiv:2202.01017*, 2022.

[17] Zhe Zhao, Zhiheng Gong, Pengkun Wang, HaiBin Wen, Cankun Guo, Bo Xue, Xi Lin, Zhenkun Wang, Qingfu Zhang, and Yang Wang. Ts-mof: Two-stage multi-objective fine-tuning for long-tailed recognition. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[18] Fudong Lin and Xu Yuan. Long-tailed recognition via information-preservable two-stage learning. *arXiv preprint arXiv:2510.08836*, 2025.

[19] Bowen Dong, Pan Zhou, Shuicheng Yan, and Wangmeng Zuo. Lpt: Long-tailed prompt tuning for image classification. *arXiv preprint arXiv:2210.01033*, 2022.

[20] Jie Huang, Zhao-Min Chen, Xiaoqin Zhang, Yisu Ge, Lusi Ye, Guodao Zhang, and Huiling Chen. Label decoupling and reconstruction: A two-stage training framework for long-tailed multi-label medical image recognition. In *Proceedings of the ACM International Conference on Multimedia*, 2024.

[21] Jiahao Chen, Bin Qin, Jiangmeng Li, Hao Chen, and Bing Su. Rethinking the bias of foundation model under long-tailed distribution. In *International Conference on Machine Learning*, 2025.

[22] Yufei Peng, Yonggang Zhang, and Yiu-Ming Cheung. Semantic-guided fine-tuning of foundation model for long-tailed visual recognition. *arXiv preprint arXiv:2507.12807*, 2025.

[23] Zhipeng Zhou, Liu Liu, Peilin Zhao, and Wei Gong. Pareto deep long-tailed recognition: A conflict-averse solution. In *The Twelfth International Conference on Learning Representations*, 2023.

[24] Xiaolei Wei, Yi Ouyang, and Haibo Ye. Divide, weight, and route: Difficulty-aware optimization with dynamic expert fusion for long-tailed recognition. *arXiv preprint arXiv:2508.19630*, 2025.

[25] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.

[26] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2537–2546, 2019.

[27] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018.

[28] Yuan Dong, Zhe Zhao, Liheng Yu, Di Wu, and Pengkun Wang. Guide: Gated uncertainty-informed disentangled experts for long-tailed recognition. In *International Conference on Learning Representations*, volume 2026, pages 26004–26034, 2026.

[29] Yuan Dong, Zhe Zhao, Liheng Yu, Di Wu, Pengkun Wang, and Yang Wang. Manidream: Manifold completion via prototype-anchored dreaming for long-tailed visual recognition. *International Journal of Computer Vision*, 134(8):386, 2026.

[30] Zhuang Qi, Lei Meng, Zhaochuan Li, Han Hu, and Xiangxu Meng. Cross-silo feature space alignment for federated learning on clients with imbalanced data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 19986–19994, 2025.

[31] Zhuang Qi, Yu Pan, Lei Meng, Sijin Zhou, Han Yu, Xiaoxiao Li, and Xiangxu Meng. Global prompt refinement with non-interfering attention masking for one-shot federated learning. *Advances in Neural Information Processing Systems*, 38:12121–12145, 2026.

[32] Zhuang Qi, Ying-Peng Tang, Lei Meng, Guoqing Chao, Lei Wu, Han Yu, and Xiangxu Meng. From selection to scheduling: Federated geometry-aware correction makes exemplar replay work better under continual dynamic heterogeneity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17494–17504, June 2026.

[33] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

[34] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[35] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018.

[36] Jianggang Zhu, Zheng Wang, Jingjing Chen, Yi-Ping Phoebe Chen, and Yu-Gang Jiang. Balanced contrastive learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6908–6917, 2022.

[37] Mengke Li, Yiu-Ming Cheung, and Zhikai Hu. Key point sensitive loss for long-tailed visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4812–4825, 2022.

[38] Yan Jin, Mengke Li, Yang Lu, Yiu-ming Cheung, and Hanzi Wang. Long-tailed visual recognition via self-heterogeneous integration with knowledge excavation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23695–23704, 2023.

[39] Siyu Sun, Han Lu, Jiangtong Li, Yichen Xie, Tianjiao Li, Xiaokang Yang, Liqing Zhang, and Junchi Yan. Rethinking classifier re-training in long-tailed recognition: Label over-smooth can balance. In *The Thirteenth International Conference on Learning Representations*, 2025.

[40] S Divakar Bhat, Amit More, Mudit Soni, and Surbhi Agrawal. Prior2posterior: Model prior correction for long-tailed learning. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1289–1298. IEEE, 2025.

[41] Jiang-Xin Shi, Tong Wei, Zhi Zhou, Jie-Jing Shao, Xin-Yan Han, and Yu-Feng Li. Long-tail learning with foundation model: Heavy fine-tuning hurts. *arXiv preprint arXiv:2309.10019*, 2023.

[42] Changyao Tian, Wenhai Wang, Xizhou Zhu, Jifeng Dai, and Yu Qiao. Vl-ltr: Learning class-wise visual-linguistic representation for long-tailed visual recognition. In *European Conference on Computer Vision*, 2022.

[43] Mengke Li, Ye Liu, Yang Lu, Yiqun Zhang, Yiu-ming Cheung, and Hui Huang. Improving visual prompt tuning by gaussian neighborhood minimization for long-tailed visual recognition. *Advances in Neural Information Processing Systems*, 37:103985–104009, 2024.

[44] Bolian Li and Ruqi Zhang. Making reliable and flexible decisions in long-tailed classification. *arXiv preprint arXiv:2501.14090*, 2025.

[45] David S. Hayden, Mao Ye, Timur Garipov, Gregory P. Meyer, Carl Vondrick, Zhao Chen, Yuning Chai, Eric M. Wolff, and Siddhartha S. Srinivasa. Generative data mining with longtail-guided diffusion. In *International Conference on Machine Learning*, 2025.

[46] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMlR, 2019.

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

Appendix

U-MOF: Uncertainty-Guided Parameter-Efficient Multi-Objective Fine-Tuning for Long-Tailed Recognition

Sections A–D provide extended related work, formulation, routing, and implementation. Section E reports objective-pool comparisons, matched baselines, attribution controls, routing interventions, feature audits, prior-correction comparisons, and frozen-encoder transfer. Section F summarizes computational cost, architecture, defaults, and sensitivity. Section G discusses limitations and broader impacts.

A Extended Related Work

This appendix provides a broader discussion of the literature most relevant to U-MOF. We organize the discussion around three directions: long-tailed recognition and stage-two adaptation, uncertainty-aware learning, and multi-objective optimization with gradient coordination. The goal of this section is not to exhaustively survey all long-tailed recognition methods, but to clarify how U-MOF differs from representative lines of work and why the proposed combination of diagnostic routing, lightweight feature adaptation, and conflict-aware coordination is useful for stage-two long-tailed fine-tuning.

A.1 Additional Long-Tailed Recognition Methods

Long-tailed recognition has been extensively studied because real-world visual data usually follow highly imbalanced class distributions. A small number of head categories contain abundant examples, while many tail categories contain only a few training samples. This imbalance causes standard empirical risk minimization to bias the learned representation and classifier toward frequent classes. Existing methods address this issue from several complementary perspectives, including loss re-balancing, re-sampling, classifier adjustment, representation learning, and multi-expert modeling.

Loss re-balancing and classifier adjustment. A large family of methods modifies the training objective to compensate for the skewed class distribution. Class-balanced re-weighting and re-sampling strategies attempt to equalize the contribution of different classes during optimization. Margin-based methods such as LDAM-DRW introduce class-dependent margins so that tail classes receive stronger decision constraints during training [1]. Balanced Softmax and logit adjustment take a probabilistic view and correct the decision rule according to class priors, thereby reducing the bias induced by imbalanced label frequencies [4, 7]. These methods demonstrate that the classifier and loss design are crucial under long-tailed distributions. However, many of them still optimize a single objective or rely on a fixed correction rule, which may be insufficient when different regions of the label distribution require different optimization behaviors.

Decision-prior correction. Logit Adjustment [7] and Prior2Posterior [40] address prior-induced prediction bias. Prior2Posterior estimates an effective model prior from predicted probabilities and applies post-hoc posterior correction. U-MOF instead studies the feature-update space, online allocation over an existing objective pool, and gradient coordination. These interventions need not be mutually exclusive. Appendix E.13 compares both prior-correction methods under the same frozen-backbone setting and evaluates U-MOF combined with Logit Adjustment; it separates this compatibility question from the parameter-matched capacity controls.

Decoupled learning and two-stage training. Another influential direction separates representation learning from classifier learning. Decoupled training shows that a strong representation learned from the original imbalanced data can be reused in a second stage, where the classifier is retrained or adjusted with a more balanced objective [2]. This observation has strongly shaped later long-tailed recognition research because it suggests that imbalance-aware adaptation can often be performed more efficiently after a generic representation has been obtained. Subsequent methods further explore how the second stage should be designed, including better classifier calibration, improved stage-two objectives, and more structured adaptation strategies.

U-MOF follows this decoupled view but focuses on a different question. Instead of treating the second stage as a simple classifier retraining step, we view it as a targeted optimization process that

must decide how to route heterogeneous objectives and how much feature-level plasticity should remain available after the backbone is frozen. This perspective is important because classifier-only adaptation may not provide sufficient flexibility for rare categories whose decision boundaries require modest feature refinement.

Representation refinement and contrastive learning. Several recent approaches improve long-tailed recognition by enhancing representation learning. Supervised contrastive learning and targeted contrastive objectives improve class separation by encouraging samples from the same class to cluster while separating different classes [8]. Head-to-tail feature transfer and related feature augmentation methods attempt to enrich tail representations by transferring useful information from head categories [9]. These methods show that tail recovery is not only a classifier-level problem; it also depends on whether the learned feature space supports sufficient discrimination for rare classes.

This observation motivates the feature-plasticity component of U-MOF. However, instead of fully retraining or substantially modifying the backbone, U-MOF introduces lightweight residual adapters in the second stage. This allows limited expert-specific feature modulation while retaining the stability and efficiency of frozen-backbone fine-tuning.

Multi-expert long-tailed recognition. Multi-expert architectures provide another way to improve long-tailed recognition. RIDE is a representative example, showing that routing diverse distribution-aware experts can improve long-tailed performance by maintaining expert complementarity [5]. Other multi-branch or expert-based methods similarly exploit specialization to address different parts of the class distribution. These methods are effective, but they often require multiple heavy branches or end-to-end expert training, which can increase training and inference cost.

U-MOF also uses an expert committee, but its design is different from full multi-expert systems. All experts share the same frozen backbone, and only lightweight residual adapters and classifier heads are trained in stage two. Therefore, the committee is used not only to improve prediction, but also to produce online diagnostic statistics for objective routing. This shared-backbone design keeps the method parameter-efficient while still allowing expert-level specialization.

Lightweight and stage-two adaptation. Recent long-tailed recognition methods have increasingly explored lightweight adaptation and explicitly designed second-stage optimization. TS-MOF shows that two-stage multi-objective fine-tuning can coordinate multiple long-tailed objectives after representation learning, highlighting the importance of stage-two weighting and gradient coordination [17]. Information-Preservable Two-Stage Learning also revisits two-stage long-tailed recognition and emphasizes that preserving useful information during representation learning can benefit the later rebalancing stage [18]. Other recent studies investigate prompt-style tuning, visual-language adaptation, foundation-model adaptation, and parameter-efficient fine-tuning under long-tailed distributions [41, 19, 42, 43, 21, 22, 11].

These developments are closely related to U-MOF because they support the view that the second stage is not merely a *post hoc* classifier adjustment step. However, U-MOF differs by explicitly coupling three ingredients within a single stage-two framework: residual expert adapters for feature plasticity, entropy-based committee diagnostics for objective routing, and classification-first projection for conflict-aware optimization. In this sense, U-MOF is best understood as a parameter-efficient stage-two optimization framework rather than a heavier end-to-end multi-expert architecture.

A.2 Additional Uncertainty-Aware Learning Approaches

Predictive uncertainty has been widely studied as a way to characterize model confidence, ambiguity, and reliability. Classical approaches distinguish different sources of uncertainty. For example, ensemble-based methods estimate predictive uncertainty by comparing predictions across multiple models, while Bayesian and evidential approaches attempt to model uncertainty in parameters or class-probability distributions [33–35]. These methods have been used in calibration, active learning, out-of-distribution detection, robust learning, and reliability analysis.

Uncertainty in long-tailed recognition. Uncertainty is especially relevant to long-tailed recognition because difficult samples are not homogeneous. Some samples are difficult because the model lacks sufficient knowledge of rare categories. Other samples are difficult because the visual evidence

is ambiguous, noisy, or intrinsically hard to classify. Treating these cases identically can lead to suboptimal optimization behavior. For example, aggressively increasing tail-oriented emphasis for inherently ambiguous samples may amplify noisy gradients, while applying only smoothing-oriented pressure to genuine tail-knowledge gaps may underfit rare categories.

Existing long-tailed methods often use loss values, confidence scores, class-wise accuracy, or validation feedback as routing or weighting signals. These signals are useful but largely symptomatic: they indicate that the model is struggling, but they do not directly separate different sources of difficulty. In contrast, U-MOF uses committee-derived statistics to obtain a more structured online diagnostic signal during stage-two optimization.

Diagnostic uncertainty proxies in U-MOF. U-MOF computes two entropy-based committee statistics: average within-expert predictive entropy and cross-expert disagreement. The former measures how uncertain individual experts are on average, while the latter measures how much the experts disagree after observing the same sample. These two quantities are used to route objective emphasis between smoothing-oriented and tail-oriented losses.

It is important to clarify that U-MOF does not claim to provide calibrated Bayesian uncertainty estimates. Since all experts share the same frozen backbone and differ only through lightweight adapters and classifier heads, the committee members should not be interpreted as posterior samples from a Bayesian model. The entropy identity is exact, but interpreting its terms as particular causal or Bayesian sources of uncertainty is not warranted by the shared-backbone committee. This distinction is central to our formulation. The purpose of the committee statistics is not to solve uncertainty calibration, but to provide online routing signals that are more informative than delayed empirical summaries such as historical class-wise accuracy or validation feedback.

Relation to reliability and calibration. Recent work has also studied reliability, calibration, and decision quality under imbalanced distributions [44, 45]. These studies are complementary to U-MOF. Calibration-oriented methods usually focus on whether predicted probabilities are reliable at evaluation time, whereas U-MOF uses uncertainty-related statistics during training to route heterogeneous objectives. In other words, uncertainty in U-MOF is not primarily an output property to be calibrated, but an internal optimization signal that determines how the second stage allocates capacity across objective groups.

This design is particularly suitable for stage-two long-tailed fine-tuning. Since the backbone is frozen, the method needs a lightweight and online signal to determine when to emphasize tail-oriented adaptation and when to apply stabilizing smoothing. Committee diagnostics provide such a signal without requiring an additional validation-driven routing loop.

A.3 Additional Multi-Objective Optimization Background

The optimization component of U-MOF is related to multi-objective optimization and multi-task learning. A basic observation in this literature is that naively summing multiple objectives can be unstable when their gradients point in conflicting directions. Different objectives may prefer different descent directions, and a fixed weighted sum can overemphasize one objective while suppressing others.

Generic multi-objective and multi-task optimization. Multi-Task Learning as Multi-Objective Optimization provides a Pareto-based formulation for balancing multiple objectives [12]. GradNorm adaptively balances tasks by normalizing gradient magnitudes [13]. PCGrad reduces destructive interference by projecting gradients when they conflict [14]. Subsequent methods, such as CAGrad and Nash-MTL, further study conflict-averse or game-theoretic formulations for stable multi-objective optimization [15, 16]. These methods show that adaptive weighting and gradient coordination can be more effective than naive linear aggregation.

Multi-objective optimization in long-tailed learning. Long-tailed recognition naturally involves heterogeneous objectives. Tail-oriented losses encourage rare-category recovery, smoothing-oriented objectives stabilize uncertain predictions, and diversity regularizers maintain expert complementarity. These objectives do not necessarily agree in parameter space. For example, a diversity objective may encourage experts to separate, while a classification objective may require consistent decision

boundaries for a given class. If such gradients are directly summed, the resulting update may partially cancel useful classification directions or destabilize stage-two adaptation.

Recent long-tailed work has begun to explicitly study optimization conflicts. PLOT analyzes category-level conflicts from a Pareto perspective and shows that conflict-aware optimization can improve long-tailed recognition [23]. TS-MOF introduces a two-stage multi-objective fine-tuning engine and highlights the importance of stage-two objective weighting and coordination [17]. DQRoute and related routing-based methods further suggest that adaptive objective or expert allocation can be beneficial under imbalanced distributions [24].

Difference from generic solvers. U-MOF is related to these optimization methods but has a different focus. We do not propose a generic multi-objective solver intended for arbitrary tasks. Instead, we study how gradient coordination should be integrated into a stage-two long-tailed fine-tuning framework where the objective family has a specific structure: tail-oriented objectives, smoothing-oriented objectives, and diversity-promoting regularizers.

This structure motivates the classification-first projection strategy in U-MOF. Classification-related gradients are first coordinated to form an aggregate descent direction, and diversity gradients are then projected against this direction. This ordering reflects the role of diversity in our framework: diversity should encourage expert specialization, but it should not directly oppose classification improvement. Therefore, the projection module is not used as an isolated solver; it is coupled with diagnostic routing and residual expert adaptation to form a complete stage-two optimization loop.

Summary. Overall, prior long-tailed recognition methods motivate the need for efficient second-stage adaptation, uncertainty-aware learning motivates diagnostic routing, and multi-objective optimization motivates conflict-aware gradient coordination. U-MOF combines these ideas in a parameter-efficient stage-two framework. Its contribution lies not in introducing each ingredient independently, but in showing how lightweight feature plasticity, committee-derived routing diagnostics, and classification-first gradient projection can be jointly organized to improve long-tailed fine-tuning under a shared frozen-backbone setting.

B Extended Formulation

This appendix supplements the main text with the detailed formulation of U-MOF. We focus on three aspects that are central to the proposed stage-two optimization framework. First, we formalize the objective family used in stage-two long-tailed fine-tuning. Second, we provide the complete parameterization of the residual expert adapters. Third, we describe the entropy-based committee diagnostics used later for objective routing. Throughout this appendix, U-MOF should be understood as a stage-two optimization framework: it does not redefine the internal forms of existing long-tailed objectives, but instead specifies how heterogeneous objectives are parameterized, routed, and coordinated under a shared frozen-backbone setting.

B.1 Stage-Two Objective Family

U-MOF uses a fixed objective family during stage-two adaptation. Let

$$\Omega = \{(\phi_e, \psi_e)\}_{e=1}^{E} \tag{19}$$

denote the trainable stage-two parameters, where ϕ_e and ψ_e are the residual adapter and classifier-head parameters of expert e, respectively. The backbone parameters θ remain frozen throughout stage two.

The objective family is divided into three groups:

$$\mathcal{O} = \mathcal{O}^{\mathrm{mine}} \cup \mathcal{O}^{\mathrm{smooth}} \cup \mathcal{O}^{\mathrm{div}}. \tag{20}$$

The mining group $\mathcal{O}^{\mathrm{mine}}$ contains objectives that emphasize rare categories, hard examples, or low-density decision regions. The smoothing group $\mathcal{O}^{\mathrm{smooth}}$ contains objectives that stabilize uncertain predictions and reduce over-confident adaptation. The diversity group $\mathcal{O}^{\mathrm{div}}$ contains regularizers that encourage the expert branches to remain complementary rather than collapse to identical solutions.

In the main experiments, we instantiate the objective family with the following canonical assembly:

$$\mathcal{O}^{\text{mine}} = \{\mathcal{L}_{\text{KPS}}, \mathcal{L}_{\text{BCL}}\}, \qquad \mathcal{O}^{\text{smooth}} = \{\mathcal{L}_{\text{LOS}}\}, \qquad \mathcal{O}^{\text{div}} = \{\ell^{\text{div}}_{\text{feat}}, \ell^{\text{div}}_{\text{pred}}\}. \tag{21}$$

Here, $\mathcal{L}_{\text{KPS}}$ and $\mathcal{L}_{\text{BCL}}$ follow their original formulations and serve as representative tail-oriented objectives, while $\mathcal{L}_{\text{LOS}}$ is used as the smoothing-oriented objective. The two diversity objectives are defined in Appendix C.3.

Expert-wise application of classification objectives. For classification-related objectives

$$k \in \mathcal{O}^{\text{mine}} \cup \mathcal{O}^{\text{smooth}},$$

U-MOF applies the corresponding objective to the expert-specific prediction rather than to a single committee-averaged prediction. Concretely, for input-label pairs $(x, y) \in \mathcal{B}_t$, expert e produces an adapted feature h_e, pooled feature $\tilde{z}_e$, logits s_e, and predictive distribution $p_e(\cdot \mid x)$. The expert-wise objective is written as

$$\ell^t_k(\Omega) = \frac{1}{E}\sum_{e=1}^{E} \frac{1}{|\mathcal{B}_t|} \sum_{(x,y)\in\mathcal{B}_t} \mathcal{L}_k\left(s_e(x), \tilde{z}_e(x), y\right), \tag{22}$$

where $\mathcal{L}_k$ denotes the original loss form of objective k. The arguments in Eq. (22) are written generically because different losses use different inputs. For example, cross-entropy or label-smoothing objectives operate mainly on logits and labels, while contrastive or balanced objectives may additionally use feature representations or classifier weights. This notation preserves the original formulation of each objective while making explicit that classification-related losses are applied at the expert branch level.

Diversity objectives across experts. Unlike the classification-related objectives, the diversity objectives are defined across the expert committee. The feature-space diversity term $\ell^{\text{div}}_{\text{feat}}$ is computed from the collection of expert-adapted features, while the prediction-space diversity term $\ell^{\text{div}}_{\text{pred}}$ is computed from the collection of expert logits or predictive distributions. These objectives are therefore not applied independently to each expert. Instead, they regularize the relationships among experts so that the committee maintains complementary feature responses and non-identical predictive behavior. Their explicit forms are provided in Appendix C.3.

Objective gradients. For a mini-batch $\mathcal{B}_t$ at iteration t, let $\ell^t_k(\Omega)$ denote the k-th objective in the stage-two objective pool. Its gradient with respect to the trainable stage-two parameters is

$$g^t_k = \nabla_\Omega \ell^t_k. \tag{23}$$

The goal of U-MOF is to construct a final update direction from the set of objective gradients $\{g^t_k\}_{k\in\mathcal{O}}$. This process involves three design choices: lightweight feature adaptation through residual expert adapters, diagnostic objective routing through committee-derived statistics, and conflict-aware gradient coordination through classification-first projection.

Unless otherwise specified, the controlled comparisons in the main text use the canonical objective assembly in Eq. (21). Additional objective assemblies are evaluated in Appendix E.1 to test whether U-MOF remains effective beyond this default configuration.

B.2 Residual Expert Adapters

For generality, we describe the adapter on a spatial feature map. In our ResNet implementation, the backbone returns a pooled descriptor, and the same residual modulation is applied directly to this descriptor; the implementation-specific form is provided in Appendix F.2.

The purpose of the residual expert adapter is to restore limited feature-level plasticity while keeping the pretrained backbone frozen. Let

$$h = f_\theta(x) \in \mathbb{R}^{d\times H\times W} \tag{24}$$

be the final convolutional feature map produced by the frozen backbone for input x. For each expert e, U-MOF attaches a channel-wise residual modulation block to this shared feature map.

We first compute a pooled descriptor:

$$v = \mathrm{GAP}(h) \in \mathbb{R}^d, \tag{25}$$

where $\mathrm{GAP}(\cdot)$ denotes global average pooling over spatial dimensions. The adapter then produces a channel-wise gate through a bottleneck multilayer perceptron:

$$g_e = \mathrm{Sigmoid}\left(W_e^{\mathrm{up}}\sigma\left(W_e^{\mathrm{down}}v + b_e^{\mathrm{down}}\right) + b_e^{\mathrm{up}}\right), \tag{26}$$

where

$$W_e^{\mathrm{down}} \in \mathbb{R}^{(d/r)\times d}, \qquad W_e^{\mathrm{up}} \in \mathbb{R}^{d\times(d/r)},$$

$b_e^{\mathrm{down}} \in \mathbb{R}^{d/r}$, $b_e^{\mathrm{up}} \in \mathbb{R}^d$, r is the bottleneck reduction ratio, and $\sigma(\cdot)$ denotes a nonlinear activation function.

A direct residual form such as $h + h \odot g_e$ would rescale the feature map even when the gate is initialized around $1/2$. To ensure that the adapter starts from a near-identity transformation, we use a centered residual modulation:

$$h_e = h + \alpha_e h \odot \left(g_e - \frac{1}{2}\right), \tag{27}$$

where $\odot$ denotes element-wise multiplication with broadcasting over spatial dimensions, and α_e is a residual scaling coefficient. The default is a fixed $\alpha_e = 0.10$; the learnable-scale variant is evaluated separately in Appendix F.8. This keeps the adapter close to identity while allowing gradients to flow through the residual branch at the beginning of stage two.

The classifier head operates on the pooled adapted feature:

$$\tilde{z}_e = \mathrm{GAP}(h_e), \tag{28}$$

and produces logits and predictive probabilities:

$$s_e = W_e^{\mathrm{cls}}\tilde{z}_e + b_e^{\mathrm{cls}}, \qquad p_e(y \mid x) = \mathrm{softmax}(s_e). \tag{29}$$

Trainable parameter count. Ignoring optional normalization layers, the adapter parameters for each expert include the bottleneck projection, expansion layer, and their biases. If α_e is learnable, it contributes one additional scalar. Therefore, the number of trainable adapter parameters per expert is

$$P_{\mathrm{adapter}} = \frac{2d^2}{r} + \frac{d}{r} + d + I_\alpha, \tag{30}$$

where $I_\alpha = 1$ if α_e is learnable and 0 otherwise. The classifier head contributes $Cd + C$ trainable parameters. Thus, for E experts, the total number of trainable stage-two parameters is

$$P_{\mathrm{stage2}} = E\left(\frac{2d^2}{r} + \frac{d}{r} + d + I_\alpha + Cd + C\right). \tag{31}$$

This parameter count is substantially smaller than full-backbone fine-tuning because the backbone parameters θ remain frozen. At the same time, the adapter provides more flexibility than classifier-only adaptation because each expert can modulate the feature representation before classification.

Role of the adapter. The adapter is intentionally constrained. It does not replace the frozen backbone with a second full representation learner; it exposes a restricted feature-update space for stage-two adaptation. This is a design choice, not a claim that the adapter is a minimal or sufficient representation learner for every rare category. This design is particularly useful for rare categories, where classifier-only updates may be insufficient when the frozen feature space does not already separate tail classes well. By attaching expert-specific residual adapters, U-MOF allows different experts to refine channel responses in complementary ways while preserving the efficiency and stability of a shared frozen backbone.

B.3 Committee Prediction and Diagnostic Decomposition

The residual adapters and classifier heads form a lightweight expert committee. For input x, each expert outputs a predictive distribution $p_e(\cdot \mid x)$. The committee-averaged prediction is

$$\bar{p}(\cdot \mid x) = \frac{1}{E}\sum_{e=1}^{E} p_e(\cdot \mid x). \tag{32}$$

We define the total committee entropy as

$$U^{\mathrm{tot}}(x) = H\left(\bar{p}(\cdot \mid x)\right), \tag{33}$$

where

$$H(q) = -\sum_{c=1}^{C} q_c \log q_c \tag{34}$$

denotes Shannon entropy.

Following the standard entropy-based ensemble decomposition, we compute two committee statistics. The first is the average within-expert predictive entropy:

$$U^{\mathrm{ent}}(x) = \frac{1}{E}\sum_{e=1}^{E} H\left(p_e(\cdot \mid x)\right). \tag{35}$$

This statistic measures how uncertain the individual experts are on average. The second is the cross-expert disagreement:

$$U^{\mathrm{dis}}(x) = H\left(\bar{p}(\cdot \mid x)\right) - \frac{1}{E}\sum_{e=1}^{E} H\left(p_e(\cdot \mid x)\right). \tag{36}$$

By construction,

$$U^{\mathrm{tot}}(x) = U^{\mathrm{ent}}(x) + U^{\mathrm{dis}}(x). \tag{37}$$

Interpretation as diagnostic proxies. Although Eq. (37) has the same algebraic form as common ensemble uncertainty decompositions, its interpretation in U-MOF is deliberately operational. Because all experts share the same frozen backbone and differ only through lightweight adapters and classifier heads, the experts should not be interpreted as posterior samples from a Bayesian model. Therefore, U^{ent} and U^{dis} are not treated as calibrated Bayesian uncertainty estimates. Instead, they are used as diagnostic proxies for stage-two objective routing.

Specifically, $U^{\mathrm{dis}}(x)$ measures cross-expert disagreement under the shared feature backbone. A high value indicates that the expert committee has not reached a consistent decision pattern for the input. In long-tailed fine-tuning, such disagreement can indicate regions where additional tail-oriented adaptation may be useful. In contrast, $U^{\mathrm{ent}}(x)$ measures the average predictive entropy of individual experts. A high value suggests uncertain or ambiguous predictions, for which smoothing-oriented objectives can provide stabilizing pressure.

Batch-level and normalized statistics. During stage-two training, routing is computed at the mini-batch level. For a mini-batch $\mathcal{B}_t$, we aggregate the sample-wise statistics as

$$\bar{U}_t^{\mathrm{dis}} = \frac{1}{|\mathcal{B}_t|}\sum_{x\in\mathcal{B}_t} U^{\mathrm{dis}}(x), \qquad \bar{U}_t^{\mathrm{ent}} = \frac{1}{|\mathcal{B}_t|}\sum_{x\in\mathcal{B}_t} U^{\mathrm{ent}}(x). \tag{38}$$

To keep the routing scale comparable across datasets with different numbers of classes and experts, we further normalize the diagnostics as

$$\tilde{U}_t^{\mathrm{ent}} = \frac{\bar{U}_t^{\mathrm{ent}}}{\log C}, \qquad \tilde{U}_t^{\mathrm{dis}} = \frac{\bar{U}_t^{\mathrm{dis}}}{\log E}, \tag{39}$$

where C is the number of classes and $E > 1$ is the number of experts. The $E = 1$ experiment is a non-committee reference and does not use this logarithmic disagreement normalization. The normalized quantities $\tilde{U}_t^{\mathrm{ent}}$ and $\tilde{U}_t^{\mathrm{dis}}$ are then used by the U-PLA routing mechanism in Appendix C.2.

Why committee diagnostics are useful for stage-two routing. Delayed empirical summaries, such as validation accuracy or accumulated class-wise performance, are useful but coarse. They indicate where the model performs poorly, but they do not directly reveal whether the difficulty is associated with disagreement among experts or broad predictive ambiguity. The committee diagnostics in Eq. (39) provide a lightweight online alternative. They are computed from the current mini-batch and require no additional validation-driven routing loop. This makes them suitable for parameter-efficient stage-two adaptation, where the optimizer must decide how to allocate limited update capacity across heterogeneous objectives while keeping the backbone frozen.

C Routing and Optimization Details

This appendix complements Secs. 3.3 and 3.4. The main text presents the core routing rule and the projection primitive needed to keep the framework self-contained. Here, we provide the complete routing normalization rule, define the diversity objectives used in the canonical objective assembly, and describe the full classification-first projection recursion used to construct the final stage-two update direction.

Throughout this section, the committee-derived quantities are interpreted as diagnostic routing proxies rather than calibrated Bayesian uncertainty estimates. Specifically, U^{dis} denotes cross-expert disagreement, and U^{ent} denotes average within-expert predictive entropy. This terminology is consistent with the shared frozen-backbone expert committee used by U-MOF.

C.1 Batch-Level Routing Statistics

For a mini-batch $\mathcal{B}_t$, U-MOF first computes sample-wise committee diagnostics from the expert predictive distributions. As defined in Appendix B.3, the cross-expert disagreement and within-expert entropy for an input x are

$$U^{\mathrm{dis}}(x) = H(\bar{p}(\cdot \mid x)) - \frac{1}{E}\sum_{e=1}^{E} H(p_e(\cdot \mid x)), \tag{40}$$

and

$$U^{\mathrm{ent}}(x) = \frac{1}{E}\sum_{e=1}^{E} H(p_e(\cdot \mid x)), \tag{41}$$

where

$$\bar{p}(\cdot \mid x) = \frac{1}{E}\sum_{e=1}^{E} p_e(\cdot \mid x) \tag{42}$$

is the committee-averaged predictive distribution, and $H(\cdot)$ denotes entropy.

The sample-wise diagnostics are then aggregated over the current mini-batch:

$$\bar{U}_t^{\mathrm{dis}} = \frac{1}{|\mathcal{B}_t|}\sum_{x\in\mathcal{B}_t} U^{\mathrm{dis}}(x), \qquad \bar{U}_t^{\mathrm{ent}} = \frac{1}{|\mathcal{B}_t|}\sum_{x\in\mathcal{B}_t} U^{\mathrm{ent}}(x). \tag{43}$$

To make the routing scale comparable across datasets, we normalize these quantities as

$$\tilde{U}_t^{\mathrm{dis}} = \frac{\bar{U}_t^{\mathrm{dis}}}{\log E}, \qquad \tilde{U}_t^{\mathrm{ent}} = \frac{\bar{U}_t^{\mathrm{ent}}}{\log C}. \tag{44}$$

The normalized quantities summarize complementary aspects of the current mini-batch. A large $\tilde{U}_t^{\mathrm{dis}}$ indicates that the expert committee exhibits substantial disagreement under the shared frozen backbone, suggesting that additional tail-oriented adaptation may be useful. A large $\tilde{U}_t^{\mathrm{ent}}$ indicates that individual experts are broadly uncertain, suggesting that smoothing-oriented objectives may provide stabilizing pressure. U-MOF uses these statistics only as operational routing diagnostics, not as calibrated uncertainty estimates.

C.2 Objective-Level Routing Coefficients

U-MOF converts the normalized batch-level diagnostics in Eq. (44) into objective-level routing coefficients through an uncertainty-guided proxy loss allocation rule, denoted as U-PLA. The objective family is divided into three groups:

$$\mathcal{O} = \mathcal{O}^{\mathrm{mine}} \cup \mathcal{O}^{\mathrm{smooth}} \cup \mathcal{O}^{\mathrm{div}}. \tag{45}$$

The mining group $\mathcal{O}^{\mathrm{mine}}$ contains tail-oriented objectives, the smoothing group $\mathcal{O}^{\mathrm{smooth}}$ contains stabilizing objectives, and the diversity group $\mathcal{O}^{\mathrm{div}}$ contains expert-diversity regularizers.

Group-level scores. We first compute unnormalized group-level routing scores:

$$\rho_t^{\text{mine}} = \exp\Big(\beta_m \tilde{U}_t^{\text{dis}}\Big), \qquad \rho_t^{\text{smooth}} = \exp\Big(\beta_s \tilde{U}_t^{\text{ent}}\Big), \qquad \rho_t^{\text{div}} = \exp(\beta_d), \tag{46}$$

where $\beta_m > 0$ and $\beta_s > 0$ control the sensitivity of the mining and smoothing groups to the two normalized diagnostic statistics, and β_d controls the baseline emphasis assigned to the diversity group.

This exponential mapping has two practical advantages. First, it ensures that all group scores are positive. Second, it amplifies differences in diagnostic statistics smoothly, allowing the optimizer to shift objective emphasis without requiring discontinuous routing decisions.

Group-level normalization. The group-level scores are normalized by a softmax-like rule:

$$\omega_t^q = \frac{\rho_t^q}{\rho_t^{\text{mine}} + \rho_t^{\text{smooth}} + \rho_t^{\text{div}}}, \qquad q \in \{\text{mine}, \text{smooth}, \text{div}\}. \tag{47}$$

By construction,

$$\omega_t^q > 0, \qquad \sum_{q \in \{\text{mine},\text{smooth},\text{div}\}} \omega_t^q = 1. \tag{48}$$

Intra-group distribution. Within each group, the group-level weight is distributed among its member objectives using fixed positive intra-group priors. Let $\pi_k > 0$ denote the prior assigned to objective k. For objective $k \in \mathcal{O}^q$, its final routing coefficient is

$$\lambda_k^t = \omega_t^q \frac{\pi_k}{\sum_{j \in \mathcal{O}^q} \pi_j}. \tag{49}$$

Therefore,

$$\lambda_k^t > 0, \qquad \sum_{k \in \mathcal{O}} \lambda_k^t = 1. \tag{50}$$

Given the gradient $g_k^t = \nabla_\Omega \ell_k^t$ of the k-th objective, the routed gradient is

$$\hat{g}_k^t = \lambda_k^t g_k^t. \tag{51}$$

Equivalently, Eq. (49) can be written in the compact form used in the main text:

$$\lambda_k^t = \frac{\rho_t^q}{\sum_{q'} \rho_t^{q'}} \cdot \frac{\pi_k}{\sum_{j \in \mathcal{O}^q} \pi_j}, \qquad k \in \mathcal{O}^q.$$

Canonical setting. In the main experiments, we use the canonical objective assembly

$$\mathcal{O}^{\text{mine}} = \{\mathcal{L}_{\text{KPS}}, \mathcal{L}_{\text{BCL}}\}, \qquad \mathcal{O}^{\text{smooth}} = \{\mathcal{L}_{\text{LOS}}\}, \qquad \mathcal{O}^{\text{div}} = \{\ell_{\text{feat}}^{\text{div}}, \ell_{\text{pred}}^{\text{div}}\}. \tag{52}$$

The defaults are $\beta_m = 3$, $\beta_s = 2$, $\beta_d = 0$, and $(\pi_{\text{feat}}, \pi_{\text{pred}}) = (0.10, 0.05)$. KPS and BCL have equal mining-group priors, and LOS is the only smoothing objective. Thus, the default intra-diversity allocation is $2/3$ to feature diversity and $1/3$ to prediction diversity. Priors and routing sensitivities are fixed across the repeated runs; the batch-dependent coefficients are computed analytically, not learned as additional parameters.

Interpretation. U-PLA changes relative emphasis rather than assigning batches exclusively to one objective group. High entropy can also occur on hard or rare examples, and both mining and smoothing remain active. The fixed diversity score is normalized jointly with the other scores, so its normalized mass can vary across batches even when β_d is constant. Committee statistics are used as operational diagnostics; the comparative routing and one-step intervention evidence is reported in Appendices E.7 and E.8.

Coefficient treatment in the update. Equation (51) defines the update by multiplying each objective gradient by its computed coefficient. It does not differentiate a scalar objective of the form $\sum_k \lambda_k^t(\Omega)\ell_k^t(\Omega)$, which would introduce additional derivatives through the routing coefficients. The prediction-diversity objective remains differentiable in its own right, even though its Jensen–Shannon statistic also supplies a routing diagnostic.

C.3 Diversity Objectives

U-MOF uses two diversity regularizers to reduce expert homogenization: a feature-space diversity objective and a prediction-space diversity objective. These objectives are not intended to replace classification losses. Instead, they provide specialization pressure so that different experts can learn complementary residual adaptations under the shared frozen backbone.

Feature-space diversity. Let $\tilde{z}_e(x) \in \mathbb{R}^d$ denote the pooled adapted feature of expert e for input x:

$$\tilde{z}_e(x) = \mathrm{GAP}(h_e). \tag{53}$$

We first normalize the feature:

$$\bar{z}_e(x) = \frac{\tilde{z}_e(x)}{\|\tilde{z}_e(x)\|_2 + \varepsilon}. \tag{54}$$

The feature-space diversity loss penalizes pairwise alignment between different experts:

$$\ell_{\mathrm{feat}}^{\mathrm{div}} = \frac{1}{|\mathcal{B}_t|} \sum_{x \in \mathcal{B}_t} \frac{2}{E(E-1)} \sum_{1 \le e < e' \le E} \left(\bar{z}_e(x)^\top \bar{z}_{e'}(x) \right)^2. \tag{55}$$

Minimizing Eq. (55) discourages different experts from producing overly aligned feature representations. The squared cosine form avoids favoring anti-parallel collapse and encourages low absolute alignment among expert features.

Prediction-space diversity. Feature diversity alone does not guarantee complementary decision behavior. Therefore, U-MOF also uses a prediction-space diversity objective. For input x, the expert predictions are $\{p_e(\cdot \mid x)\}_{e=1}^{E}$, and the committee average is $\bar{p}(\cdot \mid x)$. We define the Jensen-Shannon disagreement among experts as

$$\mathrm{JSD}(x) = H(\bar{p}(\cdot \mid x)) - \frac{1}{E} \sum_{e=1}^{E} H(p_e(\cdot \mid x)). \tag{56}$$

This quantity is identical in form to the cross-expert disagreement statistic, but it is used here as a prediction-level diversity regularizer rather than as a routing diagnostic.

To encourage complementary predictions, we minimize the negative Jensen-Shannon disagreement:

$$\ell_{\mathrm{pred}}^{\mathrm{div}} = -\frac{1}{|\mathcal{B}_t|} \sum_{x \in \mathcal{B}_t} \mathrm{JSD}(x). \tag{57}$$

This objective encourages experts to maintain non-identical predictive behavior. Because this term can encourage disagreement even when the classification signal is reliable, it is treated as an auxiliary objective and is always coordinated through classification-first projection rather than directly added to the classification gradients. Therefore, in U-MOF, the gradient of $\ell_{\mathrm{pred}}^{\mathrm{div}}$ is first routed and then projected against the aggregate classification direction as described in Appendix C.4. This classification-first projection reduces opposition to the aggregate classification direction; its numerical scope is described in Appendix C.4.

Combined diversity objective. For notational convenience, the total diversity regularization term can be written as

$$\mathcal{L}_{\mathrm{div}} = \lambda_{\mathrm{feat}}^t \ell_{\mathrm{feat}}^{\mathrm{div}} + \lambda_{\mathrm{pred}}^t \ell_{\mathrm{pred}}^{\mathrm{div}}, \tag{58}$$

where $\lambda_{\mathrm{feat}}^t$ and $\lambda_{\mathrm{pred}}^t$ are obtained from the routing rule in Eq. (49). In implementation, the two diversity losses are treated as separate objectives in $\mathcal{O}^{\mathrm{div}}$, so that their gradients can be individually routed and projected.

Role of diversity in U-MOF. The diversity objectives are intentionally auxiliary. Their purpose is to prevent expert collapse and promote complementary residual adaptations. The classification-first projection rule controls this specialization pressure relative to the aggregate classification direction. This design distinguishes U-MOF from methods that simply add diversity losses to the training objective without considering gradient interference.

C.4 Conflict-Aware Projection

After routing, U-MOF obtains a set of routed objective gradients:

$$\{\hat{g}_k^t\}_{k\in\mathcal{O}}. \tag{59}$$

Scalar routing controls the relative emphasis of objectives, but it cannot remove directional conflicts in parameter space. Therefore, U-MOF applies conflict-aware projection before constructing the final update direction.

Projection primitive. Given two gradients u and v, the projection operator is defined as

$$\Pi(u;v) = u - \mathbf{1}[\langle u,v\rangle < 0]\,\frac{\langle u,v\rangle}{\|v\|_2^2+\varepsilon}v, \tag{60}$$

where $\varepsilon > 0$ provides numerical stability. Compatible gradients are unchanged. For a negative inner product, the operation attenuates the opposing component. In particular,

$$\langle \Pi(u;v), v\rangle = \frac{\varepsilon\langle u,v\rangle}{\|v\|_2^2+\varepsilon} \quad \text{when } \langle u,v\rangle < 0.$$

Exact orthogonality applies to the unstabilized projection with $\varepsilon = 0$ and $v \neq 0$; the implemented expression is its stabilized form. The reference is the aggregate classification direction, not every individual classification gradient, and this operation does not establish monotonic held-out accuracy or Pareto optimality.

Classification-related aggregation. Let

$$\mathcal{O}^{\text{cls}} = \mathcal{O}^{\text{mine}} \cup \mathcal{O}^{\text{smooth}} \tag{61}$$

denote the classification-related objectives. U-MOF first coordinates these objectives to form an aggregate classification descent direction. Given a fixed deterministic order over $\mathcal{O}^{\text{cls}}$, we initialize

$$d_{\text{cls}}^t = 0. \tag{62}$$

For each $k \in \mathcal{O}^{\text{cls}}$, we compute

$$\tilde{g}_k^t = \begin{cases} \hat{g}_k^t, & \text{if } d_{\text{cls}}^t = 0, \\ \Pi(\hat{g}_k^t; d_{\text{cls}}^t), & \text{otherwise}, \end{cases} \tag{63}$$

and update

$$d_{\text{cls}}^t \leftarrow d_{\text{cls}}^t + \tilde{g}_k^t. \tag{64}$$

This step reduces conflicts among the classification-related objectives before any diversity regularizer is considered. The deterministic ordering is kept fixed across training, making the procedure reproducible. In the default implementation, we place tail-oriented objectives before smoothing-oriented objectives, which matches the intended role of first emphasizing rare-category recovery and then incorporating stabilizing smoothing pressure.

Diversity projection. After obtaining d_{cls}^t, U-MOF projects the diversity gradients against the aggregate classification direction. For each $k \in \mathcal{O}^{\text{div}}$, we compute

$$\tilde{g}_k^t = \Pi(\hat{g}_k^t; d_{\text{cls}}^t). \tag{65}$$

The final update direction is then

$$d_t = d_{\text{cls}}^t + \sum_{k\in\mathcal{O}^{\text{div}}} \tilde{g}_k^t. \tag{66}$$

The stage-two parameters are updated by

$$\Omega_{t+1} = \Omega_t - \eta d_t, \tag{67}$$

where η is the learning rate.

Why classification-first projection is used. The classification-first ordering reflects the intended role of diversity in U-MOF. Tail-oriented and smoothing-oriented objectives directly optimize recognition performance, whereas diversity objectives act as auxiliary regularizers that maintain expert complementarity. Therefore, diversity gradients should not be allowed to directly oppose the classification descent direction. Projecting diversity gradients against d_{cls}^t reduces their opposing component while retaining the aggregate classification direction as the primary reference.

Difference from generic gradient surgery. The projection rule in Eq. (60) is related to gradient-surgery methods, but its use in U-MOF is structurally constrained by the stage-two long-tailed objective family. Rather than applying pairwise projection uniformly among all objectives, U-MOF first forms a classification direction and then treats diversity as a regularizing component. This design matches the role of the objectives in long-tailed adaptation and avoids allowing auxiliary diversity objectives to dominate the update.

C.5 Summary of the Routing and Projection Procedure

The complete routing and projection procedure can be summarized as follows.

1. Compute expert predictions $\{p_e(\cdot \mid x)\}_{e=1}^{E}$ for each $x \in \mathcal{B}_t$.
2. Compute batch-level diagnostics $\bar{U}_t^{\mathrm{dis}}$ and $\bar{U}_t^{\mathrm{ent}}$.
3. Convert the diagnostics into group weights ω_t^{mine}, $\omega_t^{\mathrm{smooth}}$, and ω_t^{div}.
4. Distribute each group weight to objective-level coefficients $\{\lambda_k^t\}_{k\in\mathcal{O}}$ using intra-group priors.
5. Compute routed gradients $\hat{g}_k^t = \lambda_k^t g_k^t$.
6. Coordinate classification-related gradients to obtain d_{cls}^t.
7. Project diversity gradients against d_{cls}^t.
8. Aggregate the projected gradients into d_t and update Ω.

This procedure forms the optimization core of U-MOF. Diagnostic routing determines how much each objective should contribute, while classification-first projection determines which gradient components are compatible with the primary classification descent. Together, these two mechanisms allow U-MOF to coordinate heterogeneous stage-two objectives under a lightweight frozen-backbone adaptation setting.

D Optimization Algorithm and Implementation

This appendix provides the algorithmic and implementation details that support Sec. 3.1–3.4. The main text describes U-MOF as a two-stage framework: the first stage learns a generic representation, and the second stage freezes the backbone and updates only lightweight expert-specific parameters. Here, we provide explicit pseudocode for the two training stages, isolate the routing and projection subprocedure, and analyze the computational cost of the proposed stage-two optimizer.

D.1 Full Training Algorithm

Algorithms 1 and 2 give the two-stage training procedure. Algorithm 1 trains a generic backbone on the original long-tailed training set. Algorithm 2 freezes the backbone and updates only the residual expert adapters and classifier heads. During stage two, each mini-batch is processed by the shared frozen backbone, routed according to committee diagnostics, and optimized through classification-first conflict-aware projection.

The stage-one objective $\mathcal{L}_{\mathrm{base}}$ can be instantiated by the standard training loss used by the corresponding benchmark protocol. U-MOF does not require modifying the stage-one backbone training procedure. Its contribution is concentrated in stage two, where objective routing, lightweight feature adaptation, and gradient projection are applied to the frozen-backbone adaptation problem.

Algorithm 1 Stage-One Generic Representation Learning

Require: Long-tailed training set $\mathcal{D}$; backbone f_θ; learning rate η_1; stage-one epochs T_1
Ensure: Trained backbone f_θ

```
Initialize backbone parameters θ
for epoch = 1, ..., T_1 do
    for mini-batch B ⊂ D do
        Compute predictions using the backbone and the stage-one classifier
        Compute the stage-one classification loss L_base
        Update backbone parameters:
```

$$\theta \leftarrow \theta - \eta_1 \nabla_\theta \mathcal{L}_{\text{base}}$$

```
    end for
end for
```

D.2 Routing and Projection Procedure

Algorithm 3 isolates the routing and projection procedure used in every stage-two iteration. The procedure takes as input the objective gradients $\{g_k^t\}_{k\in\mathcal{O}}$, the batch-level committee diagnostics, and the objective family partition. It first computes U-PLA routing coefficients, then constructs the classification-related direction, and finally projects diversity gradients against this direction.

The projection operator $\Pi(\cdot;\cdot)$ is defined as:

$$\Pi(u; v) = u - \mathbf{1}[\langle u, v\rangle < 0] \, \frac{\langle u, v\rangle}{\|v\|_2^2 + \varepsilon} v. \tag{68}$$

The operator leaves compatible gradients unchanged and attenuates negative alignment with the reference. The stabilization term means that orthogonality is approximate, as detailed in Appendix C.4. Classification objectives are coordinated first, and diversity remains auxiliary to the resulting aggregate.

Ordering of classification objectives. In implementation, the order of classification-related objectives is kept fixed for reproducibility. We place tail-oriented objectives before smoothing-oriented objectives in the default configuration. This ordering first constructs a descent direction oriented toward rare-category recovery and then incorporates stabilizing smoothing pressure. We find this deterministic ordering more reproducible than random pairwise projection, and it aligns with the intended role of the objective groups.

Training-time only routing. U-PLA routing and conflict-aware projection are used only during stage-two training. At inference time, U-MOF does not compute routing coefficients, objective gradients, or projected updates. Prediction is obtained by averaging the expert predictive distributions, as described in Eq. (18).

D.3 Complexity Analysis

We analyze the additional cost introduced by U-MOF relative to a frozen-backbone stage-two baseline. The main cost comes from three sources: residual expert adapters, expert classifier heads, and multi-objective gradient coordination. Since the backbone is shared and frozen, U-MOF avoids the cost of full multi-expert backbone replication and full-parameter fine-tuning.

Trainable parameters. Let the backbone feature dimension be d, the number of classes be C, the bottleneck reduction ratio be r, and the number of experts be E. For each expert, the bottleneck adapter contains approximately

$$P_{\text{adapter}} \approx \frac{2d^2}{r} + \frac{d}{r} + d + I_\alpha, \tag{69}$$

trainable parameters, where $I_\alpha = 1$ if the residual scale α_e is learnable and 0 otherwise. The classifier head contains

$$P_{\text{head}} = Cd + C. \tag{70}$$

Algorithm 2 Stage-Two Parameter-Efficient Multi-Objective Fine-Tuning

Require: Long-tailed training set $\mathcal{D}$; frozen backbone f_θ; number of experts E; objective family $\mathcal{O}$; bottleneck reduction ratio r; learning rate η_2; stage-two epochs T_2
Ensure: Stage-two parameters $\Omega = \{(\phi_e, \psi_e)\}_{e=1}^{E}$

Freeze backbone parameters θ
Initialize expert adapters $\{a_{\phi_e}\}_{e=1}^{E}$ with bottleneck ratio r
Initialize classifier heads $\{h_{\psi_e}\}_{e=1}^{E}$
Set trainable stage-two parameters $\Omega = \{(\phi_e, \psi_e)\}_{e=1}^{E}$
for epoch $= 1, \ldots, T_2$ **do**
  for mini-batch $\mathcal{B}_t \subset \mathcal{D}$ **do**
    Compute shared frozen features $h = f_\theta(x)$, $x \in \mathcal{B}_t$
    for expert $e = 1, \ldots, E$ **do**
      Compute centered residual gate g_e using Eq. (26)
      Compute adapted feature:

$$h_e = h + \alpha_e h \odot \left(g_e - \frac{1}{2} \right)$$

      Compute expert logits $s_e = h_{\psi_e}(\mathrm{GAP}(h_e))$
      Compute expert prediction $p_e(\cdot \mid x) = \mathrm{softmax}(s_e)$
    end for
    Compute batch diagnostics $\bar{U}_t^{\mathrm{dis}}$ and $\bar{U}_t^{\mathrm{ent}}$
    Normalize diagnostics:

$$\tilde{U}_t^{\mathrm{dis}} = \frac{\bar{U}_t^{\mathrm{dis}}}{\log E}, \qquad \tilde{U}_t^{\mathrm{ent}} = \frac{\bar{U}_t^{\mathrm{ent}}}{\log C}$$

    Compute routing coefficients $\{\lambda_k^t\}_{k \in \mathcal{O}}$ using U-PLA
    for objective $k \in \mathcal{O}$ **do**
      Evaluate objective $\ell_k^t(\Omega)$
      Compute objective gradient:

$$g_k^t = \nabla_\Omega \ell_k^t$$

      Apply routing:

$$\hat{g}_k^t = \lambda_k^t g_k^t$$

    end for
    Compute projected update direction d_t using Algorithm 3
    Update stage-two parameters:

$$\Omega \leftarrow \Omega - \eta_2 d_t$$

  end for
end for

Therefore, the total number of trainable stage-two parameters is approximately

$$P_{\mathrm{stage2}} = E \left(\frac{2d^2}{r} + \frac{d}{r} + d + I_\alpha + Cd + C \right). \tag{71}$$

The backbone parameters are not included in P_{stage2}, since they are frozen during stage two and require no optimizer states.

Forward computation. At inference time, U-MOF requires one shared backbone forward pass. The additional expert computation consists of adapter gating, channel-wise modulation, classifier-head evaluation, and probability averaging. For one input, the approximate additional computation per expert is

$$\mathcal{C}_{\mathrm{expert}} \approx \frac{2d^2}{r} + dHW + Cd, \tag{72}$$

Algorithm 3 U-PLA Routing and Classification-First Projection

Require: Objective gradients $\{g_k^t\}_{k\in\mathcal{O}}$; normalized batch diagnostics $\tilde{U}_t^{\mathrm{dis}}, \tilde{U}_t^{\mathrm{ent}}$; objective groups $\mathcal{O}^{\mathrm{mine}}, \mathcal{O}^{\mathrm{smooth}}, \mathcal{O}^{\mathrm{div}}$; intra-group priors $\{\pi_k\}$; routing sensitivities $\beta_m, \beta_s, \beta_d$
Ensure: Final stage-two update direction d_t

Step 1: Compute group-level routing scores

$$\rho_t^{\mathrm{mine}} = \exp(\beta_m \tilde{U}_t^{\mathrm{dis}}), \qquad \rho_t^{\mathrm{smooth}} = \exp(\beta_s \tilde{U}_t^{\mathrm{ent}}), \qquad \rho_t^{\mathrm{div}} = \exp(\beta_d)$$

Step 2: Normalize group weights

$$\omega_t^q = \frac{\rho_t^q}{\rho_t^{\mathrm{mine}} + \rho_t^{\mathrm{smooth}} + \rho_t^{\mathrm{div}}}, \qquad q \in \{\mathrm{mine}, \mathrm{smooth}, \mathrm{div}\}$$

Step 3: Compute objective-level routing coefficients
for group $q \in \{\mathrm{mine}, \mathrm{smooth}, \mathrm{div}\}$ **do**
  for objective $k \in \mathcal{O}^q$ **do**

$$\lambda_k^t = \omega_t^q \frac{\pi_k}{\sum_{j\in\mathcal{O}^q} \pi_j}$$

$$\hat{g}_k^t = \lambda_k^t g_k^t$$

  end for
end for

Step 4: Build classification-related descent direction
Set $\mathcal{O}^{\mathrm{cls}} = \mathcal{O}^{\mathrm{mine}} \cup \mathcal{O}^{\mathrm{smooth}}$
Initialize $d_{\mathrm{cls}}^t = 0$
for objective $k \in \mathcal{O}^{\mathrm{cls}}$ in a fixed deterministic order **do**
  if $d_{\mathrm{cls}}^t = 0$ **then**
    $\tilde{g}_k^t = \hat{g}_k^t$
  else
    $\tilde{g}_k^t = \Pi(\hat{g}_k^t; d_{\mathrm{cls}}^t)$
  end if
  $d_{\mathrm{cls}}^t \leftarrow d_{\mathrm{cls}}^t + \tilde{g}_k^t$
end for

Step 5: Project diversity gradients
Initialize $d_t = d_{\mathrm{cls}}^t$
for objective $k \in \mathcal{O}^{\mathrm{div}}$ **do**
  $\tilde{g}_k^t = \Pi(\hat{g}_k^t; d_{\mathrm{cls}}^t)$
  $d_t \leftarrow d_t + \tilde{g}_k^t$
end for
return d_t

where dHW corresponds to channel-wise modulation of the feature map. Thus, the total additional computation for E experts is

$$\mathcal{C}_{\mathrm{extra}} \approx E\left(\frac{2d^2}{r} + dHW + Cd\right). \tag{73}$$

Because the backbone is evaluated only once, this cost is substantially smaller than full multi-expert systems that replicate the backbone for every expert.

Stage-two optimization cost. Let $K = |\mathcal{O}|$ be the number of objectives in the stage-two objective family, and let $P_\Omega = P_{\mathrm{stage2}}$ be the number of trainable stage-two parameters. Computing objective gradients over the lightweight parameter set has cost proportional to

$$\mathcal{O}(KP_\Omega) \tag{74}$$

after the forward graph for the stage-two branches has been constructed. The projection procedure requires inner products and vector updates between objective gradients, whose cost is

$$\mathcal{O}(KP_\Omega) \tag{75}$$

for the running-reference recursion in Algorithm 3: each classification gradient is checked against one running aggregate, and each diversity gradient against one fixed classification reference. An all-pairs implementation would instead require $\mathcal{O}(K^2P_\Omega)$. These counts describe vector coordination, not the complete cost of objective-specific backpropagation; measured stage-two runtime is reported separately.

Memory cost. U-MOF stores optimizer states only for the stage-two parameters Ω. Since the backbone is frozen, it does not require gradient buffers or optimizer states during stage-two training. Multi-objective optimization requires temporary storage for objective gradients. A straightforward implementation stores K gradient vectors over Ω, requiring

$$\mathcal{O}(KP_\Omega) \tag{76}$$

additional memory. This overhead can be further reduced by computing and projecting gradients sequentially, keeping only the current objective gradient and the running aggregate direction.

Inference-time cost. At inference time, U-MOF does not perform routing, gradient computation, or projection. It computes

$$p_{\text{U-MOF}}(\cdot \mid x) = \frac{1}{E}\sum_{e=1}^{E} p_e(\cdot \mid x)$$

after one shared backbone forward pass. Therefore, inference cost is dominated by the frozen backbone and lightweight expert heads. This is more expensive than a single classifier head but substantially cheaper than full multi-expert architectures with separate backbones.

Comparison with full fine-tuning and full multi-expert systems. Compared with full fine-tuning, U-MOF avoids updating the backbone and therefore reduces trainable parameters, optimizer states, and the risk of disrupting pretrained representations. Compared with full multi-expert systems, U-MOF avoids replicating the backbone across experts. The resulting design preserves the practical efficiency of two-stage adaptation while providing more flexibility than classifier-only fine-tuning.

D.4 Implementation Notes

We summarize several implementation details that are important for reproducibility and stable training.

Backbone freezing. During stage two, all backbone parameters are frozen. In implementation, the backbone forward can be computed without storing parameter gradients. The resulting feature map is then passed to the trainable expert adapters and classifier heads. This reduces memory consumption and ensures that stage-two adaptation remains parameter-efficient.

Adapter initialization. The centered adapter form

$$h_e = h + \alpha_e h \odot \left(g_e - \frac{1}{2}\right)$$

is used to make the adapter start close to the identity transformation. We initialize the final expansion layer of the adapter near zero so that $g_e \approx 1/2$ at the beginning of stage two. The residual scaling coefficient α_e is fixed to a small positive value, or initialized to a small positive value when implemented as a learnable scalar. This prevents abrupt perturbation of the frozen representation while still allowing gradients to flow through the residual branch during early adaptation.

Gradient computation. For each objective ℓ_k^t, we compute gradients with respect to the stage-two parameters Ω. The backbone parameters are excluded from the gradient list. In practice, one can use automatic differentiation to compute objective-specific gradients through `autograd.grad` and then apply routing and projection before assigning the final aggregated direction to the trainable parameters.

Numerical stability. The projection operator includes a small constant ε in the denominator:

$$\|v\|_2^2 + \varepsilon.$$

This prevents numerical instability when the reference gradient has a very small norm. We also normalize routing coefficients so that they are positive and sum to one, which prevents uncontrolled scaling of the objective family.

Routing hyperparameters. The routing sensitivities β_m, β_s, and β_d control the relative responsiveness of the mining, smoothing, and diversity groups. These coefficients are applied to normalized diagnostics $\tilde{U}_t^{\mathrm{dis}}$ and $\tilde{U}_t^{\mathrm{ent}}$, which improves comparability across datasets with different numbers of classes and experts. The structural defaults are selected before test evaluation and shared across the three ResNet benchmarks and repeated runs; the training schedules follow the corresponding benchmark protocols. We do not tune hyperparameters on the test set. Sensitivity analyses for U-PLA and the diversity regularization strength are provided in Appendix F.3 and Appendix F.4.

Intra-group priors. KPS and BCL use equal mining priors and LOS is the single smoothing objective. The default diversity priors are $(0.10, 0.05)$ for feature and prediction diversity. Their within-group normalization is retained when applying the allocation rule; alternative objective combinations use the corresponding objective groups.

Reproducibility. All reported repeated-run results are obtained using fixed training protocols and different random seeds. For controlled comparisons, the frozen backbone, objective pool, expert scaffold, and training schedule are kept identical unless explicitly stated. This ensures that differences among variants mainly reflect changes in routing, projection, or stage-two parameterization rather than unrelated implementation choices.

Practical implementation summary. The stage-two implementation can be summarized in four practical steps. First, freeze the pretrained backbone and initialize lightweight expert adapters and classifier heads. Second, compute committee predictions and batch-level diagnostic statistics for each mini-batch. Third, compute objective-specific gradients over the stage-two parameters and route them using U-PLA. Fourth, apply classification-first projection and update only the trainable stage-two parameters. This procedure is compatible with standard deep-learning frameworks and requires no modification to the backbone architecture.

E Extended Empirical Results and Analysis

This appendix presents additional empirical results that complement the main text. The main paper focuses on broad benchmark comparisons, controlled stage-two engine comparisons, component ablations, and validation-skew robustness. Here, we provide more detailed evidence for the robustness, attribution, diagnostic routing behavior, and inference behavior of U-MOF.

We organize this appendix as follows. Section E.1 evaluates whether U-MOF remains effective across diverse objective assemblies. Section E.2 studies the sensitivity to skewed validation splits. Section E.3 extends the component ablation across multiple imbalance ratios. Sections E.4–E.7 provide closest-baseline, strict component-removal, inference-rule, and routing/solver controls. Section E.8 summarizes the evidence supporting the diagnostic use of committee statistics. The remaining sections further present end-to-end trade-offs, optimization dynamics, and representation diversity analysis.

Note on Evaluation Metrics. Following standard long-tailed protocols, the "Tail" group refers to classes with fewer than 20 training samples. For CIFAR-100-LT with IR $= 10$, the minimum class size is 50, so the Tail group is empty and denoted by "$-$". The "Overall" or "All" metric is the macro-averaged top-1 accuracy across all 100 classes. Therefore, Overall accuracy may differ from a simple average of the reported Head and Medium groups when the Tail group is absent or when group sizes differ.

E.1 Robustness to Diverse Objective Configurations

A common concern in multi-objective learning is that performance gains may depend heavily on a specific hand-crafted combination of losses. To evaluate whether U-MOF behaves as a general stage-two optimization engine rather than a configuration-specific tuning strategy, we benchmark it across a broad collection of objective assemblies on CIFAR-100-LT under three imbalance ratios.

For each objective assembly, we compare three coordination strategies under the same objective pool: direct linear combination, the prior TS-MOF engine, and the proposed U-MOF engine. The results are shown in Table 6. The results in Table 6 show that U-MOF improves over direct linear

Table 6: Comprehensive objective-combination benchmark on CIFAR-100-LT across varying imbalance ratios. We follow the objective assemblies evaluated in TS-MOF [17] and compare direct linear aggregation, the TS-MOF engine, and the proposed U-MOF engine under each objective pool. U-MOF results are reported as mean ± standard deviation over repeated runs.

Objective Assembly / Engine	IR=10				IR=50				IR=100			
	Head	Medium	Tail	All	Head	Medium	Tail	All	Head	Medium	Tail	All
CE + BS + LOS	74.8	55.4	–	68.2	76.5	46.1	30.0	56.4	79.5	45.5	22.8	53.8
+ TS-MOF Engine	74.49	59.55	–	69.86	75.76	51.61	42.17	59.81	78.80	51.83	35.10	56.25
+ U-MOF Engine (Ours)	74.0 ±0.3	61.8 ±0.4	–	**70.4** ±0.2	75.4 ±0.2	52.8 ±0.3	**44.5** ±0.5	**60.5** ±0.2	77.2 ±0.3	52.6 ±0.4	**38.4** ±0.5	**57.0** ±0.2
CE + KPS + LOS	75.2	57.2	–	68.8	77.0	46.8	31.5	57.1	79.2	44.0	23.5	54.5
+ TS-MOF Engine	74.22	62.81	–	70.68	76.22	51.63	42.06	59.99	78.46	49.14	39.57	56.53
+ U-MOF Engine (Ours)	73.5 ±0.3	64.1 ±0.4	–	**71.0** ±0.2	75.5 ±0.3	53.0 ±0.4	**44.1** ±0.4	**60.6** ±0.2	76.9 ±0.2	50.8 ±0.3	**41.2** ±0.5	**57.2** ±0.2
CE + BCL + LOS	75.5	56.0	–	68.6	76.8	45.7	30.8	56.9	79.6	43.9	23.2	54.4
+ TS-MOF Engine	75.04	58.26	–	69.84	76.07	50.49	42.11	59.47	78.94	49.84	34.87	55.57
+ U-MOF Engine (Ours)	74.1 ±0.2	60.5 ±0.3	–	**70.5** ±0.2	75.4 ±0.2	52.0 ±0.4	**44.5** ±0.5	**60.2** ±0.2	76.9 ±0.3	51.6 ±0.3	**38.2** ±0.4	**56.5** ±0.2
CE + LDAM-DRW + LOS	75.6	56.7	–	68.4	76.6	46.3	30.2	56.6	79.4	44.6	22.9	54.0
+ TS-MOF Engine	74.90	59.71	–	70.19	75.95	51.05	42.61	59.74	78.23	51.34	35.20	55.91
+ U-MOF Engine (Ours)	74.0 ±0.3	61.2 ±0.4	–	**70.6** ±0.2	75.3 ±0.3	52.3 ±0.3	**44.2** ±0.4	**60.4** ±0.2	76.6 ±0.3	52.2 ±0.4	**38.0** ±0.5	**56.8** ±0.2
BS + KPS + LOS	74.8	57.4	–	68.9	76.5	47.0	31.2	57.3	78.5	46.3	23.8	54.8
+ TS-MOF Engine	73.99	61.42	–	70.09	75.61	53.17	42.00	60.36	77.49	51.54	38.27	56.64
+ U-MOF Engine (Ours)	73.4 ±0.3	63.5 ±0.3	–	**70.7** ±0.2	75.1 ±0.2	54.0 ±0.4	**44.8** ±0.5	**61.1** ±0.2	75.8 ±0.3	52.6 ±0.3	**40.5** ±0.4	**57.4** ±0.2
BS + BCL + LOS	75.0	56.8	–	68.7	77.0	46.5	31.5	57.0	79.5	45.1	23.5	54.2
+ TS-MOF Engine	74.25	60.06	–	69.85	76.39	51.20	42.17	59.90	78.89	51.49	34.70	56.04
+ U-MOF Engine (Ours)	73.7 ±0.2	62.5 ±0.4	–	**70.6** ±0.2	75.5 ±0.3	52.6 ±0.3	**44.5** ±0.4	**60.6** ±0.2	76.9 ±0.2	52.8 ±0.4	**38.5** ±0.5	**56.9** ±0.2
BS + LDAM-DRW + LOS	74.9	56.5	–	68.6	75.8	46.8	30.9	57.2	79.3	44.5	23.6	54.0
+ TS-MOF Engine	74.19	59.58	–	69.66	74.90	51.95	42.33	59.63	78.20	51.23	35.17	55.85
+ U-MOF Engine (Ours)	73.6 ±0.3	62.2 ±0.3	–	**70.5** ±0.2	74.3 ±0.2	53.3 ±0.4	**44.6** ±0.5	**60.5** ±0.2	76.5 ±0.3	52.5 ±0.3	**38.8** ±0.4	**56.7** ±0.2
KPS + LDAM-DRW + LOS	74.2	58.2	–	68.8	74.2	46.5	33.2	56.5	76.0	43.6	25.2	53.6
+ TS-MOF Engine	73.77	61.52	–	69.97	73.02	51.41	46.89	59.46	75.06	49.60	40.87	55.89
+ U-MOF Engine (Ours)	73.0 ±0.2	63.8 ±0.4	–	**70.8** ±0.2	72.5 ±0.3	52.6 ±0.3	**48.8** ±0.4	**60.3** ±0.2	73.5 ±0.3	51.1 ±0.4	**42.5** ±0.5	**56.6** ±0.2
KPS + BCL + LOS	75.5	58.5	–	69.4	76.5	46.2	32.1	58.1	79.5	44.1	25.4	54.5
+ TS-MOF Engine	74.75	62.03	–	70.81	75.24	50.76	47.56	60.22	79.00	49.29	39.97	56.89
+ U-MOF Engine (Ours)	73.2 ±0.3	**66.6** ±0.4	–	**71.1** ±0.2	74.9 ±0.2	**52.2** ±0.3	**49.9** ±0.4	**61.1** ±0.2	76.8 ±0.3	**51.4** ±0.3	**41.9** ±0.4	**57.5** ±0.2

aggregation across a diverse range of objective assemblies. Direct aggregation is often insufficient for coordinating heterogeneous long-tailed objectives, especially under severe imbalance. U-MOF generally provides stronger Medium, Tail, and Overall accuracy than the TS-MOF engine, suggesting that diagnostic routing and classification-first projection remain useful beyond the canonical objective pool.

E.2 Sensitivity to Skewed Validation

A practical concern in long-tailed learning is whether the overall training protocol is overly sensitive to the use of a balanced validation split. In U-MOF, the routing coefficients are computed from online committee diagnostics and do not directly depend on validation summaries. To evaluate stability, we construct an imbalanced validation split following the same skewed distribution as the training data.

Table 7: Sensitivity to validation skew on CIFAR-100-LT. U-MOF remains relatively stable when the validation split is imbalanced. LOS and TS-MOF under balanced validation are included as reference points.

Method	Validation Source	IR=10				IR=50				IR=100			
		Head	Medium	Tail	All	Head	Medium	Tail	All	Head	Medium	Tail	All
LOS	Balanced Val	71.9	62.3	–	69.0	72.4	51.4	40.2	58.0	70.3	52.3	36.6	53.9
TS-MOF	Balanced Val	74.7	62.0	–	70.8	75.2	50.7	47.5	60.2	79.0	49.2	39.9	56.8
U-MOF	Imbalanced Val	72.8 ±0.3	65.8 ±0.4	–	70.6 ±0.2	74.4 ±0.3	51.7 ±0.3	48.2 ±0.5	60.6 ±0.2	76.4 ±0.3	50.8 ±0.4	40.5 ±0.4	56.9 ±0.2
U-MOF	Balanced Val	73.2 ±0.3	66.6 ±0.4	–	71.1 ±0.2	74.9 ±0.2	52.2 ±0.3	49.9 ±0.4	61.1 ±0.2	76.8 ±0.3	51.4 ±0.3	41.9 ±0.4	57.5 ±0.2

Table 7 shows that replacing the standard balanced validation split with an imbalanced one causes only limited degradation for U-MOF, for example from 57.5% to 56.9% at IR $= 100$. This supports the stability of U-MOF under the reported skewed-validation setting.

E.3 Extended Ablation across Imbalance Ratios

We further examine the contribution of the main components on CIFAR-100-LT across three imbalance ratios. We consider three implementation factors: U-PLA routing, orthogonal projection, and committee averaging at inference.

Table 8: Extended component ablation across different imbalance ratios on CIFAR-100-LT. The full configuration consistently provides the strongest Tail and Overall performance under severe imbalance.

U-PLA	Orthogonal Proj.	Committee Avg.	IR=10				IR=50				IR=100			
			Head	Medium	Tail	All	Head	Medium	Tail	All	Head	Medium	Tail	All
×	×	×	72.3 ±0.3	47.3 ±0.4	–	64.6 ±0.2	74.2 ±0.3	43.5 ±0.4	22.9 ±0.6	52.4 ±0.3	76.5 ±0.3	46.9 ±0.4	18.1 ±0.6	48.6 ±0.3
×	×	✓	**74.7** ±0.2	60.8 ±0.3	–	70.4 ±0.2	**75.3** ±0.2	50.4 ±0.3	44.5 ±0.5	59.6 ±0.2	**78.2** ±0.2	49.1 ±0.4	38.6 ±0.5	56.1 ±0.2
✓	✓	×	73.1 ±0.3	65.5 ±0.4	–	70.6 ±0.2	74.5 ±0.3	51.5 ±0.3	48.5 ±0.4	60.5 ±0.2	76.2 ±0.3	50.9 ±0.3	40.5 ±0.5	57.0 ±0.2
×	✓	✓	74.3 ±0.2	60.3 ±0.4	–	69.9 ±0.3	75.5 ±0.3	51.0 ±0.3	43.5 ±0.5	59.7 ±0.2	78.4 ±0.2	49.8 ±0.4	36.9 ±0.5	56.0 ±0.2
✓	×	✓	73.5 ±0.3	64.2 ±0.4	–	70.3 ±0.2	74.8 ±0.2	51.8 ±0.3	46.8 ±0.5	60.1 ±0.2	76.5 ±0.3	50.2 ±0.4	39.5 ±0.5	56.3 ±0.2
✓	✓	✓	73.2 ±0.3	**66.6** ±0.4	–	**71.1** ±0.2	74.9 ±0.2	**52.2** ±0.3	**49.9** ±0.4	**61.1** ±0.2	76.8 ±0.3	**51.4** ±0.3	**41.9** ±0.4	**57.5** ±0.2

Adding U-PLA and orthogonal projection improves Tail and Overall accuracy, indicating that the training-time optimization mechanism itself is beneficial even before using the full committee prediction at inference.

E.4 Matched Closest-Baseline Comparison

TS-MOF is the closest stage-two baseline. Table 9 reports five-run matched reimplementations on CIFAR-100-LT, ImageNet-LT, and iNaturalist 2018. Within each dataset, the two engines share the frozen backbone, objective pool, adapter-head scaffold, stage-two schedule, validation protocol, expert number, and uniform probability averaging. The changed factor is the routing and gradient-coordination engine.

These matched results are distinct from the original-paper TS-MOF values quoted in Table 2. The latter are 65.9/52.5/42.6/56.3 on ImageNet-LT and 72.8/73.6/70.5/72.3 on iNaturalist 2018 (Head/Medium/Tail/Overall). The matched reimplementation below gives 66.2/50.8/43.1/55.9 and 73.1/72.8/70.9/71.8, respectively. The original-paper rows provide benchmark context and are not substituted into controlled engine comparisons.

Table 9: Matched stage-two engine comparison on three datasets. Both engines use the same adaptation scaffold and evaluation protocol within each dataset. All values are mean ± standard deviation over five runs; the TS-MOF rows are matched reimplementations, not the original-paper numbers quoted in Table 2.

Dataset	Method	Head	Medium	Tail	Overall
CIFAR-100-LT IR $= 100$	TS-MOF Engine	**79.0** ±0.2	49.2 ±0.3	39.9 ±0.2	56.8 ±0.3
	U-MOF Engine	76.8 ±0.3	**51.4** ±0.3	**41.9** ±0.4	**57.5** ±0.2
ImageNet-LT	TS-MOF	66.2 ±0.3	50.8 ±0.2	43.1 ±0.4	55.9 ±0.2
	U-MOF	**67.6** ±0.2	**51.3** ±0.3	**51.5** ±0.3	**57.6** ±0.1
iNaturalist 2018	TS-MOF	73.1 ±0.4	**72.8** ±0.3	70.9 ±0.4	71.8 ±0.2
	U-MOF	**76.4** ±0.2	70.8 ±0.2	**75.1** ±0.3	**73.1** ±0.1

The trade-off depends on the dataset. U-MOF improves Medium/Tail/Overall on CIFAR-100-LT while reducing Head by 2.2 points. On ImageNet-LT it improves all four reported metrics relative to the matched engine. On iNaturalist 2018 it improves Head/Tail/Overall but lowers Medium by 2.0 points. The controlled claim is therefore improved Tail and Overall under a shared scaffold, not uniform group-wise dominance or equal measured wall-clock cost.

E.5 Controlled Component Removal Analysis

To further isolate the role of each component, we provide a controlled component-removal analysis where each variant removes or replaces one major design choice from the full U-MOF framework. Small numerical deviations from the extended ablation in Section E.3 are due to independent repeated runs. The variant "w/o Residual Adapter" removes feature modulation but retains multiple expert classifier heads and prediction-space diversity. Feature-space diversity is necessarily disabled in this variant because expert-specific feature modulation is no longer available.

Table 10: Component-removal and capacity controls on CIFAR-100-LT with IR $= 100$. Routing and projection removals keep the adapter-head capacity fixed. Removing adapters also disables the unavailable expert-feature diversity term; the wider-head control matches the reported 252.204K parameter budget. Values are mean $\pm$ standard deviation over five runs.

Variant	Head	Medium	Tail	Overall
Full U-MOF	76.8 $\pm$0.3	**51.4** $\pm$0.3	**41.9** $\pm$0.4	**57.5** $\pm$0.2
Parameter-matched wider classifier heads	77.6 $\pm$0.3	49.6 $\pm$0.4	37.0 $\pm$0.5	55.9 $\pm$0.3
w/o Residual Adapter *(Classifier-Only Heads)*	**77.9** $\pm$0.3	49.5 $\pm$0.3	36.8 $\pm$0.5	55.8 $\pm$0.3
w/o U-PLA Routing *(Equal Weighting)*	78.2 $\pm$0.2	48.9 $\pm$0.4	37.5 $\pm$0.5	56.1 $\pm$0.2
w/o Classification-First Projection	77.4 $\pm$0.3	50.1 $\pm$0.3	38.6 $\pm$0.6	56.4 $\pm$0.3
w/o Diversity Objectives	77.1 $\pm$0.3	50.8 $\pm$0.2	40.2 $\pm$0.4	57.0 $\pm$0.2

Wider classifier heads remain close to classifier-only adaptation despite matching the reported adapter-head parameter budget. Conversely, removing routing or projection leaves feature-adaptation capacity in place but reduces performance. These controls support joint contributions from feature plasticity and its optimization, without treating the adapter-removal row as a pure one-variable test of capacity.

E.6 Inference Rule and Committee Averaging

U-MOF uses objective routing and gradient projection only during stage-two training. At inference time, it averages expert predictions.

Table 11: Inference-rule comparison on CIFAR-100-LT with IR $= 100$. All rows evaluate the same trained U-MOF model using different prediction aggregation rules. The default U-MOF inference uses uniform committee averaging and does not use routing or projection at test time.

Inference Rule	Head	Medium	Tail	Overall
Primary Expert Only	76.5 $\pm$0.4	50.9 $\pm$0.3	40.8 $\pm$0.5	56.9 $\pm$0.3
Best Single Expert *(Validation-Selected)*	76.7 $\pm$0.3	51.1 $\pm$0.4	41.0 $\pm$0.4	57.1 $\pm$0.2
Uniform Committee Average *(Default)*	**76.8** $\pm$0.3	**51.4** $\pm$0.3	**41.9** $\pm$0.4	**57.5** $\pm$0.2

E.7 Alternative Routing and Solver Comparisons

The routing and solver comparisons use the same CIFAR-100-LT IR=100 adaptation scaffold. Reverse U-PLA exchanges the disagreement and entropy assignments; this is a diagnostic-assignment control, not sufficient evidence by itself that the proposed assignment is optimal. Table 12 retains the detailed routing and solver results, while Table 13 tests alternatives based on random variation, class frequency, online loss, an early-frozen allocation rule, and a learned scalar router.

The random control matches routing moments; class-frequency and online-loss controls replace the committee diagnostics with their named signals. The early-frozen control stops updating the routing rule after epoch 5, while the learned scalar router is evaluated under the matched trainable budget. U-PLA has the highest reported mean Tail/Overall in this comparison. The learned-router gap is small, so the table supports a favorable empirical operating point rather than a statistical-significance or universal-optimality claim. The solver block separately evaluates how classification-first ordering compares with generic coordination choices under the same objective pool.

Table 12: Alternative routing and solver comparisons on CIFAR-100-LT with IR $= 100$.

Variant Type	Method	Head	Medium	Tail	Overall
Weighting Policy	Confidence Scaling	**79.1** ± 0.3	47.5 ± 0.4	31.4 ± 0.5	53.5 ± 0.3
	Pure Entropy Scaling	78.5 ± 0.3	49.3 ± 0.4	38.2 ± 0.5	56.3 ± 0.2
	Empirical PLA *(TS-MOF)*	79.0 ± 0.2	49.2 ± 0.3	39.9 ± 0.2	56.8 ± 0.3
	Reverse U-PLA *(Negative Control)*	78.5 ± 0.3	47.8 ± 0.5	33.2 ± 0.6	54.6 ± 0.3
	U-PLA	76.8 ± 0.3	**51.4** ± 0.3	**41.9** ± 0.4	**57.5** ± 0.2
Gradient Solver	MGDA	78.2 ± 0.3	49.6 ± 0.4	38.5 ± 0.5	56.4 ± 0.3
	GradNorm	77.8 ± 0.4	48.1 ± 0.4	36.2 ± 0.5	55.2 ± 0.3
	Base PCGrad *(TS-MOF)*	**79.0** ± 0.5	49.2 ± 0.3	39.9 ± 0.2	56.8 ± 0.4
	Classification-First Projection	76.8 ± 0.3	**51.4** ± 0.3	**41.9** ± 0.4	**57.5** ± 0.2

Table 13: Additional routing controls on CIFAR-100-LT with IR=100. The frozen backbone, adapter scaffold, objective pool, schedule, inference rule, and trainable-parameter budget are matched. All results are mean $\pm$ standard deviation over five runs.

Routing policy	Tail	Overall
Random routing, matched moments	36.9 $\pm$ 0.5	55.8 $\pm$ 0.3
Class-frequency routing	39.4 $\pm$ 0.5	56.5 $\pm$ 0.2
Online-loss routing	39.8 $\pm$ 0.5	56.6 $\pm$ 0.2
Frozen after epoch 5	40.7 $\pm$ 0.5	57.1 $\pm$ 0.2
Learned scalar router, matched budget	41.2 $\pm$ 0.4	57.3 $\pm$ 0.2
U-PLA	41.9 $\pm$ 0.4	57.5 $\pm$ 0.2

E.8 Diagnostic Routing Validity and One-Step Intervention Audit

Routing controls assess whether the diagnostic assignment is useful, but a frequency correlation alone does not establish that the preferred objective produces a better subsequent update. A one-step intervention audit therefore compares virtual mining-only and smoothing-only updates from the same training state. The stage-two parameters are cloned, the two updates are norm-matched, and their immediate effects are evaluated on a separate held-out evaluation mini-batch. This evaluation batch is excluded from gradient computation, actual parameter updates, model selection, and hyperparameter tuning. The virtual branches do not replace the training trajectory.

Mining more often gives the larger immediate Tail-loss reduction for high-disagreement batches, while smoothing more often gives the larger all-class negative-log-likelihood reduction for high-entropy batches after disagreement matching. Shuffling the diagnostics weakens this association. These results provide intervention-based empirical support for the allocation hypothesis, but do not identify calibrated uncertainty types or prove that either objective is optimal for every batch. High entropy may reflect difficult rare examples as well as ambiguity, and positive normalized weights keep both classification groups active.

E.9 Parameter-Efficient Fine-Tuning versus End-to-End Optimization

To further clarify the role of stage-two parameter efficiency, we compare the proposed frozen-backbone design with an end-to-end variant in which the full backbone is unfrozen.

This comparison is illustrative rather than absolute. End-to-end fine-tuning introduces additional optimization choices, such as balancing backbone and head learning rates and avoiding disruption of the stage-one representation. Under this straightforward unfrozen-backbone setting, constrained stage-two plasticity provides a more stable adaptation path for tail recovery.

E.10 Optimization Dynamics

To better understand the role of conflict-aware projection, we track gradient conflicts throughout training on CIFAR-100-LT. A conflict is recorded whenever the inner product between the gradients of two distinct objectives is negative: $\langle g_i^t, g_j^t \rangle < 0$.

Table 14: One-step intervention audit. Percentages are computed within each seed over audited batch pairs and then summarized across five seeds. The first two rows use different evaluation criteria and should not be interpreted as a common accuracy metric.

Audit condition	Preferred virtual update	Frequency (%)
Top-quartile disagreement; larger Tail-loss reduction	Mining	69.4 $\pm$ 2.1
Top-quartile entropy after matching disagreement; larger all-class NLL reduction	Smoothing	66.8 $\pm$ 1.9
Shuffled diagnostic assignment	Diagnostic control	50.7 $\pm$ 1.8

Table 15: Illustrative comparison between a straightforward unfrozen-backbone variant and parameter-efficient stage-two adaptation on CIFAR-100-LT with IR $= 100$. The end-to-end variant is not intended as a fully tuned baseline.

Optimization Paradigm	Head	Medium	Tail	Overall (%)
Unfrozen Backbone *(End-to-End MOO)*	**78.5** $\pm$0.3	42.1 $\pm$0.4	19.8 $\pm$0.6	49.3 $\pm$0.3
Frozen Backbone + Residual Adapters *(Ours)*	76.8 $\pm$0.3	**51.4** $\pm$0.3	**41.9** $\pm$0.4	**57.5** $\pm$0.2

Figure 3 shows that U-MOF reduces the observed conflict rate during stage-two optimization, which is consistent with the design goal of mitigating destructive interference among heterogeneous objectives.

E.11 Representation Diversity Analysis

We analyze whether the residual adapters and diversity-aware optimization encourage expert differentiation in the shared feature space.

Representation diversity. We compute linear Centered Kernel Alignment (CKA) [46] between expert feature representations on the CIFAR-100-LT validation set. CKA is computed using pre-classifier adapted features and averaged over all expert pairs. **U-MOF expert pairs exhibit an average CKA similarity of 0.71**, indicating non-identical expert representations. This suggests that the parameter-efficient experts do not simply collapse to identical feature representations despite relying on a shared frozen backbone.

Frequency-stratified representation audit. CKA measures between-expert similarity, not tail-feature quality. A separate audit therefore uses pre-classifier features to examine class-wise discrimination. Class centroids are computed from labeled training features, and held-out validation features are used only for analysis. The margin is cosine similarity to the true centroid minus similarity to the nearest incorrect centroid; compactness is cosine similarity to the true centroid. Validation labels and audit statistics are not used for optimization, model selection, or hyperparameter tuning.

Tail margin and compactness increase with adapters and improve further with the complete optimization procedure, while Head margin decreases slightly. Class-wise margin gain has a Spearman correlation of -0.47 ± 0.04 with log class frequency. This supports a frequency-dependent improvement pattern, not a claim that the adapter architecture is inherently tail-selective or that it uses a different sampler.

Qualitative Feature Visualization. We extract pre-classifier features from the CIFAR-100-LT validation set and project them using t-SNE.

Figure 4 provides qualitative supporting evidence that U-MOF yields more separated projected clusters for underrepresented classes compared to the baseline configuration, suggesting that lightweight feature-level plasticity helps organize rare-category representations.

E.12 Adapter Placement across Training Stages

The placement control uses the same adapter architecture and total training budget. The stage-one-only variant trains adapters with stage-one empirical risk minimization and freezes them during classifier-only stage two. The both-stage variant continues updating the same adapters during stage

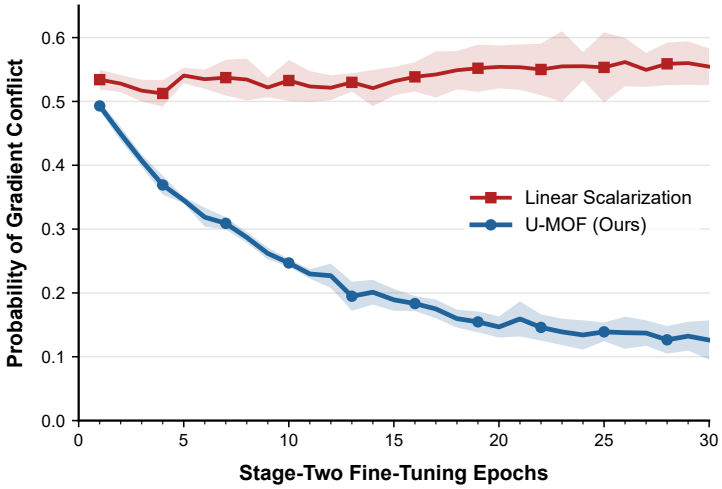


Figure 3: **Temporal evolution of gradient conflict probabilities on CIFAR-100-LT.** The naive linear scalarization baseline maintains a relatively high conflict rate throughout training, while U-MOF shows a lower and gradually decreasing conflict rate.

Table 16: Frequency-stratified feature audit on CIFAR-100-LT with IR=100. Values are mean ± standard deviation over five runs. The metrics describe the pre-classifier representation rather than calibrated predictive uncertainty.

Representation	Head margin	Medium margin	Tail margin	Tail compactness
Classifier-only	0.238 ± 0.003	0.148 ± 0.003	0.074 ± 0.004	0.612 ± 0.006
Adapter + equal weighting	0.236 ± 0.003	0.157 ± 0.003	0.091 ± 0.004	0.628 ± 0.005
Full U-MOF	0.233 ± 0.003	0.166 ± 0.003	0.108 ± 0.003	0.641 ± 0.004

two. The default stage-two-only variant inserts near-identity adapters after backbone freezing, leaving the original representation-learning stage unchanged.

Stage-two insertion and both-stage training have similar performance relative to the observed run-to-run variability. The result supports stage-two insertion as an effective way to preserve generic representation learning and isolate imbalance-aware adaptation, not as evidence that earlier adapter training is intrinsically inferior. The adapter itself is not class-selective; the objectives and coordinated updates determine its frequency-dependent behavior.

E.13 Prior-Correction Comparisons and Compatibility

Logit Adjustment [7] and Prior2Posterior [40] address decision-prior bias. The comparison uses the same frozen backbone, data split, stage-two training budget, validation protocol, and five-run reporting. Both prior-correction baselines are applied to the same classifier-only stage-two setting. In U-MOF + Logit Adjustment, prior correction is applied to expert logits while retaining U-MOF's adapter scaffold and uniform committee inference. The comparison with the classifier-only methods is not an all-method parameter-matched feature-ablation; the capacity-matched test is reported separately in Appendix E.5.

Adding Logit Adjustment to U-MOF improves Tail/Overall by 0.8/0.4 percentage points with a 0.4-point Head decrease. This provides evidence of compatibility in the evaluated setting. Prior correction and diagnostic coordination address different parts of the adaptation procedure; neither result implies that one universally subsumes the other.

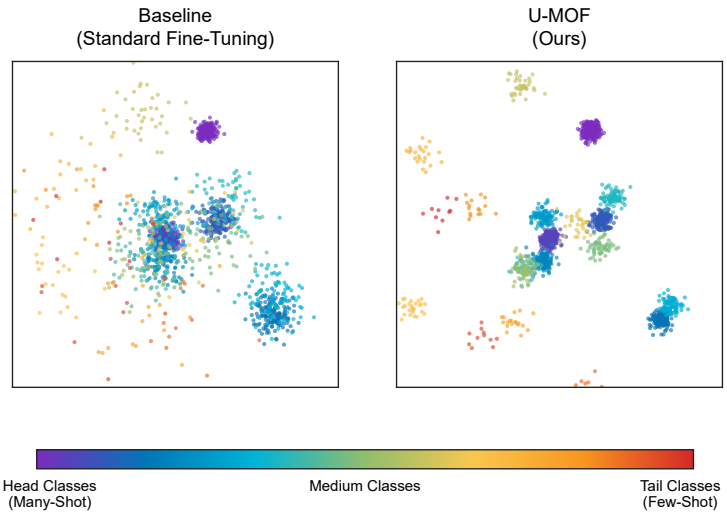


Figure 4: **t-SNE visualization of feature representations across class frequencies on CIFAR-100-LT.** The color gradient indicates class frequency, ranging from Head classes to Tail classes.

Table 17: Adapter-placement controls on CIFAR-100-LT with IR=100. Results are mean ± standard deviation over five runs.

Placement	Head	Medium	Tail	Overall
Stage-one adapter only	77.8 ± 0.3	49.7 ± 0.3	37.2 ± 0.5	56.0 ± 0.2
Adapters in both stages	77.0 ± 0.3	51.1 ± 0.3	41.5 ± 0.4	57.4 ± 0.2
Stage-two only, U-MOF	76.8 ± 0.3	51.4 ± 0.3	41.9 ± 0.4	57.5 ± 0.2

E.14 Transfer to a Frozen CLIP Image Encoder

A separate ImageNet-LT experiment evaluates pooled-embedding adaptation with a frozen CLIP ViT-B/16 image encoder [47]. The final two rows share the frozen encoder, adapter architecture, canonical objective pool, $E = 3$, training schedule, classification-first projection, uniform committee inference, and trainable-parameter budget; only U-PLA is removed from the middle row. The frozen linear probe provides a representation-specific baseline.

Relative to the parameter-matched adapter baseline, U-MOF improves Tail by 1.4 points and Overall by 0.6, with a 0.3-point Head decrease. The result demonstrates transfer for this particular frozen encoder and pooled-embedding adaptation setting. It does not establish performance across arbitrary vision transformers, layer-wise adapter placement, prompt tuning, or end-to-end foundation-model fine-tuning.

F Implementation Details and Hyperparameter Sensitivity

This appendix summarizes the implementation details of U-MOF and evaluates the sensitivity of its key hyperparameters.

F.1 Measured Computational Complexity

To explicitly demonstrate the practical efficiency of U-MOF, we report measured parameter counts, multiply-accumulate operations (MACs), wall-clock training time, and peak GPU memory for the CIFAR-100-LT controlled setting with ResNet-34 and $E = 3$. Extra MACs refer to the adapter-head branches and exclude the shared frozen-backbone forward pass. Only adapter and classifier-head parameters are passed to the stage-two optimizer. Wall-clock and memory measurements are recorded under the same batch-size, data-loading, and hardware configuration used in the controlled

Table 18: Prior correction and compatibility on CIFAR-100-LT with IR=100. Values are mean ± standard deviation over five runs.

Method	Head	Medium	Tail	Overall
Logit Adjustment	75.9 ± 0.3	49.1 ± 0.4	37.2 ± 0.5	55.9 ± 0.3
Prior2Posterior	76.1 ± 0.3	49.8 ± 0.3	38.6 ± 0.5	56.4 ± 0.2
U-MOF	76.8 ± 0.3	51.4 ± 0.3	41.9 ± 0.4	57.5 ± 0.2
U-MOF + Logit Adjustment	76.4 ± 0.3	51.9 ± 0.3	42.7 ± 0.4	57.9 ± 0.2

Table 19: ImageNet-LT transfer with a frozen CLIP ViT-B/16 image encoder. Results are mean ± standard deviation over five runs. These rows are separate from the ResNet-based benchmark comparison.

Setting	Head	Medium	Tail	Overall
Frozen linear probe	80.3 ± 0.2	76.0 ± 0.2	71.7 ± 0.3	77.1 ± 0.1
Same adapter scaffold, without U-PLA	80.0 ± 0.2	76.5 ± 0.2	72.8 ± 0.3	77.5 ± 0.1
Full U-MOF	79.7 ± 0.2	77.2 ± 0.2	74.2 ± 0.3	78.1 ± 0.1

experiments. Unless otherwise specified, these measurements are obtained on an NVIDIA RTX 4090 GPU.

The parameter and MAC measurements characterize the inference-side stage-two footprint, while Table 22 reports training-time overhead. U-MOF requires objective-specific gradients during stage-two training, so its wall-clock cost can be higher than a classifier-only stage-two update. However, this overhead is incurred only during training and is restricted to the lightweight adapter-head parameter set. At inference time, routing and projection are removed, and U-MOF uses one shared frozen-backbone forward followed by lightweight adapter-head evaluations and uniform committee averaging.

F.2 Detailed Network Architecture Configurations

A central design choice of U-MOF is to equip each expert with a lightweight residual adapter while keeping the pretrained backbone frozen. In our implementation, the adapter operates on the pooled backbone descriptor rather than on a separate replicated backbone.

Backbone feature tensor. Let

$$v \in \mathbb{R}^{B\times d} \tag{77}$$

denote the pooled feature descriptor extracted from the final layer of the frozen backbone, where B is the mini-batch size and d is the channel dimension. In our experiments, $d = 512$ for ResNet-34 and $d = 2048$ for ResNet-50. The backbone is frozen throughout stage two, and gradients are computed only for the expert adapters and classifier heads.

Channel-wise residual adapter. Rather than applying spatial convolutions, the expert adapter operates directly on the pooled descriptor. For expert e, a bottleneck multilayer perceptron produces a channel-wise gate:

$$g_e = \mathrm{Sigmoid}\left(W_e^{\mathrm{up}}\,\mathrm{ReLU}\left(W_e^{\mathrm{down}}v + b_e^{\mathrm{down}}\right) + b_e^{\mathrm{up}}\right), \tag{78}$$

where

$$W_e^{\mathrm{down}} \in \mathbb{R}^{(d/r)\times d}, \qquad W_e^{\mathrm{up}} \in \mathbb{R}^{d\times(d/r)}. \tag{79}$$

The bias terms satisfy $b_e^{\mathrm{down}} \in \mathbb{R}^{d/r}$ and $b_e^{\mathrm{up}} \in \mathbb{R}^d$. Unless otherwise specified, the bottleneck reduction ratio is fixed to $r = 16$ in all experiments.

To make the residual branch start from a near-identity transformation, we use the centered modulation form:

$$\tilde{z}_e = v + \alpha_e v \odot \left(g_e - \frac{1}{2}\right), \tag{80}$$

Table 20: Shared structural defaults for the ResNet benchmark experiments. These constants are fixed across repeated runs; objective weights remain batch-adaptive. Benchmark-specific training schedules are separate from these structural defaults.

Setting	Default
Experts E; bottleneck ratio r	3; 16
Residual scale α_e	Fixed 0.10
Routing sensitivities $(\beta_m, \beta_s, \beta_d)$	$(3, 2, 0)$
Mining objectives and within-group allocation	KPS and BCL; equal priors
Smoothing objective	LOS
Diversity priors $(\pi_{\text{feat}}, \pi_{\text{pred}})$	$(0.10, 0.05)$
Nonlinearity	ReLU
Trainable modules in stage two	Expert adapters and classifier heads
Stage-two optimizer	SGD
Classification-objective ordering	Mining objectives before smoothing
Inference	Uniform average of expert probabilities
Statistical reporting	Five-run mean $\pm$ standard deviation

Table 21: Measured parameter and MAC statistics of U-MOF on CIFAR-100-LT with ResNet-34 and $E = 3$. The additional adapter-head branches introduce negligible inference-side computational overhead relative to the shared frozen encoder.

Quantity	Measured Value	Remark
Encoder parameters	21.277M	Frozen in stage two
Adapter parameters	98.304K	Three expert adapters
Classifier-head parameters	153.900K	Three expert heads
Optimized stage-two parameters	**252.204K**	Adapters + heads
Encoder MACs	1.163G	Shared backbone forward
Extra adapter-head MACs	**251.904K**	Three expert branches
Full U-MOF inference MACs	1.164G	Encoder + adapter-head branches
Extra MAC ratio	**0.0217%**	Branch MACs / encoder MACs

where $\odot$ denotes element-wise multiplication, and α_e is a residual scaling coefficient. The classifier head of expert e operates on the adapted feature:

$$s_e = W_e^{\text{cls}} \tilde{z}_e + b_e^{\text{cls}}, \qquad p_e(\cdot \mid x) = \text{softmax}(s_e). \tag{81}$$

Initialization strategy. To stabilize early stage-two optimization, we initialize the final expansion layer of each adapter near zero. With the centered form in Eq. (80), this makes $g_e \approx 1/2$ at initialization and therefore yields $\tilde{z}_e \approx v$. This keeps the model close to the pretrained frozen-backbone behavior while allowing gradients to flow through the adapter branch, preventing abrupt feature drift during the early iterations.

F.3 Sensitivity of the U-PLA Exponential Mapping

We vary (β_m, β_s) on CIFAR-100-LT with IR $= 100$ to evaluate the robustness of the exponential mapping.

F.4 Sensitivity of Diversity Regularization

We vary the intra-group prior weights $(\pi_{\text{feat}}, \pi_{\text{pred}})$ on CIFAR-100-LT with IR $= 100$.

Under Eq. (49), only the relative intra-group priors affect the normalized allocation: multiplying both priors by the same positive constant leaves the coefficients unchanged. In particular, $(0.10, 0.05)$ and $(0.20, 0.10)$ specify the same 2:1 normalized allocation, not different overall diversity strengths. The nominal sweep therefore concerns feature-versus-prediction allocation; absolute group mass is controlled by the group-level score. The listed results should not be used to infer a stronger-diversity effect merely from the common scaling of both priors.

Table 22: Measured wall-clock and peak-memory overhead on CIFAR-100-LT with ResNet-34 and $E = 3$. All stage-two configurations use the same frozen stage-one backbone. Wall-clock time is measured for the full stage-two training run under the controlled experimental setting.

Configuration	Trainable Parameters	Stage-Two Time	Peak Memory	Notes
Classifier-only stage two	153.900K	0.18 GPU-hours	1.1 GB	Three classifier heads without adapter branches
TS-MOF matched scaffold	252.204K	0.19 GPU-hours	1.3 GB	Same objective pool and expert scaffold as U-MOF
U-MOF	252.204K	0.22 GPU-hours	1.4 GB	U-PLA routing + classification-first projection

Table 23: Sensitivity analysis of the U-PLA exponential mapping coefficients (β_m, β_s) on CIFAR-100-LT with IR $= 100$. Results are reported as mean $\pm$ standard deviation over five independent runs.

Disagreement Coeff. (β_m)	Entropy Coeff. (β_s)	Head	Medium	Tail	Overall (%)
1.0	1.0	**77.5** $\pm$0.3	50.8 $\pm$0.3	40.5 $\pm$0.4	57.1 $\pm$0.3
2.0	1.0	77.2 $\pm$0.2	51.1 $\pm$0.3	41.2 $\pm$0.5	57.3 $\pm$0.2
3.0 *(Default)*	2.0 *(Default)*	76.8 $\pm$0.3	51.4 $\pm$0.3	**41.9** $\pm$0.4	**57.5** $\pm$0.2
4.0	2.0	76.2 $\pm$0.3	51.5 $\pm$0.4	41.8 $\pm$0.5	57.4 $\pm$0.2
5.0	3.0	75.8 $\pm$0.4	**51.6** $\pm$0.3	41.6 $\pm$0.5	57.2 $\pm$0.3

F.5 Sensitivity to the Number of Experts

We evaluate the sensitivity of U-MOF to different expert numbers. Increasing E mainly affects the lightweight adapter-head branches and the quality of committee diagnostics.

The $E = 1$ row is a non-committee control: cross-expert disagreement and pairwise diversity do not have their multi-expert meaning, and division by $\log E$ must not be applied. Among the multi-expert settings, the gain saturates beyond $E = 3$, while branch MACs grow approximately linearly. The default $E = 3$ balances committee diagnostics and branch cost.

F.6 Adapter Design and Initialization

We ablate the residual adapter design to evaluate whether the centered modulation is necessary.

These results support the use of the centered residual form, which starts from a near-identity transformation and avoids abrupt perturbation of the frozen representation. The uncentered modulation variant performs worse, suggesting that uncontrolled feature scaling can slightly degrade the stage-two convergence trajectory.

F.7 Reproducibility Notes

Backbones. We use ResNet-34 on CIFAR-100-LT and ResNet-50 on ImageNet-LT and iNaturalist 2018. The backbone is trained during stage one and frozen during stage two.

Two-stage training. Stage two uses the same long-tailed training split. No test-set labels are used for training, validation, or hyperparameter selection. The second stage is run for 20 to 30 epochs depending on the dataset.

Optimizer. Stage-two optimization uses SGD. Objective-specific gradients are computed with respect to the trainable stage-two parameter set Ω, routed by U-PLA, and coordinated using classification-first projection.

Routing coefficients and objective ordering. The default U-PLA routing coefficients are $\beta_m = 3.0$ and $\beta_s = 2.0$. Tail-oriented objectives are processed before the smoothing-oriented objective in the projection sequence. This deterministic ordering is kept fixed across runs.

Statistical reporting. Unless otherwise specified, U-MOF results and controlled variants with uncertainty ranges are reported as mean $\pm$ standard deviation over five independent runs. Baseline values without uncertainty ranges are quoted from the corresponding papers or measured once under the matched protocol. When a result is obtained from a single controlled run, we explicitly indicate this in the corresponding caption and omit the uncertainty range.

Table 24: Sensitivity analysis of the diversity intra-group prior weights $(\pi_{\text{feat}}, \pi_{\text{pred}})$ on CIFAR-100-LT with IR $= 100$. Results are reported as mean $\pm$ standard deviation over five independent runs.

Feature Diversity Prior (π_{feat})	Prediction Diversity Prior (π_{pred})	Head	Medium	Tail	Overall (%)
0.01	0.05	**77.9** ± 0.3	50.4 ± 0.4	40.1 ± 0.5	56.9 ± 0.3
0.05	0.05	77.4 ± 0.2	51.0 ± 0.3	41.3 ± 0.4	57.3 ± 0.2
0.10 *(Default)*	0.05 *(Default)*	76.8 ± 0.3	**51.4** ± 0.3	**41.9** ± 0.4	**57.5** ± 0.2
0.20	0.10	76.3 ± 0.3	51.2 ± 0.3	41.8 ± 0.5	57.2 ± 0.2
0.50	0.20	75.1 ± 0.4	50.9 ± 0.4	41.0 ± 0.6	56.5 ± 0.3

Table 25: Sensitivity of computational overhead and performance to the number of experts E on CIFAR-100-LT with IR $= 100$. Extra Branch MACs measure only the adapter-head branches and exclude the shared frozen-backbone forward. Routing and projection are training-time mechanisms and are not used for inference.

Number of Experts E	Extra Branch MACs	Head	Medium	Tail	Overall
1	83.968K	**78.2** ± 0.3	48.5 ± 0.4	35.6 ± 0.5	55.3 ± 0.2
2	167.936K	77.4 ± 0.3	50.2 ± 0.3	39.5 ± 0.4	56.5 ± 0.2
3 *(Default)*	251.904K	76.8 ± 0.3	51.4 ± 0.3	41.9 ± 0.4	57.5 ± 0.2
4	335.872K	76.6 ± 0.2	51.6 ± 0.3	42.3 ± 0.5	57.6 ± 0.2
5	419.840K	76.5 ± 0.3	**51.7** ± 0.3	**42.4** ± 0.4	**57.6** ± 0.2

Computational measurements. Wall-clock and peak-memory measurements are recorded under the same hardware and data-loading configuration used for the controlled CIFAR-100-LT experiments. Stage-two time measures the full second-stage training run after the backbone has been pretrained and frozen. Peak memory is measured during the stage-two training loop and includes the forward/backward passes required by the corresponding objective configuration. Inference-side MACs are reported separately from training-time gradient-computation overhead because routing and projection are not executed at test time.

F.8 Residual-Scale Sensitivity

The residual scale controls how strongly the centered gate changes the frozen descriptor. The shared default is a fixed $\alpha_e = 0.10$. The controlled CIFAR-100-LT IR=100 sweep below compares fixed scales and a learnable variant without changing the objective pool or committee structure.

The results are relatively stable around the default. A substantially larger fixed scale lowers the mean, while the learnable variant does not improve on the fixed default in this setting. This supports a conservative residual magnitude, not a claim that the same scale is optimal for arbitrary frozen representations.

G Limitations and Broader Impacts

This appendix concludes by discussing the main limitations of U-MOF and its broader implications. Our goal is to state these points transparently and conservatively, so that the scope of the proposed framework is clear. U-MOF is designed as a practical parameter-efficient stage-two optimization framework for long-tailed recognition, focusing on efficient adaptation under data imbalance.

G.1 Limitations

Representation and task assumptions. U-MOF targets closed-set long-tailed image classification and assumes a usable frozen stage-one representation. Shallow channel-wise adapters can reweight existing information, but may be insufficient when the backbone contains almost no discriminative information for a rare class. The design does not establish robustness to open-set classes, continual changes, multi-label prediction, or distribution shift.

Diagnostic scope and batch dependence. Disagreement and within-expert entropy are operational allocation proxies rather than calibrated Bayesian uncertainty estimates. High entropy can reflect hard rare examples as well as ambiguous observations. Small or non-IID mini-batches may yield

Table 26: Adapter design ablation on CIFAR-100-LT with IR $= 100$. The centered residual adapter is compared with classifier-only tuning and uncentered modulation.

Adapter Variant	Final Head	Final Med	Final Tail	Final Overall
Classifier-Only *(No Adapter)*	**77.9** ± 0.3	49.5 ± 0.3	36.8 ± 0.5	55.8 ± 0.3
Uncentered Gate $\tilde{z}_e = v + v \odot g_e$	77.1 ± 0.4	50.4 ± 0.4	39.8 ± 0.5	56.6 ± 0.3
Centered Gate without Res-Scale ($\alpha_e = 1$)	76.6 ± 0.3	50.8 ± 0.3	41.1 ± 0.4	57.1 ± 0.2
Centered Gate with Fixed α_e *(Default)*	76.8 ± 0.3	**51.4** ± 0.3	**41.9** ± 0.4	**57.5** ± 0.2
Centered Gate with Learnable α_e	76.9 ± 0.3	51.2 ± 0.4	41.8 ± 0.5	57.4 ± 0.3

Table 27: Sensitivity to the residual scale on CIFAR-100-LT with IR=100. Overall accuracy is reported as mean $\pm$ standard deviation over five runs.

Scale	0.05	0.10 (default)	0.20	0.50	Learnable
Overall	57.2 $\pm$ 0.2	57.5 $\pm$ 0.2	57.4 $\pm$ 0.2	57.0 $\pm$ 0.3	57.4 $\pm$ 0.3

noisy diagnostics, and experts sharing one frozen encoder can agree on the same error. The $E = 1$ setting is a non-committee reference. The routing controls and one-step intervention audit support the evaluated allocation rule empirically, not its universal optimality.

Manual objective structure and numerical coordination. The objective pool and its mining, smoothing, and diversity grouping are specified manually. Structural defaults are shared in the reported experiments, but new domains can require different objectives or operating points. Classification-first projection controls conflicts relative to a running aggregate; its stabilized form does not certify simultaneous descent of all objectives, Pareto optimality, or monotonic evaluation accuracy. The two-scale diversity-prior sweep must also be interpreted through within-group normalization rather than absolute nominal prior size.

Accuracy trade-offs and computational cost. Better Tail and Overall accuracy need not improve every frequency group. The matched CIFAR-100-LT comparison trades some Head accuracy for Medium/Tail recovery, and the iNaturalist 2018 comparison lowers Medium accuracy. Stage-two training requires objective-specific gradients and adds measured runtime and memory relative to the matched engine. At inference, the backbone is shared, but all lightweight expert branches are evaluated; routing-free inference does not mean zero added computation.

Evaluation coverage. The main benchmark suite is ResNet-based. The separate frozen CLIP ViT-B/16 experiment establishes pooled-embedding transfer only for that setting. Broader vision-transformer and CLIP variants, layer-wise adapter placement, and general foundation-model fine-tuning remain untested. Detailed routing, placement, and representation audits concentrate on CIFAR-100-LT; matched large-scale engine comparisons do not supply equally detailed one-factor evidence for every component on every dataset.

G.2 Broader Impacts

Long-tailed recognition methods can have positive practical impacts in settings where rare categories are important. Many real-world datasets are naturally imbalanced, and rare cases may carry high value. Examples include medical image analysis, biodiversity monitoring, anomaly detection, industrial inspection, and safety-critical perception. In such applications, improving rare-category recognition may help reduce the majority-class bias that often arises from standard empirical risk minimization. By focusing on parameter-efficient stage-two adaptation, U-MOF may also reduce the cost of improving long-tailed models when full retraining is expensive.

At the same time, improved tail recognition does not automatically imply fairness, robustness, or safety. A method may improve accuracy on statistically rare classes while still inheriting biases from the training data, annotation process, or pretrained backbone. Because U-MOF freezes the stage-one backbone and only adapts lightweight stage-two modules, the method builds upon the representation learned by the pretrained backbone, and therefore shares its underlying inductive biases. Therefore, U-MOF should not be interpreted as a standalone solution to dataset bias, demographic imbalance, or

representational inequity. Careful dataset auditing, subgroup evaluation, calibration analysis, and deployment-specific risk assessment remain necessary.

There are also potential risks if long-tailed recognition systems are deployed in high-stakes settings without sufficient validation. Better recognition of rare categories can be beneficial, but it can also increase confidence in predictions for cases where the model remains uncertain or biased. In medical, legal, surveillance, or safety-critical applications, errors on rare categories may have serious consequences. For this reason, methods like U-MOF should be used with transparent reporting of limitations, uncertainty behavior, failure modes, and domain-specific validation results.

More broadly, U-MOF contributes to the development of efficient adaptation mechanisms for imbalanced visual recognition. We hope that this line of work encourages future research that not only improves benchmark accuracy, but also studies calibration, fairness, robustness, and safety under realistic long-tailed distributions. Improvements in rare-category recognition should be paired with responsible data governance, careful evaluation protocols, and human oversight when deployed in consequential applications.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We list our contributions in detail at the end of the Introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the relevant assumptions and proofs in Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the code source in the abstract and the supplementary boundary in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the source of the dataset and code in Appendix.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We describe the relevant details in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report performance metrics such as average accuracy on multiple datasets.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We mention the computational resources used for the experiments in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read and understood the code of ethics; and have done our best to conform.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix G.2 discusses potential positive impacts of improving rare-category recognition and efficient adaptation, as well as risks related to inherited dataset or backbone bias, overconfidence in high-stakes deployments, and the need for calibration, fairness, robustness, and human oversight.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work does not release high-risk models, scraped datasets, or assets requiring special safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We are not shipping our code with any source code or binary files from any other existing libraries, so there are no concerns over getting permission or including a license. We did cite open-sourced libraries in our paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We will release our code base with included readme files. We do not ship any source code or binary files from any other existing libraries.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Our work does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.