FEATURE SUPPORT EROSION IN LONG-TAIL FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained models provide strong prior knowledge for long-tailed recognition, making fine-tuning a promising alternative to training from scratch. However, we find that tail classes with similar training frequencies can behave very differently during fine-tuning: some remain stable, while others suffer severe forgetting. This suggests that class frequency alone is insufficient to explain tail-class stability. Our analysis reveals a key factor behind this difference: *Feature Support Erosion*, where pretrained discriminative directions of different classes receive unequal support from the dominant training signals. To characterize this effect, we use class-wise Average Gradient Outer Product (AGOP) to identify pretrained discriminative directions and measure their support from head-class training signals. Based on this analysis, we propose **S**upport-**A**ware **D**irectional **P**reservation (SADP), which assigns stronger preservation to poorly supported classes and selectively constrains lightweight feature updates along their pretrained discriminative directions. Extensive experiments demonstrate that SADP consistently outperforms strong pretrained fine-tuning methods, effectively reducing tail-class forgetting while preserving the flexibility of downstream fine-tuning.

1 INTRODUCTION

Long-tail learning aims to address class imbalance, where a few head classes have many training samples, while most tail classes have limited data. Models trained on such imbalanced data tend to favor head classes and perform poorly on tail classes. To address this problem, previous studies have explored several approaches, including class re-balancing through re-sampling (Chawla et al., 2002) and re-weighting (Cui et al., 2019; Cao et al., 2019), decoupled representation and classifier learning (Kang et al., 2020; Zhou et al., 2020), and logit adjustment based on class frequencies (Menon et al., 2021). Despite these advances, learning from limited tail-class data remains a major challenge in long-tail recognition (Zhang et al., 2023).

Recently, the success of large-scale pretrained models has opened up new opportunities for long-tail learning. Instead of training models from scratch, several studies have explored how to effectively fine-tune pretrained models on long-tail datasets. For example, BALLAD (Ma et al., 2021) adopts a two-stage training strategy that combines fine-tuning with a lightweight adapter. VL-LTR (Tian et al., 2022) introduces additional textual knowledge to improve visual representations, while RAC (Long et al., 2022) uses external data retrieval to enhance recognition. LPT (Dong et al., 2022) explores prompt tuning for long-tail learning. More recently, LIFT (Shi et al., 2024) shows that extensive fine-tuning can hurt tail-class performance, while lightweight fine-tuning can better preserve pretrained knowledge with fewer trainable parameters, and SRCD (Wan et al., 2026) leverages pretrained knowledge through shortcut-resistant CAM distillation to improve tail-class recognition.

While these methods have shown the potential of pretrained models for long-tail learning, why different tail classes behave differently during fine-tuning remains unclear. As shown in Figure 1, tail classes with similar training frequencies can exhibit remarkably different performance changes after fine-tuning. Some tail classes maintain or even improve their original recognition performance, which we refer to as *Stable Tails*. In contrast, other tail classes suffer severe performance drops and may even lose the recognition ability already acquired during pretraining. We refer to these classes as *Forgotten Tails*. Notably, such contrasting behaviors can occur among classes with comparable numbers of training samples. This observation suggests that class frequency alone cannot fully

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

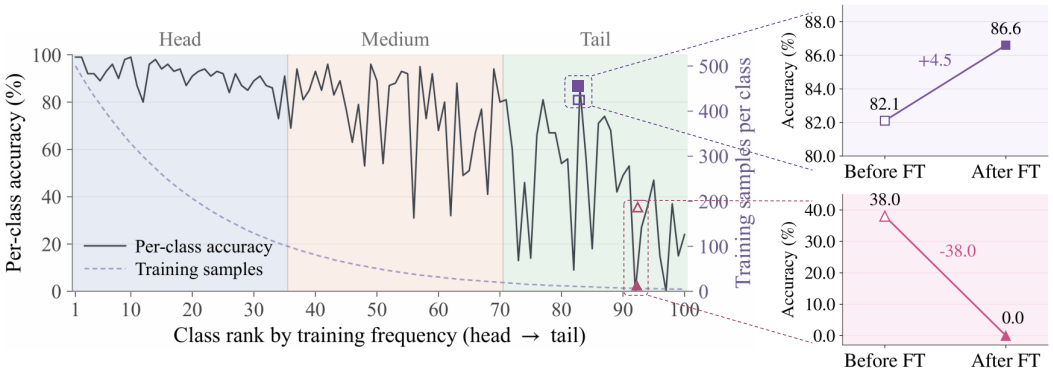


Figure 1: Tail classes with similar frequencies can behave very differently after fine-tuning. Some tail classes remain stable or improve, while others suffer severe forgetting despite having similar numbers of training samples.

explain the behavior of tail classes during fine-tuning. Beyond learning from limited data, an equally important challenge is therefore to understand why pretrained knowledge is preserved for some tail classes but forgotten for others.

To understand this phenomenon, we study long-tail fine-tuning from the perspective of feature learning. Since head classes dominate the training data, their gradients have a strong influence on the learned representations. However, different tail classes may receive different levels of support from these training signals. Stable Tails tend to rely on pretrained feature directions that are also covered by head-class training signals. As a result, these directions can receive continued support during fine-tuning, even when the classes themselves contain only a few samples. In contrast, the pretrained discriminative directions of Forgotten Tails receive less support from the dominant training signals. These directions are therefore more vulnerable to feature updates and may gradually lose their original discriminative structure. We refer to this phenomenon as *Feature Support Erosion*. This view suggests that the stability of a tail class depends not only on how many training samples it has, but also on how well its pretrained discriminative features are supported during fine-tuning.

Motivated by this observation, we propose **S**upport-**A**ware **D**irectional **P**reservation (SADP), which selectively preserves pretrained features according to their support during long-tail fine-tuning. Inspired by the Average Gradient Outer Product (AGOP) (Radhakrishnan et al., 2024), we first identify the key discriminative directions of each class in the pretrained model. We then measure how strongly these directions overlap with the dominant training signals to estimate class-level feature support. Feature support determines *how strongly* a class should be protected, while its pretrained AGOP determines *which feature directions* should be preserved. Accordingly, SADP assigns stronger preservation to poorly supported classes and selectively constrains feature updates along their pretrained discriminative directions. Rather than restricting the entire feature space, the remaining directions are left free to learn downstream information. In this way, SADP reduces destructive changes to vulnerable pretrained features while retaining the flexibility required for long-tail fine-tuning.

Our main contributions are summarized as follows:

- We reveal a clear class-level difference in tail-class behavior during long-tail fine-tuning. Even with similar training frequencies, some tail classes remain stable while others suffer severe forgetting, showing that class frequency alone is insufficient to explain their performance changes.
- We introduce *Feature Support Erosion* to explain this phenomenon. Using class-wise AGOP, we characterize pretrained discriminative directions and show that their support from dominant training signals is closely related to feature retention and tail-class forgetting.
- We propose Support-Aware Directional Preservation (SADP), which selectively preserves poorly supported pretrained discriminative directions. We theoretically show that SADP provides support-dependent control of feature drift, while extensive experiments demonstrate consistent improvements over strong long-tailed learning and pretrained fine-tuning methods.

2 Related Work

Long-Tailed Learning. Long-tailed Learning has been studied from multiple perspectives, including loss re-weighting, data augmentation, representation learning, and classifier regularization. Representative training-from-scratch methods include MetaSAug (Li et al., 2021), CMO (Park et al., 2022), SAFA (Hong et al., 2022), GML (Du & Wu, 2023), ConCutMix (Pan et al., 2024), LOS (Sun et al., 2025), and CAR (Zhu et al., 2026). With the emergence of foundation models, recent studies have increasingly focused on transferring pretrained knowledge to long-tailed recognition: BALLAD (Ma et al., 2021) performs balanced fine-tuning with pretrained vision-language models, LPT (Dong et al., 2022) introduces parameter-efficient prompt tuning, Decoder (Wang et al., 2024) explores lightweight decoding for imbalanced recognition, PECEL (Ru et al., 2024) employs complementary parameter-efficient experts, LIFT (Shi et al., 2024) shows that lightweight fine-tuning can better preserve tail-class performance than heavy fine-tuning, and SRCD (Wan et al., 2026) alleviates tail-class shortcut learning through CAM distillation. Meanwhile, several works have shown that class frequency alone is insufficient to characterize long-tailed learning. CDB (Sinha et al., 2022) and Difficulty-Net (Sinha & Ohashi, 2023) model class difficulty beyond sample frequency, ReCT (Li et al., 2024) further considers class correlation and representation bias, and DBM (Son et al., 2025) incorporates instance-level difficulty into long-tailed optimization. Different from these methods, our work does not focus on estimating which classes are harder to learn, but on understanding why pretrained class-specific features are preserved for some tail classes while forgotten for others during fine-tuning. We attribute this difference to feature support and use class-wise AGOP to characterize how strongly pretrained discriminative directions are covered by dominant training signals. Detailed descriptions of the compared methods are provided in Appendix B.

3 Preliminaries

3.1 Long-Tailed Learning

Long-tailed learning aims to address the class imbalance problem, where different classes have highly unequal numbers of training samples. Formally, let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ denote a training dataset containing N samples from K classes, where x_i is an input image and $y_i \in \{1, \dots, K\}$ is its class label. Let n_c denote the number of training samples belonging to class c. Without loss of generality, we assume that the classes are sorted in descending order of frequency, i.e., $n_1 \geq n_2 \geq \cdots \geq n_K$. The degree of class imbalance is commonly measured by the imbalance factor $\mathrm{IF} = n_1/n_K$. Following common practice, we divide the classes into three groups according to their training frequencies: head classes $\mathcal{H}$, medium classes $\mathcal{M}$, and tail classes $\mathcal{T}$, where $n_c > 100$, $20 < n_c \leq 100$, and $n_c \leq 20$, respectively (Kang et al., 2020). Head classes contain abundant training samples, whereas tail classes have only limited supervision. As a result, models trained on long-tailed data tend to favor head classes and perform poorly on tail classes. The goal of long-tailed learning is to learn a classifier that achieves strong recognition performance across all classes despite the highly imbalanced training distribution (Zhang et al., 2023).

3.2 Revisiting Average Gradient Outer Product

Average Gradient Outer Product (AGOP) (Radhakrishnan et al., 2024) provides a gradient-based tool for studying feature learning in neural networks. Its main idea is that the gradients of model outputs with respect to their inputs reveal the directions to which the model is most sensitive. By aggregating these gradients across multiple samples, AGOP captures the dominant directions of the model's input-output sensitivity. Formally, consider a scalar-valued function $f : \mathbb{R}^d \to \mathbb{R}$ and a set of representations $\{h_i\}_{i=1}^{N}$, where $h_i \in \mathbb{R}^d$. The AGOP matrix is defined as

$$A_f = \frac{1}{N} \sum_{i=1}^{N} \nabla_h f(h_i) \nabla_h f(h_i)^\top,$$

where $\nabla_h f(h_i) \in \mathbb{R}^d$ is the gradient of the model output with respect to the input representation h_i. Each outer product describes the local sensitivity of the model to changes in its input representation, while their average summarizes this sensitivity across the dataset. For a vector-valued function, AGOP can be extended by summing the gradient outer products of its individual output components.

The AGOP matrix is symmetric and positive semidefinite. For any unit vector $v \in \mathbb{R}^d$, we have

$$v^\top A_f v = \frac{1}{N} \sum_{i=1}^{N} \left(v^\top \nabla_h f(h_i) \right)^2 .$$

This quantity measures the average squared sensitivity of the model output along direction v. A larger value indicates that changes along this direction have a stronger local influence on the model output. By performing eigendecomposition $A_f = U \Lambda U^\top$, we can identify the dominant sensitivity directions through the eigenvectors associated with the largest eigenvalues. Previous work has shown that AGOP is closely related to the feature learning process in neural networks (Radhakrishnan et al., 2024). Motivated by this connection, we apply AGOP to intermediate visual representations to characterize class-specific discriminative directions and study how these directions relate to the training signals produced during long-tailed fine-tuning.

4 METHODOLOGY

4.1 ANALYZING FEATURE SUPPORT EROSION

We investigate why tail classes with similar training frequencies can behave differently during fine-tuning. We refer to classes that remain stable as *Stable Tails* and those that suffer substantial degradation as *Forgotten Tails*. We hypothesize that this difference arises from how well their pretrained discriminative features are supported during fine-tuning. To examine this, we analyze feature support from dominant training signals, feature retention after fine-tuning, and the resulting change in pretrained recognition ability.

Do pretrained tail features receive sufficient training support? We first examine whether the discriminative features of different tail classes are equally supported during fine-tuning. Let $h(x) \in \mathbb{R}^d$ denote an intermediate visual representation. For a class c, we use the gradient of its classification margin, $m_c(x) = z_c(x) - \max_{j \neq c} z_j(x)$, with respect to $h(x)$ to construct a class-specific AGOP on the pretrained model:

$$A_c^{\mathrm{pre}} = \frac{1}{|\mathcal{D}_c|} \sum_{x \in \mathcal{D}_c} g_c(x) g_c(x)^\top, \qquad g_c(x) = \nabla_{h(x)} m_c(x).$$

Here, A_c^{pre} summarizes the feature directions to which the prediction of class c is most sensitive in the pretrained model. We eigendecompose A_c^{pre} and retain its dominant eigenvectors as U_c^{pre}. These directions form the main discriminative subspace of class c. We further define $P_c^{\mathrm{pre}} = U_c^{\mathrm{pre}} (U_c^{\mathrm{pre}})^\top$, which is simply the projection matrix onto this subspace.

Since head classes contribute most of the training samples, we use their gradients to characterize the dominant optimization signal. At epoch t, let B_{head}^t denote the AGOP constructed from the training-loss gradients of head-class samples. We define the *Head Feature Support* of class c as

$$S_c^{\mathrm{head}}(t) = \frac{\mathrm{Tr}(P_c^{\mathrm{pre}} B_{\mathrm{head}}^t)}{\mathrm{Tr}(B_{\mathrm{head}}^t)}.$$

Intuitively, $S_c^{\mathrm{head}}(t)$ measures how much of the head-class training signal falls into the pretrained discriminative subspace of class c. If the value is high, the features used by this tail class are also frequently involved in head-class optimization and can therefore receive training support even when the class itself has few samples. As shown in Fig. 2 *left*, Stable Tails consistently receive much stronger head feature support than Forgotten Tails. In contrast, the pretrained directions of Forgotten Tails remain weakly covered throughout fine-tuning. This result suggests that tail classes with similar frequencies can experience substantially different feature-level training environments.

Are unsupported pretrained features actually lost? A low support score alone does not imply that the pretrained representation has changed. We therefore next examine whether the pretrained discriminative directions are actually retained during fine-tuning. At epoch t, we compute the current class-specific AGOP A_c^t in the same way as A_c^{pre}. While A_c^{pre} describes which directions class c relies on before fine-tuning, A_c^t describes its current discriminative structure after the model has

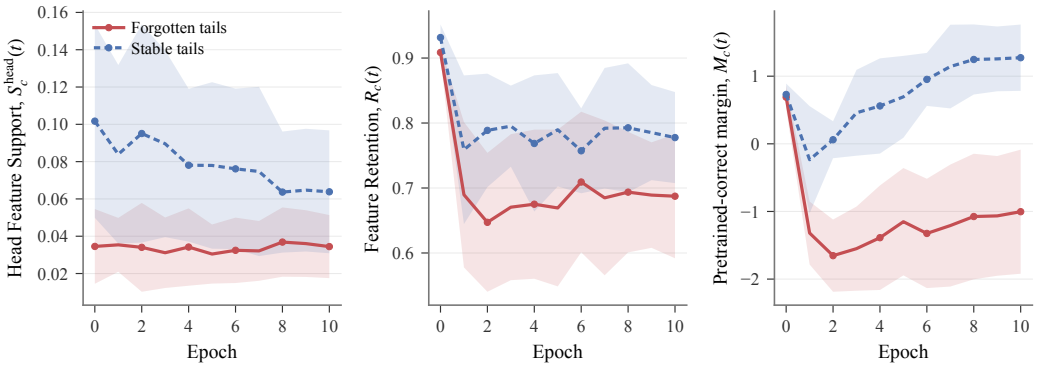


Figure 2: **Feature Support Erosion during long-tailed fine-tuning.** We compare Forgotten Tails and Stable Tails throughout training. *left*: Stable Tails consistently receive stronger support from head-class training signals. *middle*: Although both groups start with similar feature retention, Forgotten Tails lose more of their pretrained discriminative structure after fine-tuning begins. *right*: The pretrained-correct margin of Stable Tails gradually recovers, while that of Forgotten Tails remains negative. The shaded regions indicate the variation across classes.

been updated. We measure their relation by projecting the current AGOP onto the fixed pretrained subspace:

$$R_c(t) = \frac{\mathrm{Tr}(P_c^{\mathrm{pre}} A_c^t)}{\mathrm{Tr}(A_c^t)}.$$

We call $R_c(t)$ *Feature Retention*. A high value means that the current class prediction still relies strongly on the directions that were important in the pretrained model, whereas a low value indicates that its discriminative structure has moved away from the pretrained subspace. Fig. 2 *middle* shows that Stable and Forgotten Tails have similar retention before fine-tuning, but quickly diverge once training begins. Stable Tails maintain a substantially larger portion of their pretrained feature structure, while Forgotten Tails exhibit a much stronger decrease. Together with Fig. 2 *left*, this observation suggests that classes receiving weaker external support are also more likely to lose their pretrained discriminative directions.

Does feature erosion lead to forgetting? Finally, we examine whether the loss of pretrained feature structure is reflected in recognition behavior. Class accuracy alone mixes newly learned samples with previously recognized samples, making it difficult to isolate forgetting. We therefore focus on samples that are correctly classified by the pretrained model. For class c, let $\mathcal{K}_c = \{x \mid y = c, \hat{y}_{\mathrm{pre}}(x) = c\}$. We track their average classification margin during fine-tuning:

$$M_c(t) = \frac{1}{|\mathcal{K}_c|} \sum_{x \in \mathcal{K}_c} \left[z_c^t(x) - \max_{j \neq c} z_j^t(x) \right].$$

We refer to this quantity as the *Pretrained-Correct Margin*. Unlike the previous two measures, which describe the feature space, this metric directly measures whether examples that the pretrained model could already recognize remain correctly separated from competing classes. As shown in Fig. 2 *right*, both groups experience an initial decrease in margin after fine-tuning starts. However, Stable Tails gradually recover their pretrained discrimination, whereas the margin of Forgotten Tails remains low and becomes negative. This shows that the difference observed in feature retention is also reflected at the prediction level: Forgotten Tails not only change their feature structure, but also lose recognition ability that was already present before fine-tuning.

Overall, the above observations provide a consistent explanation for the different behaviors of tail classes. Stable Tails receive stronger support from the dominant training signals and better preserve their pretrained discriminative structure, allowing their recognition ability to recover during fine-tuning. Forgotten Tails, in contrast, receive much weaker support, exhibit stronger feature drift, and fail to retain their pretrained recognition ability. We refer to this phenomenon as **Feature Support Erosion**. These findings motivate a support-aware preservation strategy: classes with insufficient feature support should receive stronger protection, while the class-specific AGOP provides the directions that should be preserved.

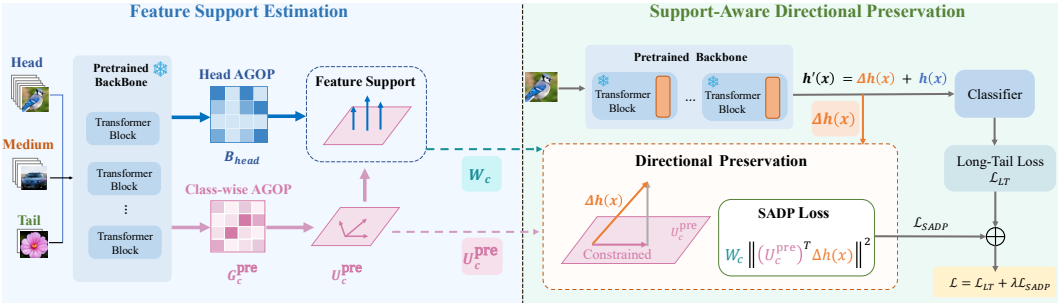


Figure 3: Overview of SADP. We estimate class-specific feature support from pretrained and head-class AGOPs, and use it to selectively preserve poorly supported discriminative directions during lightweight fine-tuning.

4.2 Support-Aware Directional Preservation

The above analysis suggests that tail classes with insufficient feature support are more likely to lose their pretrained discriminative structure during fine-tuning. Based on this observation, we propose **S**upport-**A**ware **D**irectional **P**reservation (SADP). As illustrated in Figure 3, SADP consists of feature support estimation and support-aware directional preservation. The key idea is simple: feature support determines *how strongly* a class should be protected, while its pretrained AGOP determines *which feature directions* should be preserved.

Support-Aware Preservation Strength. For each tail class c, we use the pretrained discriminative subspace U_c^{pre} obtained from its class-wise AGOP. Although tail classes contain limited samples, SADP only relies on the leading low-dimensional subspace rather than an accurate estimate of the full AGOP matrix. We provide a detailed discussion on the reliability of tail-class AGOP estimation in Appendix C.5. To avoid using any information from the fine-tuned model, we estimate feature support only once before fine-tuning. Specifically, we measure the overlap between the pretrained discriminative subspace of class c and the head-class training gradients:

$$S_c = \frac{\mathrm{Tr}(P_c^{\mathrm{pre}} B_{\mathrm{head}})}{\mathrm{Tr}(B_{\mathrm{head}})},$$

where $P_c^{\mathrm{pre}} = U_c^{\mathrm{pre}} (U_c^{\mathrm{pre}})^\top$ is the projector onto the pretrained discriminative subspace, and B_{head} is the AGOP constructed from head-class training gradients on the pretrained model. To account for the subspace dimension, we further normalize the support as $\bar{S}_c = \frac{d}{k} S_c$, where d is the feature dimension and k is the number of retained AGOP directions.

We then convert feature support into a class-specific preservation weight:

$$w_c = \exp\left(-\frac{\bar{S}_c}{\tau}\right),$$

where τ is a temperature parameter. Therefore, classes with weaker feature support receive stronger preservation, while well-supported classes are less constrained. The weights are normalized to have unit mean over tail classes for stable optimization. Different from frequency-based re-weighting, w_c depends on how well a class-specific feature subspace is supported by the dominant training signals rather than on the number of samples in the class.

Preserving Lightweight Feature Updates. A direct way to preserve pretrained knowledge is to constrain the fine-tuned representation to remain close to the pretrained one. However, this requires an additional pretrained model during training and may unnecessarily restrict the entire feature space. Instead, we build our preservation mechanism directly on lightweight fine-tuning.

Under lightweight fine-tuning, the pretrained backbone remains frozen, while a small trainable module introduces an additional feature update. We write the resulting representation as

$$h'(x) = h(x) + \Delta h(x),$$

where $h(x)$ is the representation produced by the frozen pretrained backbone and $\Delta h(x)$ denotes the modification introduced by the trainable module, such as an adapter (Chen et al., 2022) or LoRA branch (Hu et al., 2021). Since $h(x)$ remains fixed, the feature drift caused by fine-tuning is directly captured by $\Delta h(x)$. Therefore, we can preserve pretrained knowledge without maintaining an additional teacher model.

Instead of uniformly suppressing the whole update $\Delta h(x)$, we only constrain its component along the pretrained discriminative directions of the corresponding class. For a sample (x, c), we define

$$\mathcal{L}_{\mathrm{SADP}}(x, c) = \frac{w_c}{k} \left\| (U_c^{\mathrm{pre}})^\top \Delta h(x) \right\|_2^2 .$$

Here, U_c^{pre} contains the top-k directions of the pretrained class-wise AGOP, and w_c determines the preservation strength according to the feature support of class c. The projected update $(U_c^{\mathrm{pre}})^\top \Delta h(x)$ measures how strongly fine-tuning modifies the pretrained discriminative subspace. SADP therefore suppresses potentially destructive changes along poorly supported directions, while leaving the remaining feature space free for downstream learning.

This design is closely related to the lightweight fine-tuning principle studied in LIFT (Shi et al., 2024). While LIFT shows that restricting the overall extent of fine-tuning can better preserve pretrained knowledge in long-tailed learning, our analysis further reveals that different classes and feature directions require different levels of protection. SADP therefore moves from parameter-level lightweight fine-tuning to class- and direction-aware preservation, controlling not only how much the model is updated, but also where these updates should be restricted.

Training Objective. SADP can be directly added to the original long-tailed fine-tuning objective. For a mini-batch $\mathcal{B}$, the final loss is

$$\mathcal{L} = \mathcal{L}_{\mathrm{LT}} + \lambda \frac{1}{|\mathcal{B}_{\mathcal{T}}|} \sum_{(x_i, y_i) \in \mathcal{B}_{\mathcal{T}}} \mathcal{L}_{\mathrm{SADP}}(x_i, y_i),$$

where $\mathcal{L}_{\mathrm{LT}}$ is the original long-tailed training objective, $\mathcal{B}_{\mathcal{T}}$ denotes the tail-class samples in the mini-batch, and λ controls the preservation strength.

All class-specific AGOP subspaces and support weights are computed once before fine-tuning and remain fixed during training. Therefore, SADP introduces no additional pretrained model and does not require repeatedly computing AGOP during optimization. The only additional information stored for each class is a low-rank subspace U_c^{pre} and a scalar support weight w_c, resulting in a lightweight preservation mechanism that directly targets the feature directions most vulnerable to long-tailed fine-tuning.

We further analyze how SADP controls the change of pretrained discriminative features during fine-tuning. Let $P_c^{\mathrm{pre}} = U_c^{\mathrm{pre}} (U_c^{\mathrm{pre}})^\top$ denote the orthogonal projector onto the pretrained discriminative subspace of class c. The following theorem characterizes the accumulated feature drift under a local feature-space approximation.

Theorem 1 (Support-Aware Feature Drift Bound)**.** *Let h_t denote the feature representation at step t, and let g_t be the gradient of the base long-tailed objective with respect to the local feature update. Under the local quadratic model of SADP, the accumulated drift within the pretrained discriminative subspace after T update steps satisfies*

$$\left\| P_c^{\mathrm{pre}} (h_T - h_0) \right\|_2 \le \sum_{t=0}^{T-1} \frac{\eta_t}{1 + \eta_t \lambda w_c} \left\| P_c^{\mathrm{pre}} g_t \right\|_2 ,$$

where η_t is the local step size. In particular, if $\eta_t = \eta$ and $\|P_c^{\mathrm{pre}} g_t\|_2 \le G_c$, then

$$\left\| P_c^{\mathrm{pre}} (h_T - h_0) \right\|_2 \le \frac{\eta T G_c}{1 + \eta \lambda w_c}.$$

Since w_c is chosen to decrease with feature support S_c, Theorem 1 shows that SADP imposes a tighter control on the pretrained discriminative directions of poorly supported classes. Meanwhile, feature updates outside these directions remain unconstrained under the local model. This provides a theoretical interpretation of the support-aware and directional design of SADP. The detailed proof is provided in Appendix A.

Table 1: Top-1 accuracy (%) comparison on ImageNet-LT and CIFAR100-LT (IF=100) under both the training-from-scratch and fine-tuning settings. Results are reported for Head, Medium, Tail, and Overall. The best results are in **bold**, and the second-best are underlined.

Methods	Venue	ImageNet-LT				CIFAR100-LT			
		Head	Medium	Tail	Overall	Head	Medium	Tail	Overall
Training from scratch									
CE	–	69.71	43.91	16.30	46.51	68.20	45.37	15.17	41.40
Focal (Lin et al., 2017)	ICCV'2017	67.57	49.80	22.83	49.78	66.43	47.49	21.87	43.13
MetaSAug (Li et al., 2021)	CVPR'2021	66.63	47.86	30.57	51.94	67.31	47.91	27.57	45.55
CMO (Park et al., 2022)	CVPR'2022	67.40	50.20	28.97	52.15	68.34	49.37	28.53	46.01
SAFA (Hong et al., 2022)	ECCV'2022	67.20	52.03	32.03	55.29	67.29	54.94	29.47	48.02
GML (Du & Wu, 2023)	CVPR'2023	69.03	53.66	32.17	55.24	68.63	55.20	30.97	50.23
ConCutMix (Pan et al., 2024)	TIP'2024	69.23	48.94	31.00	54.97	67.83	53.54	30.60	47.86
LOS (Sun et al., 2025)	ICLR'2025	70.79	55.31	32.73	56.20	69.21	57.71	29.42	50.85
CAR (Zhu et al., 2026)	CVPR'2026	67.00	54.57	35.77	57.48	65.37	58.60	34.03	51.85
Fine-tuning foundation model									
Decoder (Wang et al., 2024)	IJCV'2024	76.0	72.1	69.3	73.2	83.6	78.5	60.3	74.8
PECEL (Ru et al., 2024)	MM'2024	81.3	77.6	73.0	78.3	84.8	79.1	61.3	75.7
LIFT (Shi et al., 2024)	ICML'2024	79.7	76.2	72.8	77.1	88.3	81.6	67.4	79.7
SRCD (Wan et al., 2026)	ICML'2026	80.4	77.7	75.2	78.4	88.0	82.5	67.9	80.0
CAR (Zhu et al., 2026)	CVPR'2026	81.2	**77.9**	74.6	78.5	88.7	82.3	67.4	80.1
SADP (Ours)	–	**81.6**	77.8	**75.9**	**79.0**	**89.1**	**82.8**	**68.8**	**80.8**

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate SADP on four long-tailed benchmarks: CIFAR100-LT (Krizhevsky et al., 2009), ImageNet-LT (Deng et al., 2009), Places-LT (Zhou et al., 2017), and iNaturalist 2018 (Van Horn et al., 2018). We report Top-1 accuracy for Head (> 100), Medium (20–100), Tail (< 20), and Overall classes.

Baselines. We compare SADP with representative long-tailed recognition methods under both training-from-scratch and foundation-model fine-tuning settings. For training from scratch, we include CE, Focal Loss (Lin et al., 2017), MetaSAug (Li et al., 2021), CMO (Park et al., 2022), SAFA (Hong et al., 2022), GML (Du & Wu, 2023), ConCutMix (Pan et al., 2024), LOS (Sun et al., 2025), and CAR (Zhu et al., 2026). For foundation-model fine-tuning, we compare with BALLAD (Ma et al., 2021), LPT (Dong et al., 2022), Decoder (Wang et al., 2024), PECEL (Ru et al., 2024), LIFT (Shi et al., 2024), SRCD (Wan et al., 2026), and CAR (Zhu et al., 2026). LIFT serves as our main lightweight fine-tuning baseline. Detailed descriptions of the competing methods are provided in Appendix D.2.

Implementation details. We use CLIP ViT-B/16 as the pretrained backbone and follow the lightweight fine-tuning setting of LIFT (Shi et al., 2024). The pretrained backbone is frozen, while AdaptFormer modules and the cosine classifier are optimized. We train all models with SGD using a batch size of 128, an initial learning rate of 0.01, momentum of 0.9, and weight decay of 5×10^{-4}, with cosine learning-rate decay. The input resolution is 224×224, and logit-adjusted loss is used as the base classification objective. For SADP, the class-specific AGOP subspaces and feature-support scores are computed once before fine-tuning and kept fixed during training. We set the preservation coefficient to $\lambda = 0.6$ for all datasets unless otherwise specified.

5.2 Comparison Results

Tables 1 and 2 compare SADP with representative long-tailed recognition methods under both training-from-scratch and foundation-model fine-tuning settings. SADP achieves the best overall performance across all benchmarks and consistently performs favorably on tail classes, while maintaining competitive results on head and medium classes. These results demonstrate that support-aware directional preservation effectively reduces tail-class degradation without sacrificing the performance of frequent classes, and generalizes well across different dataset scales and imbalance patterns.

Table 2: Top-1 accuracy (%) comparison on iNaturalist and Places-LT under the fine-tuning setting. Results are reported for Head, Medium, Tail, and Overall. The best results are in **bold**, and the second-best are underlined.

Methods	Venue	iNaturalist				Places-LT			
		Head	Medium	Tail	Overall	Head	Medium	Tail	Overall
BALLAD (Ma et al., 2021)	–	–	–	–	–	49.3	50.2	48.4	49.5
LPT (Dong et al., 2022)	ICLR'2023	–	–	79.3	76.1	49.3	52.3	46.9	50.1
Decoder (Wang et al., 2024)	IJCV'2024	–	–	–	59.2	–	–	–	46.8
PECEL (Ru et al., 2024)	MM'2024	76.8	80.6	81.3	80.5	50.1	51.2	48.3	50.2
LIFT (Shi et al., 2024)	ICML'2024	72.4	79.0	81.1	79.1	51.3	52.2	50.5	51.5
SRCD (Wan et al., 2026)	ICML'2026	74.1	79.4	81.5	79.7	51.6	53.0	52.8	52.5
CAR (Zhu et al., 2026)	CVPR'2026	77.4	80.7	81.4	80.6	–	–	–	–
SADP (Ours)	–	**79.2**	**81.7**	**82.3**	**81.7**	**52.1**	**53.7**	**53.8**	**53.1**

Table 3: Ablation study on ImageNet-LT. We compare different fine-tuning and feature-preservation strategies. Results are reported in terms of Top-1 accuracy (%).

Method	Head	Medium	Tail	Overall
Linear Probe	77.6	73.7	68.0	74.4
Full Fine-tuning	82.2	73.9	53.8	74.4
LIFT	79.7	76.2	72.8	77.1
LIFT + Uniform Preservation	79.2	75.1	70.5	76.1
LIFT + Support-aware Full	80.7	75.8	73.1	77.3
LIFT + SADP (Ours)	**81.6**	**77.8**	**75.9**	**79.0**

5.3 Ablation Study

Effect of Support-Aware Directional Preservation. We conduct a controlled comparison on ImageNet-LT to examine the key designs of SADP. Starting from the same pretrained model, we compare Linear Probe, Full Fine-tuning, and LIFT with three preservation variants: *Uniform Preservation*, which constrains the full feature update with the same strength across classes; *Support-aware Full*, which introduces class-specific weights based on feature support; and SADP, which further restricts preservation to the pretrained discriminative directions. As shown in Table 3, support-aware preservation improves over uniform preservation, while SADP achieves the best overall performance by combining class-specific weighting with directional constraints. These results show that both *how much* and *which directions* to preserve are important for effective long-tailed fine-tuning.

Additional Analysis. We provide further analyses in Appendix C to better understand the behavior and effectiveness of SADP. Specifically, we examine the relationship between class frequency and tail-class behavior, discuss the use of classification margin for constructing class-wise AGOP, analyze class-wise improvements on tail classes, and report the additional computational cost of SADP. These results provide complementary evidence for our feature-support perspective and the design of SADP.

6 Conclusion

In this work, we study the heterogeneous behavior of tail classes during long-tailed fine-tuning of pretrained models. We show that classes with similar training frequencies can exhibit very different levels of forgetting, suggesting that class frequency alone is insufficient to explain their behavior. We introduce the perspective of Feature Support Erosion, where pretrained discriminative directions receive unequal support from the dominant training signals. Based on class-wise AGOP, we characterize such feature support and further propose SADP, which selectively constrains lightweight feature updates along poorly supported pretrained directions. Experiments on four long-tailed benchmarks demonstrate consistent improvements over strong training-from-scratch and fine-tuning baselines. These results highlight feature support as an important factor for understanding and improving long-tailed fine-tuning with pretrained models.

AI Assistance Disclosure

In this work, we used generative AI tools to assist with writing and language polishing, as well as literature retrieval and discovery of relevant prior work. All AI-assisted content and retrieved references were reviewed and verified by the authors. We take full responsibility for the final content of this work.

References

Shaden Alshammari, Yu-Xiong Wang, Deva Ramanan, and Shu Kong. Long-tailed recognition via weight balancing. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6887–6897. IEEE, 2022.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, 2019.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in neural information processing systems*, 35:16664–16678, 2022.

Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 715–724, 2021.

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*, pp. 9268–9277, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

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

Yingxiao Du and Jianxin Wu. No one left behind: Improving the worst categories in long-tailed learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15804–15813. IEEE, 2023.

Yan Hong, Jianfu Zhang, Zhongyi Sun, and Ke Yan. Safa: Sample-adaptive feature augmentation for long-tailed image classification. In *European conference on computer vision*, pp. 587–603. Springer, 2022.

Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6626–6636, 2021.

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

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Bo Li, Yongqiang Yao, Jingru Tan, Ruihao Gong, Jianwei Lu, and Ye Luo. Rectify representation bias in vision-language models for long-tailed recognition. *Neural Networks*, 172:106134, 2024.

Shuang Li, Kaixiong Gong, Chi Harold Liu, Yulin Wang, Feng Qiao, and Xinjing Cheng. Metasaug: Meta semantic augmentation for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5212–5221, 2021.

Tianhong Li, Peng Cao, Yuan Yuan, Lijie Fan, Yuzhe Yang, Rogerio S 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*, pp. 6918–6928, 2022.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Alexander Long, Wei Yin, Thalaiyasingam Ajanthan, Vu Nguyen, Pulak Purkait, Ravi Garg, Alan Blair, Chunhua Shen, and Anton Van den Hengel. Retrieval augmented classification for long-tail visual recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6949–6959. IEEE, 2022.

Teli Ma, Shijie Geng, Mengmeng Wang, Jing Shao, Jiasen Lu, Hongsheng Li, Peng Gao, and Yu Qiao. A simple long-tailed recognition baseline via vision-language model. *arXiv preprint arXiv:2111.14745*, 2021.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=37nvvqkCo5.

Haolin Pan, Yong Guo, Mianjie Yu, and Jian Chen. Enhanced long-tailed recognition with contrastive cutmix augmentation. *IEEE Transactions on Image Processing*, 33:4215–4230, 2024.

Seulki Park, Youngkyu Hong, Byeongho Heo, Sangdoo Yun, and Jin Young Choi. The majority can help the minority: Context-rich minority oversampling for long-tailed classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6887–6896, 2022.

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*, pp. 8748–8763. PmLR, 2021.

Adityanarayanan Radhakrishnan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin. Mechanism for feature learning in neural networks and backpropagation-free machine learning models. *Science*, 383(6690):1461–1467, 2024.

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.

Lixiang Ru, Xin Guo, Lei Yu, Yingying Zhang, Jiangwei Lao, Jian Wang, Jingdong Chen, Yansheng Li, and Ming Yang. Parameter-efficient complementary expert learning for long-tailed visual recognition. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 5393–5402, 2024.

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. In *Forty-first International Conference on Machine Learning*, 2024.

Saptarshi Sinha and Hiroki Ohashi. Difficulty-net: Learning to predict difficulty for long-tailed recognition. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6433–6442. IEEE, 2023.

Saptarshi Sinha, Hiroki Ohashi, and Katsuyuki Nakamura. Class-difficulty based methods for long-tailed visual recognition. *International Journal of Computer Vision*, 130(10):2517–2531, 2022.

Minseok Son, Inyong Koo, Jinyoung Park, and Changick Kim. Difficulty-aware balancing margin loss for long-tailed recognition. In *Proceedings of the aaai conference on artificial intelligence*, volume 39, pp. 20522–20530, 2025.

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 *International Conference on Learning Representations*, volume 2025, pp. 26681–26697, 2025.

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*, pp. 73–91. Springer, 2022.

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 *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 8769–8778. IEEE, 2018.

Wenhai Wan, Teng Zhang, Shao-Yuan Li, Xinrui Wang, Qiang-Sheng Hua, and Songcan Chen. Shortcut-resistant CAM distillation for long-tailed recognition. In *Forty-third International Conference on Machine Learning*, 2026.

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.

Yidong Wang, Zhuohao Yu, Jindong Wang, Qiang Heng, Hao Chen, Wei Ye, Rui Xie, Xing Xie, and Shikun Zhang. Exploring vision-language models for imbalanced learning. *International Journal of Computer Vision*, 132(1):224–237, 2024.

Lingjie Yi, Michael Yao, Weimin Lyu, Haibin Ling, Raphael Douady, and Chao Chen. Geometry of long-tailed representation learning: Rebalancing features for skewed distributions. In *International Conference on Learning Representations*, volume 2025, pp. 69619–69651, 2025.

Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(9):10795–10816, 2023.

Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16489–16498, June 2021.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9716–9725. IEEE, 2020.

Ziquan Zhu, Gaojie Jin, Hanruo Zhu, Si-Yuan Lu, Yunxiao Zhang, Zeyu Fu, Ronghui Mu, Guoqiang Zhang, Zhao Sun, Yuhang Xia, et al. Confusion-aware spectral regularizer for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28712–28722, 2026.

Appendix

CONTENTS

A Proof of the Support-Aware Feature Drift Bound **14**

A.1 Local Feature-Space Model 14

A.2 Directional Decomposition 14

A.3 One-Step Directional Preservation 15

A.4 Accumulated Feature Drift 15

A.5 Connection to Feature Support 16

B Related Work **16**

B.1 Long-Tailed Recognition 16

B.2 Long-Tailed Learning with Pretrained Models 17

B.3 Gradient-Based Characterization of Feature Learning 18

B.4 Relation to Our Work 18

C More Analysis and Discussion **19**

C.1 Class Frequency Does Not Fully Explain Tail-Class Performance 19

C.2 Why Do We Use the Classification Margin? 20

C.3 Class-wise Effect on Tail Classes 21

C.4 Computational Cost 23

C.5 Discussion on the Reliability of Tail-Class AGOP 24

D Experimental Details **25**

D.1 Implementation Details 25

D.2 Competitors 26

E Algorithm **29**

A Proof of the Support-Aware Feature Drift Bound

In this section, we provide the detailed analysis of Theorem 1. Our goal is to characterize how SADP affects feature updates along the pretrained discriminative directions.

A.1 Local Feature-Space Model

For class c, let

$$P_c = P_c^{\mathrm{pre}} = U_c^{\mathrm{pre}} (U_c^{\mathrm{pre}})^\top$$

be the orthogonal projector onto its pretrained discriminative subspace. Since the columns of U_c^{pre} are orthonormal, P_c satisfies

$$P_c^\top = P_c, \qquad P_c^2 = P_c.$$

We analyze one optimization step through a local quadratic model. Let g_t denote the gradient of the base long-tailed objective with respect to the local feature update at step t. The local objective with SADP is

$$\mathcal{Q}_t(\Delta h) = \langle g_t, \Delta h \rangle + \frac{1}{2\eta_t} \|\Delta h\|_2^2 + \frac{\lambda w_c}{2} \|P_c \Delta h\|_2^2,$$

where $\eta_t > 0$ controls the local update scale. The first term is the first-order approximation of the base objective, the second term is the standard quadratic proximity term, and the last term corresponds to directional preservation.

Without SADP, minimizing the first two terms gives the baseline local update

$$\Delta h_t^{\mathrm{base}} = -\eta_t g_t.$$

A.2 Directional Decomposition

We first show that SADP only suppresses the component of the update inside the pretrained discriminative subspace.

Lemma 1 (Directional Update Decomposition)**.** *The minimizer of $\mathcal{Q}_t$ is*

$$\Delta h_t^{\mathrm{SADP}} = -\eta_t (I - P_c) g_t - \frac{\eta_t}{1 + \eta_t \lambda w_c} P_c g_t.$$

Proof. Taking the derivative of $\mathcal{Q}_t$ with respect to Δh gives

$$\nabla_{\Delta h} \mathcal{Q}_t = g_t + \frac{1}{\eta_t} \Delta h + \lambda w_c P_c \Delta h,$$

where we have used $P_c^\top P_c = P_c$.

Setting the derivative to zero yields

$$(I + \eta_t \lambda w_c P_c)\, \Delta h_t^{\mathrm{SADP}} = -\eta_t g_t.$$

Because P_c is an orthogonal projector, any vector can be uniquely decomposed into components inside and outside its range:

$$g_t = P_c g_t + (I - P_c) g_t.$$

Moreover,

$$P_c (I - P_c) = 0.$$

Therefore, $I + \eta_t \lambda w_c P_c$ acts with eigenvalue $1 + \eta_t \lambda w_c$ on the range of P_c, and with eigenvalue 1 on its orthogonal complement. Its inverse is thus

$$(I + \eta_t \lambda w_c P_c)^{-1} = (I - P_c) + \frac{1}{1 + \eta_t \lambda w_c} P_c.$$

Substituting this expression into the update gives

$$\Delta h_t^{\mathrm{SADP}} = -\eta_t (I - P_c) g_t - \frac{\eta_t}{1 + \eta_t \lambda w_c} P_c g_t,$$

which proves the result. □

A.3 One-Step Directional Preservation

Lemma 1 directly gives the amount by which SADP suppresses the update along the pretrained discriminative subspace.

Lemma 2 (One-Step Directional Suppression)**.** *For each optimization step,*

$$\left\|P_c \Delta h_t^{\mathrm{SADP}}\right\|_2 = \rho_{c,t} \left\|P_c \Delta h_t^{\mathrm{base}}\right\|_2,$$

where

$$\rho_{c,t} = \frac{1}{1+\eta_t \lambda w_c} \in (0,1].$$

In contrast, the update in the orthogonal complement is unchanged:

$$(I-P_c)\Delta h_t^{\mathrm{SADP}} = (I-P_c)\Delta h_t^{\mathrm{base}}.$$

Proof. Applying P_c to the update in Lemma 1 gives

$$P_c \Delta h_t^{\mathrm{SADP}} = -\frac{\eta_t}{1+\eta_t \lambda w_c} P_c g_t.$$

For the baseline update,

$$P_c \Delta h_t^{\mathrm{base}} = -\eta_t P_c g_t.$$

Taking the Euclidean norm on both sides gives

$$\left\|P_c \Delta h_t^{\mathrm{SADP}}\right\|_2 = \frac{1}{1+\eta_t \lambda w_c} \left\|P_c \Delta h_t^{\mathrm{base}}\right\|_2.$$

Similarly, applying $I - P_c$ to Lemma 1 gives

$$(I-P_c)\Delta h_t^{\mathrm{SADP}} = -\eta_t (I-P_c) g_t = (I-P_c)\Delta h_t^{\mathrm{base}}.$$

This completes the proof. □

A.4 Accumulated Feature Drift

We now extend the one-step result to multiple optimization steps.

Proof of Theorem 1. The accumulated change of the feature representation after T steps can be written as

$$h_T - h_0 = \sum_{t=0}^{T-1} \Delta h_t^{\mathrm{SADP}}.$$

Projecting this change onto the pretrained discriminative subspace gives

$$P_c(h_T - h_0) = \sum_{t=0}^{T-1} P_c \Delta h_t^{\mathrm{SADP}}.$$

Using Lemma 1,

$$P_c(h_T - h_0) = -\sum_{t=0}^{T-1} \frac{\eta_t}{1+\eta_t \lambda w_c} P_c g_t.$$

By the triangle inequality,

$$\left\|P_c(h_T - h_0)\right\|_2 \le \sum_{t=0}^{T-1} \frac{\eta_t}{1+\eta_t \lambda w_c} \left\|P_c g_t\right\|_2.$$

This proves the first statement of Theorem 1.

If the local step size is fixed, i.e., $\eta_t = \eta$, and the projected gradient is bounded as

$$\|P_c g_t\|_2 \le G_c \qquad \text{for all } t,$$

then

$$\|P_c(h_T - h_0)\|_2 \leq \sum_{t=0}^{T-1} \frac{\eta}{1+\eta\lambda w_c} G_c = \frac{\eta T G_c}{1+\eta\lambda w_c}.$$

This proves the second statement. □

A.5 Connection to Feature Support

Finally, we show how the above bound connects to the feature support used by SADP.

Corollary 1 (Support-Aware Preservation)**.** *Suppose the preservation weight is given by*

$$w_c = f(S_c),$$

where f is a non-negative decreasing function of feature support, i.e., $f'(S_c) < 0$. Then the upper bound

$$\mathcal{B}_c(S_c) = \frac{\eta T G_c}{1+\eta\lambda f(S_c)}$$

is increasing with respect to S_c when G_c is held fixed. Consequently, lower feature support induces a tighter control on the pretrained discriminative feature drift.

Proof. Differentiating the bound with respect to S_c gives

$$\frac{\partial \mathcal{B}_c}{\partial S_c} = -\frac{\eta^2 T G_c \lambda f'(S_c)}{(1+\eta\lambda f(S_c))^2}.$$

Since $\eta > 0$, $\lambda > 0$, $G_c \geq 0$, and $f'(S_c) < 0$, we have

$$\frac{\partial \mathcal{B}_c}{\partial S_c} > 0.$$

Thus, a smaller S_c produces a smaller drift upper bound, completing the proof. □

The above analysis provides two complementary interpretations of SADP. First, the directional regularizer selectively suppresses updates within the pretrained discriminative subspace while leaving its orthogonal complement unchanged under the local model. Second, the support-dependent weight w_c makes this suppression stronger for poorly supported classes. Therefore, SADP provides targeted control over the pretrained feature directions that are more vulnerable during long-tailed fine-tuning.

B Related Work

B.1 Long-Tailed Recognition

Long-tailed recognition has been extensively studied from data re-balancing, loss re-weighting, classifier adjustment, representation learning, and ensemble learning. Early methods mainly compensate for the highly skewed class distribution through re-sampling or re-weighting. Class-Balanced Loss (Cui et al., 2019) re-weights samples according to the effective number of training examples, while LDAM (Cao et al., 2019) introduces class-dependent margins to enlarge the decision margins of minority classes. Balanced Softmax (Ren et al., 2020) modifies the softmax function according to class priors, and Logit Adjustment (Menon et al., 2021) corrects the bias induced by long-tailed label distributions directly in the logit space. LADE (Hong et al., 2021) further disentangles the influence of training and test label distributions to improve robustness under distribution shifts.

Another influential direction is decoupled learning. Kang et al. (2020) show that representations learned from the original long-tailed distribution can remain competitive, while much of the imbalance can be attributed to the classifier, motivating classifier re-training (cRT), learnable weight scaling (LWS), and τ-normalization. Following this paradigm, MiSLAS (Zhong et al., 2021) improves classifier re-training through label-aware smoothing and shifted batch normalization, while

Weight Balancing (Alshammari et al., 2022) studies the imbalance of classifier weight norms and improves both representation and classifier learning through regularization. More recently, LOS (Sun et al., 2025) revisits classifier re-training from the perspective of logit magnitude and proposes label over-smoothing, while CAR (Zhu et al., 2026) introduces a confusion-aware spectral regularizer to explicitly reduce inter-class confusion.

Beyond classifier calibration, many works focus on learning stronger tail representations. BBN (Zhou et al., 2020) uses two complementary branches with different sampling distributions to jointly learn representations and classifiers. RIDE (Wang et al., 2020) employs multiple distribution-aware experts and encourages expert diversity to improve performance across different frequency regions. PaCo (Cui et al., 2021) combines supervised contrastive learning with learnable class centers, while TSC (Li et al., 2022) improves the uniformity of long-tailed feature distributions through targeted supervised contrastive learning. Feature augmentation approaches also transfer richer variations to minority classes, including MetaSAug (Li et al., 2021), CMO (Park et al., 2022), and SAFA (Hong et al., 2022). More recent studies investigate long-tailed representation geometry: GML (Du & Wu, 2023) places greater emphasis on poorly performing categories, while FeatRecon (Yi et al., 2025) analyzes how head-dominated training distorts feature geometry and reconstructs the feature space to improve tail-class separability.

Despite their different formulations, most conventional methods are developed under a training-from-scratch setting, where the model must acquire discriminative representations directly from the imbalanced downstream data. Consequently, class frequency naturally plays a central role in sampling, weighting, margin design, and classifier correction. Our setting is different: a pretrained model already contains rich visual and semantic structures before long-tailed fine-tuning. Therefore, the key question is not only how to compensate for insufficient tail supervision, but also how downstream optimization interacts with and potentially modifies the knowledge inherited from pretraining.

B.2 Long-Tailed Learning with Pretrained Models

The emergence of large-scale pretrained models has introduced a different paradigm for long-tailed learning. Instead of learning visual representations solely from imbalanced downstream data, these approaches exploit the rich knowledge acquired during large-scale pretraining. BALLAD (Ma et al., 2021) is an early attempt to transfer vision-language pretraining to long-tailed recognition. It first fine-tunes the pretrained vision-language model on the target dataset and then freezes the backbone while learning an adapter with re-balanced samples. VL-LTR (Tian et al., 2022) further exploits both visual and linguistic information by learning class-wise visual-linguistic representations, demonstrating the benefit of text supervision particularly for data-scarce categories. RAC (Long et al., 2022) takes a different route and augments a pretrained image classifier with an external retrieval branch, allowing retrieved visual and textual information to complement parametric representations, especially for tail classes.

As pretrained models become increasingly large, parameter-efficient fine-tuning has received growing attention. LPT (Dong et al., 2022) freezes the pretrained model and introduces shared and group-specific learnable prompts for long-tailed classification. Wang et al. (Wang et al., 2024) systematically study vision-language models under imbalanced data and introduce a lightweight decoder to capture fine-grained information for tail classes, while also examining their interaction with conventional imbalance-aware objectives. ReCT (Li et al., 2024) studies representation bias in vision-language models and attributes part of the tail-class degradation to confusion with highly correlated head classes, motivating the separation and rectification of shared and class-specific representations.

Recent works further exploit external knowledge or parameter-efficient modules to improve long-tailed fine-tuning. PECEL (Ru et al., 2024) employs multiple parameter-efficient experts that specialize in complementary subsets of classes while sharing parameters to control training cost. LIFT (Shi et al., 2024) directly studies the effect of fine-tuning strength and shows that heavy fine-tuning can deteriorate tail-class performance; it therefore advocates adaptive lightweight fine-tuning to retain more pretrained knowledge. These studies establish that how pretrained models are fine-tuned is crucial for long-tailed recognition, rather than simply increasing model capacity or downstream supervision.

More recent work has begun to investigate finer-grained failure modes during foundation-model fine-tuning. SRCD (Wan et al., 2026) studies shortcut learning in tail classes and transfers more reliable visual evidence through class activation map distillation. These approaches indicate that pretrained knowledge can be distorted in different ways during long-tailed fine-tuning, including representation bias, concept confusion, and shortcut dependence.

Our work is related to this line of research but focuses on a different question. Existing methods mainly improve how pretrained knowledge is transferred, reduce the overall extent of fine-tuning, enrich supervision, or correct particular representation biases. In contrast, we investigate why pretrained class-specific features are preserved for some tail classes but substantially degraded for others even under the same long-tailed fine-tuning process. We show that different classes receive unequal optimization support for their pretrained discriminative directions. This motivates us to characterize *feature support* using class-wise AGOP and to selectively preserve poorly supported directions, rather than treating all pretrained features or all tail classes uniformly.

B.3 Gradient-Based Characterization of Feature Learning

Our analysis is also related to recent studies that use gradients to characterize the internal feature structure of neural networks. The Average Gradient Outer Product (AGOP) summarizes the input- or feature-space sensitivity of a model through the second-order statistics of its gradients (Radhakrishnan et al., 2024). Given a representation h and a scalar function f, AGOP can be written as

$$A_f = \frac{1}{N} \sum_{i=1}^{N} \nabla_h f(h_i) \nabla_h f(h_i)^\top.$$

Since A_f is positive semi-definite, its dominant eigenspace captures directions along which the model output exhibits high average sensitivity. This makes AGOP a useful tool for studying the feature directions emphasized by a learned predictor. Importantly, AGOP describes the strength of gradient variation along different directions rather than the sign of an update. Therefore, it is more naturally interpreted as a measure of feature dependence or directional coverage than as a measure of whether an optimization update is beneficial or harmful.

We adapt this idea to long-tailed fine-tuning in a class-specific manner. For each class, we construct the pretrained AGOP using the gradient of its classification margin, allowing us to identify a compact subspace associated with its pretrained discriminative behavior. We then compare this subspace with the gradient structure induced by the actual long-tailed training objective, particularly the dominant signals generated by head classes. This provides a measure of how strongly the pretrained discriminative directions of a class are covered by the optimization process. Unlike conventional gradient-conflict analyses, which mainly study the alignment or disagreement between individual update vectors, our goal is to characterize whether a pretrained feature subspace receives sufficient optimization support over training.

B.4 Relation to Our Work

The above lines of research address complementary aspects of long-tailed learning. Conventional methods mainly compensate for unequal class frequencies; difficulty-aware methods refine this view by considering class- or instance-dependent hardness; and pretrained-model-based methods aim to better retain and transfer knowledge acquired before downstream training. Our work connects the latter two perspectives from the viewpoint of optimization. We show that similar-frequency tail classes can rely on pretrained discriminative directions that receive very different levels of support from dominant training signals. When this support is insufficient, the corresponding pretrained feature structure is more likely to drift during fine-tuning and can eventually result in severe forgetting. Based on this observation, SADP uses feature support to determine *how strongly* a class should be protected and pretrained class-wise AGOP to determine *which directions* should be preserved.

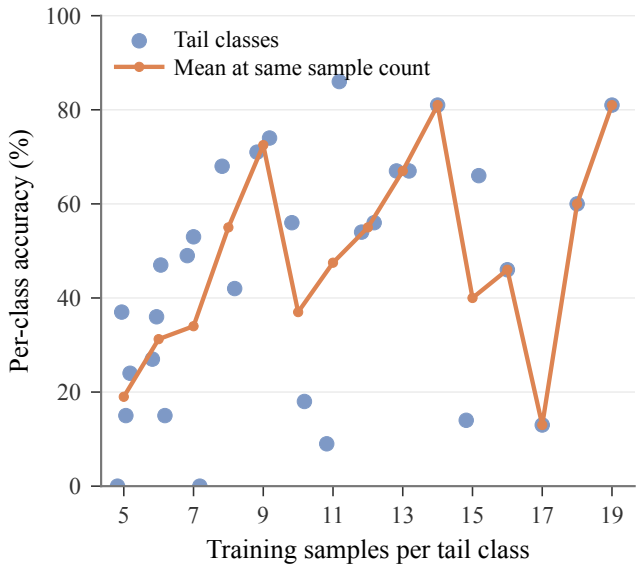


Figure 4: Class frequency alone does not determine tail-class performance. Each point denotes a tail class, and the orange curve shows the mean accuracy of classes with the same number of training samples. Large accuracy variations remain among classes with similar or identical frequencies.

C More Analysis and Discussion

C.1 Class Frequency Does Not Fully Explain Tail-Class Performance

Figure 4 provides a more detailed view of the relationship between training frequency and recognition performance within the tail region. Each blue point corresponds to one tail class, while the orange curve reports the mean accuracy of classes sharing the same number of training samples. Although class frequency is an important factor in long-tailed recognition, the figure shows that it does not fully determine tail-class performance.

A first observation is the large variation among classes with the same or very similar sample counts. At several frequencies, classes with nearly identical amounts of supervision exhibit substantially different accuracies, ranging from near-zero recognition to relatively high performance. This variation remains visible even when the sample count is fixed, indicating that the number of training examples alone cannot explain why some tail classes are well preserved while others fail severely after fine-tuning. In other words, frequency provides only a coarse description of the supervision available to a class, but does not characterize how that supervision interacts with the representation already learned by the pretrained model.

A second observation is that the mean accuracy does not increase monotonically with the number of tail-class samples. More training examples generally provide stronger supervision, but within this low-data regime, classes with slightly more samples are not necessarily easier to recognize than classes with fewer samples. This is consistent with the fact that long-tailed classes can differ in many other aspects, such as visual variability, inter-class similarity, and their relationship with the pretrained representation. Therefore, two classes with the same frequency need not have the same effective learning condition during downstream fine-tuning.

Importantly, our claim is not that class frequency is irrelevant. Frequency remains a major source of imbalance and is explicitly used by many successful long-tailed methods. Rather, our observation is that frequency is *insufficient* for explaining the strong class-level heterogeneity that remains after controlling for sample count. This distinction is important: methods based on re-sampling, re-weighting, or class-dependent margins can correct the imbalance in the amount of supervision, but they do not directly describe whether the pretrained discriminative features of a particular class are reinforced or neglected by the optimization signals generated from other classes.

One possible concern is that the variation in Figure 4 may simply reflect intrinsic class difficulty. While class difficulty can certainly contribute to the observed spread, it does not directly answer the question studied in this work. Difficulty describes how hard a class is to recognize, whereas we focus on whether its pretrained class-specific feature structure is preserved during fine-tuning. A tail class may be difficult yet still retain its pretrained representation, or it may initially be well recognized but later suffer substantial forgetting. Thus, difficulty and feature preservation capture different aspects of long-tailed behavior.

Another concern is that the orange curve can fluctuate because only a limited number of tail classes may share exactly the same sample count. We therefore use this curve only as an illustrative summary rather than as a statistical estimator of a monotonic frequency–accuracy relationship. The more important evidence lies in the class-level scatter itself: substantial accuracy variation remains even when frequency is matched. In our main analysis, we further reduce this potential confound by comparing Stable and Forgotten tail classes with similar training frequencies.

Finally, Figure 4 should be viewed as evidence for the limitation of frequency-based explanations rather than as direct evidence for our proposed mechanism. The mechanism is examined separately through the feature-support analysis in the main paper. Specifically, we show that Stable and Forgotten tail classes exhibit a clear gap in Head Feature Support, followed by different levels of Feature Retention and pretrained-correct margin during fine-tuning. Together, these results suggest that beyond the amount of class-specific supervision, whether the pretrained discriminative directions of a class are sufficiently covered by the dominant training signals is an important factor in determining its stability during long-tailed fine-tuning.

C.2 Why Do We Use the Classification Margin?

A key design choice in our analysis is to construct the pretrained class-wise AGOP using the gradient of the classification margin rather than the cross-entropy loss. This choice follows from what we want the pretrained AGOP to represent. Our goal is not to measure how difficult a pretrained sample is to optimize, but to identify the feature directions that are important for distinguishing a particular class from its competing classes.

For a sample of class c, we define the classification margin as

$$m_c(x) = z_c(x) - \max_{j \neq c} z_j(x),$$

where $z_c(x)$ is the logit of class c. We then compute the class-specific gradient with respect to the intermediate representation $h(x)$,

$$g_c(x) = \nabla_h m_c(x),$$

and construct the corresponding class-wise AGOP from these gradients. Intuitively, $g_c(x)$ describes which feature directions most strongly affect the separation between class c and its strongest competing class. The dominant eigenspace of the resulting AGOP therefore provides a natural approximation to the pretrained discriminative subspace of class c.

Using the class logit z_c alone would not provide the same information. A feature direction may strongly increase z_c while simultaneously increasing the logits of competing classes, in which case it does not necessarily improve the actual decision for class c. The margin explicitly incorporates this relative competition. In particular, the term $\max_{j \neq c} z_j$ focuses on the most competitive alternative class, making the resulting gradient closely related to the local decision boundary that determines whether a sample is recognized as class c. This is particularly useful in long-tailed recognition, where forgetting often appears as a pretrained class being overtaken by another class after fine-tuning.

Cross-entropy provides a different type of signal. For a sample with label c, its gradient depends not only on the class-separating directions but also on the current softmax confidence. Once a pretrained model already classifies a sample with high confidence, the cross-entropy gradient can become very small. Conversely, samples that are poorly classified can produce much larger gradients. As a result, a CE-based AGOP tends to emphasize samples that currently incur a large loss, mixing two

factors: the feature directions used for class discrimination and the current difficulty or confidence of individual samples. This is appropriate when the goal is to characterize the optimization signal, but less desirable when the goal is to recover the class-specific structure already encoded by the pretrained model.

The margin largely separates these two roles. Even when the model is already confident, the margin gradient can still reveal which representation directions control the relative separation between the target class and its strongest competitor. Therefore, it is better suited to our purpose of asking *what pretrained feature directions does class c rely on?*, rather than *which samples currently produce the largest training loss?* Importantly, we do not interpret the magnitude of the margin itself as the amount of pretrained knowledge. The margin is only used to define a class-discriminative scalar function whose gradient exposes the relevant feature directions.

This also explains why we intentionally use different gradients on the two sides of our feature-support analysis. For the pretrained class-wise AGOP, we use margin gradients because we aim to identify the discriminative directions associated with class c. In contrast, for the head-class AGOP used to measure feature support, we use gradients of the actual fine-tuning objective,

$$q_i = \nabla_h \mathcal{L}_{\text{train}}(x_i, y_i),$$

because this side is intended to represent the optimization signal that the model actually receives during long-tailed fine-tuning. Thus, the two gradients answer two different questions: the margin gradient characterizes *what the pretrained class depends on*, while the training-loss gradient characterizes *what the downstream optimization reinforces*. Their overlap consequently measures whether the pretrained discriminative subspace of a class is sufficiently covered by the actual training dynamics.

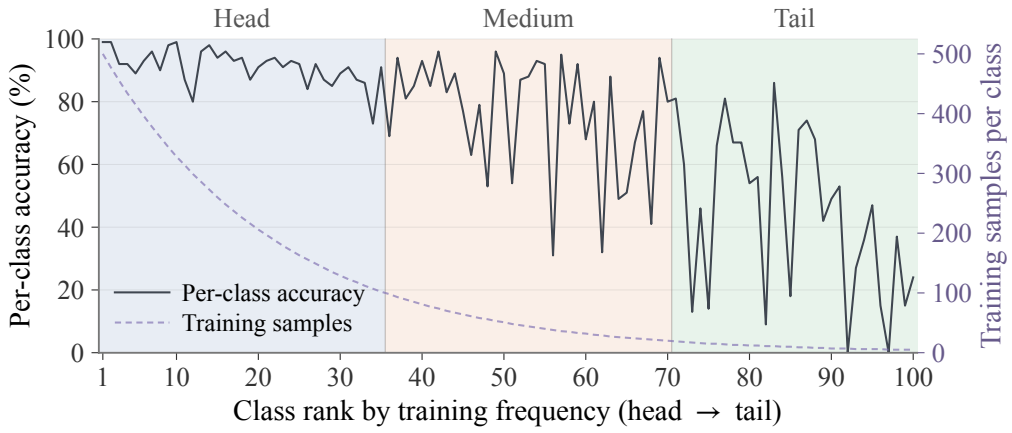
Finally, we note that AGOP is constructed from gradient outer products and is therefore insensitive to gradient sign. Our support metric should consequently be interpreted as measuring directional *coverage* rather than whether an individual update is beneficial or harmful. This distinction is important: low support indicates that a pretrained discriminative subspace receives little optimization energy, while the subsequent feature-retention and forgetting analyses are used to examine whether such insufficient support is associated with the erosion of pretrained class-specific structure.

C.3 Class-wise Effect on Tail Classes

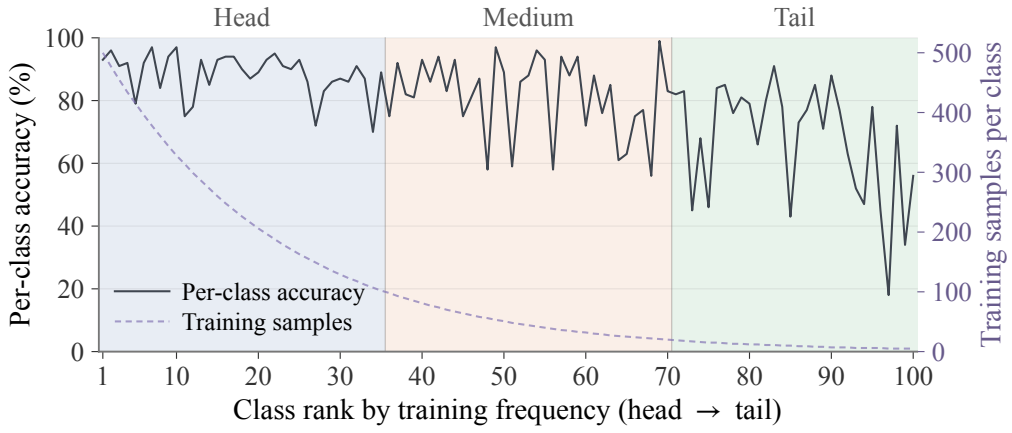
To better understand where the improvement of SADP comes from, we first compare the per-class accuracy of the baseline and SADP across the full class-frequency spectrum in Figure 5. Classes are ranked by training frequency from head to tail, and both methods are trained on exactly the same long-tailed distribution. The baseline exhibits increasingly large performance variations as the class frequency decreases. While most head classes maintain high accuracy, the tail region contains several severe failures, including classes whose accuracy drops to nearly zero. At the same time, other tail classes with comparable numbers of training samples remain well recognized. This again shows that tail-class behavior cannot be explained by sample frequency alone.

SADP substantially alleviates this class-wise instability. Compared with the baseline, the performance of head classes remains largely unchanged, while many poorly performing tail classes obtain clear improvements. In particular, several extreme failures in the tail region are noticeably reduced, resulting in a more stable class-wise accuracy distribution. Importantly, these improvements are obtained without changing the training distribution or introducing additional tail-class samples. The difference therefore comes from how the pretrained representation is modified during fine-tuning, which is consistent with our feature-support perspective: SADP provides stronger protection for poorly supported pretrained directions while avoiding unnecessary constraints on well-supported features.

To further examine whether these improvements are broadly distributed across tail classes rather than being dominated by a few large gains, we report the empirical cumulative distribution function (ECDF) of tail-class accuracy changes in Figure 6. For an ECDF, a curve shifted further to the right indicates larger improvements over the corresponding reference. The blue curve measures the accuracy change of each tail class after fine-tuning with SADP relative to its pretrained performance. At the zero-change boundary, the cumulative fraction is approximately 0.2, indicating that around 80%



(a) Baseline



(b) SADP

Figure 5: Class-wise comparison between the baseline and SADP. Classes are ranked by training frequency from head to tail. Compared with the baseline, SADP reduces severe performance degradation among tail classes while largely preserving head-class performance.

of tail classes achieve a positive fine-tuning gain. Therefore, the benefit of SADP is not restricted to a small subset of tail classes; most tail classes either preserve or improve their pretrained recognition performance after fine-tuning.

More importantly, the orange curve reports the class-wise accuracy gain of SADP over the lightweight fine-tuning baseline. The entire curve lies to the right of the zero-change line, showing that SADP outperforms the baseline for every tail class in this experiment. This distribution-level result provides stronger evidence than the average Tail accuracy alone. An improvement in mean accuracy could, in principle, be caused by a small number of classes receiving very large gains while other classes deteriorate. In contrast, the ECDF shows that the improvement of SADP is consistently distributed across tail classes rather than arising from such a trade-off.

The two analyses therefore provide complementary evidence. The class-wise plots show that the baseline suffers from highly heterogeneous tail behavior and that SADP particularly reduces severe class-level degradation. The ECDF further confirms that this effect is systematic: most tail classes benefit from fine-tuning with SADP, and all tail classes outperform the corresponding baseline in the reported experiment. These observations are consistent with the design of SADP, which does not uniformly suppress feature learning but selectively constrains updates along poorly supported pretrained discriminative directions. As a result, SADP can reduce destructive feature changes for vulnerable tail classes while retaining sufficient flexibility for downstream learning.

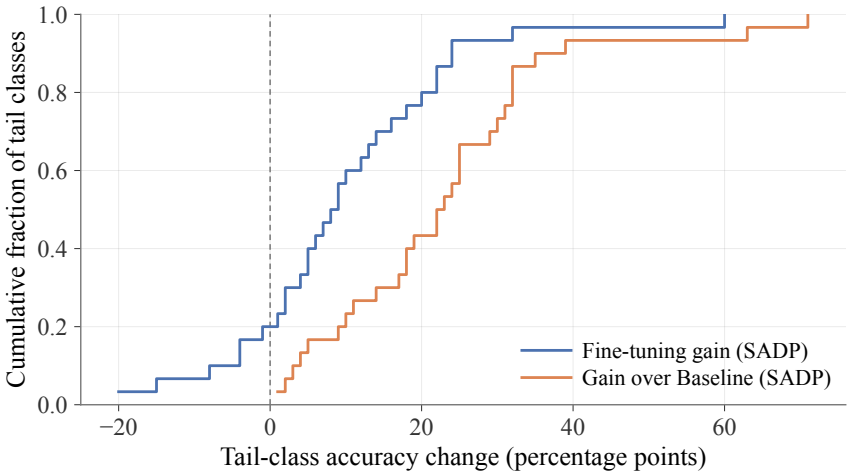


Figure 6: Distribution of tail-class accuracy gains. The empirical cumulative distribution functions (ECDFs) show the class-wise accuracy changes for tail classes. The blue curve measures the fine-tuning gain of SADP relative to the pretrained model, while the orange curve measures the gain of SADP over the baseline. A right-shifted curve indicates larger improvements. Approximately 80% of tail classes obtain positive fine-tuning gains, while the orange curve lies entirely to the right of zero, indicating consistent improvements over the baseline across all tail classes.

C.4 Computational Cost

We further analyze the computational and memory overhead introduced by SADP. All measurements are conducted on CIFAR100-LT using a single NVIDIA A100 GPU under the same fine-tuning configuration. As shown in Table 4, SADP introduces only moderate additional computation compared with the lightweight fine-tuning baseline. The training time increases from 9.88 to 11.97 minutes, corresponding to a relative cost of $1.21\times$. GPU-hours and total computation show a similar increase, indicating that the additional preservation mechanism incurs limited training overhead.

Table 4: Computational cost. All results are measured using a single NVIDIA A100 GPU under the same fine-tuning configuration. Relative cost is normalized by the baseline.

Method	Time (min)	GPU-hours	Compute (PFLOP)	Relative Cost
Baseline	9.88	0.165	23.72	$1.00\times$
SADP	11.97	0.199	28.72	$1.21\times$

The additional computation mainly comes from projecting the lightweight feature update $\Delta h(x)$ onto the pretrained class-specific subspace and evaluating the SADP regularization term. Importantly, the class-wise AGOP statistics are computed only once before fine-tuning. After eigendecomposition, we discard the full AGOP matrix and retain only the top-k eigenvectors U_c^{pre} together with the scalar support weight w_c. Therefore, no repeated AGOP computation is required during downstream optimization.

The storage overhead is also small. For K classes and a feature dimension d, each class only stores $U_c^{\mathrm{pre}} \in \mathbb{R}^{d\times k}$, resulting in a total storage complexity of

$$\mathcal{O}(Kdk),$$

rather than $\mathcal{O}(Kd^2)$ for storing the complete class-wise AGOP matrices. For example, with $d = 768$ and $k = 16$, storing the subspaces for CIFAR100-LT requires only about 2.5 MB in FP16, while the class-wise support weights introduce negligible additional storage.

SADP therefore does not require a large increase in GPU memory. A direct feature-preservation approach would require an additional frozen pretrained model to obtain the original representation

during every training iteration, introducing another backbone forward pass and the corresponding intermediate activations. In contrast, SADP operates directly on the lightweight feature update $\Delta h(x)$ produced within the existing fine-tuning model. The pretrained backbone remains frozen and shared with the original lightweight fine-tuning pipeline, while only the compact class-specific subspaces are additionally stored. Consequently, SADP achieves class- and direction-aware preservation with modest computational cost and without the memory overhead of maintaining a separate teacher network.

C.5 Discussion on the Reliability of Tail-Class AGOP

A potential concern is that class-wise AGOP may be unreliable for tail classes with very few training samples. This is a reasonable concern, since the empirical AGOP is estimated from a finite number of samples, and its full matrix can have high variance when the class size is small. However, SADP does not require an accurate estimation of the entire tail-class AGOP matrix. Instead, it only uses its leading eigenspace to identify the dominant pretrained discriminative directions. We discuss several reasons why this representation is suitable for our setting.

Pretrained rather than fine-tuned gradients. The class-wise AGOP used by SADP is computed from the frozen pretrained model before long-tailed fine-tuning. Therefore, it characterizes the discriminative structure already encoded by pretraining, rather than estimating a representation from scratch using the few tail-class samples. This distinction is important: tail samples are used as probes to reveal which directions of an already learned representation are most relevant to the class. The resulting gradients are thus determined by both the input samples and the pretrained model, whose representation has been learned from large-scale pretraining. Moreover, because the backbone is fixed during this stage, the estimated directions are not affected by the non-stationary representations introduced by downstream optimization.

SADP only relies on the dominant subspace. Although estimating a full $d \times d$ AGOP matrix from a small number of samples can be statistically noisy, SADP does not use the full matrix or require all of its eigenvectors to be reliable. We retain only the top-k eigenvectors, U_c^{pre}, which summarize the dominant directions of variation in the class-wise margin gradients. The preservation loss depends on the subspace spanned by these vectors rather than on the individual AGOP entries or the exact eigenvalues. Consequently, perturbations that change individual matrix entries but leave the leading subspace approximately unchanged have limited influence on the SADP objective.

This distinction can also be understood from the form of our regularizer:

$$\mathcal{L}_{\mathrm{SADP}}(x, c) = \frac{w_c}{k} \left\| (U_c^{\mathrm{pre}})^\top \Delta h(x) \right\|_2^2 .$$

The loss depends on the projection onto the span of U_c^{pre}. It is therefore invariant to an orthogonal change of basis within the same subspace. In other words, SADP does not require each eigenvector to have an exact and individually stable semantic interpretation; what matters is whether the estimated low-dimensional subspace captures the major pretrained discriminative directions.

Margin gradients provide a focused class-specific signal. We construct the pretrained class-wise AGOP using gradients of the classification margin rather than gradients of the cross-entropy loss. The margin directly measures the separation between the target class and its strongest competing class. Its gradient therefore focuses the AGOP on directions that locally affect the pretrained decision boundary. This avoids an additional confidence-dependent scaling introduced by cross-entropy gradients, which can make gradient magnitudes extremely small for confidently classified pretrained samples. As a result, our construction is intended to identify discriminative directions rather than estimate class difficulty from a small tail-class set.

Low sample size does not imply an arbitrary subspace. Importantly, the number of samples and the reliability of the estimated discriminative subspace should not be treated as equivalent quantities. A small class can still exhibit a coherent gradient structure if its samples activate similar pretrained discriminative directions. In such a case, a small number of samples may be sufficient to recover the dominant subspace. Conversely, even a larger class may exhibit a more diverse gradient structure. This is closely related to our main motivation: class frequency alone does not fully determine the

class-specific behavior of pretrained representations. Thus, we do not assume that every tail-class AGOP is equally accurate solely because the classes have similar sample counts.

Robustness to estimation noise. We further note that estimation noise does not automatically translate into an equally large error in SADP. The method first compresses the AGOP into a low-dimensional subspace and then uses this subspace through a projection penalty. The support score is also an aggregate overlap between this subspace and the head-class AGOP, rather than an element-wise comparison between two matrices. These operations reduce the dependence of SADP on individual noisy AGOP entries. More fundamentally, SADP is designed as a preservation mechanism: even when the estimated subspace contains moderate noise, the resulting regularization suppresses part of the lightweight feature update rather than introducing a new discriminative direction into the model. Therefore, estimation errors primarily affect which updates are preserved, rather than directly forcing the representation toward an incorrect target feature.

Limitation and practical implication. We nevertheless acknowledge that extremely small classes can lead to higher uncertainty in the estimated AGOP, particularly when their gradient directions are highly heterogeneous or when the leading eigenspace has no clear spectral separation. Our claim is therefore not that few-shot AGOP estimation is noise-free, but that SADP only requires a useful estimate of the dominant pretrained discriminative subspace, which is substantially weaker than accurately recovering the full AGOP. Empirically, the consistent tail-class improvements of SADP suggest that this approximation is sufficiently informative for long-tailed fine-tuning. A more explicit treatment of uncertainty in class-wise subspace estimation, such as shrinkage or confidence-aware preservation, is an interesting direction for future work.

D Experimental Details

D.1 Implementation Details

Training configuration. We use CLIP ViT-B/16 (Radford et al., 2021) as the pretrained backbone for all fine-tuning experiments. The input resolution is 224×224. Following the lightweight fine-tuning protocol, the pretrained backbone is kept frozen, while AdaptFormer modules and the cosine classifier are optimized. The classifier is initialized using CLIP text features, and the scale of the cosine classifier is set to 25. We use Logit Adjustment (Menon et al., 2021) as the base long-tailed classification loss.

All models are optimized using SGD with a batch size of 128, an initial learning rate of 0.01, momentum of 0.9, and weight decay of 5×10^{-4}. The learning rate is scheduled using cosine annealing. We train for 10 epochs on CIFAR100-LT, ImageNet-LT, and Places-LT, and 20 epochs on iNaturalist 2018. Automatic mixed precision is used during training. For data augmentation, we adopt random resized cropping and random horizontal flipping during training. At inference time, images are resized and center-cropped to 224×224.

Lightweight fine-tuning. We insert AdaptFormer modules into the Transformer blocks while keeping the original pretrained parameters fixed. Given K classes and L Transformer blocks, the bottleneck dimension follows the lightweight configuration

$$r = 2^{\lfloor \log_2(K/(2L)) \rfloor}.$$

Only the parameters of the lightweight modules and the classifier are updated during downstream training. This setting allows us to explicitly isolate the feature modifications introduced by fine-tuning and write the updated representation as

$$h'(x) = h(x) + \Delta h(x),$$

where $h(x)$ is produced by the frozen pretrained backbone and $\Delta h(x)$ is the feature update introduced by the trainable module.

AGOP computation. All class-specific AGOP statistics are computed once from the pretrained model before downstream fine-tuning. For each class c, we use the classification margin

$$m_c(x) = z_c(x) - \max_{j \neq c} z_j(x)$$

and compute its gradient with respect to the selected intermediate representation $h(x)$. The resulting class-wise AGOP is eigendecomposed, and its top-k eigenvectors form the pretrained discriminative subspace U_c^{pre}. In contrast, the head-class AGOP is constructed using gradients of the actual downstream training objective, so that it reflects the optimization signal received during long-tailed fine-tuning. The pretrained subspaces and corresponding feature-support scores are cached before training and remain fixed throughout fine-tuning, introducing no repeated AGOP computation during optimization.

SADP configuration. We map the feature-support score of each class to its preservation strength using a temperature $\tau = 0.9$, such that classes with lower support receive stronger regularization. The resulting class-wise weights are normalized to maintain a stable overall regularization scale. To construct the pretrained discriminative subspace, we retain the top $k = 16$ directions of the class-wise AGOP. We choose a relatively small k because SADP aims to preserve only the dominant discriminative structure rather than constrain the entire feature space. A very small k may miss useful discriminative directions, whereas a large k can include low-energy directions that are less reliably estimated for tail classes with limited samples. We therefore use $k = 16$ as a balance between preserving informative directions and avoiding excessive constraints from potentially noisy directions. The SADP regularizer is defined as

$$\mathcal{L}_{\mathrm{SADP}} = \frac{w_c}{k} \left\| (U_c^{\mathrm{pre}})^\top \Delta h(x) \right\|_2^2,$$

and the final objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{LT}} + \lambda \mathcal{L}_{\mathrm{SADP}}.$$

Unless otherwise specified, we use $\tau = 0.9$, $k = 16$, and $\lambda = 0.6$ for all datasets, while keeping the remaining optimization settings identical to the corresponding lightweight fine-tuning baseline.

D.2 Competitors

In this section, we provide additional details of the representative long-tailed recognition methods used for comparison in the main paper. We group them into training-from-scratch methods and pretrained-model-based methods according to their training paradigms.

D.2.1 Training-from-Scratch Long-Tailed Methods

Focal Loss. Focal Loss (Lin et al., 2017) modifies the standard cross-entropy objective by down-weighting well-classified examples and assigning relatively larger weights to hard examples. The modulation term depends on the predicted confidence, allowing training to focus more on samples that are difficult to classify. Although originally proposed for dense object detection, Focal Loss has been widely adopted as a simple hard-example-aware baseline for imbalanced recognition.

MetaSAug. MetaSAug (Li et al., 2021) addresses the lack of feature diversity in minority classes through semantic feature augmentation. It builds on implicit semantic data augmentation, but avoids directly estimating class-conditional statistics from the limited samples of tail classes. Instead, it uses a meta-learning procedure with a small balanced validation set to learn how semantic directions extracted from data-rich classes should be transformed and transferred to minority classes. This provides a stronger feature-space augmentation strategy for tail classes.

CMO. Context-Rich Minority Oversampling (CMO) (Park et al., 2022) improves minority-class diversity by exploiting the rich visual context contained in majority-class images. Specifically, it pastes image regions from minority classes onto majority-class images and uses the latter as diverse background contexts, producing additional training samples for tail classes. Compared with conventional random oversampling, CMO introduces richer appearance variations without changing the network architecture and can be combined with different long-tailed objectives.

SAFA. Sample-Adaptive Feature Augmentation (SAFA) (Hong et al., 2022) performs augmentation directly in the feature space. It extracts diverse semantic directions from head classes and transfers them to tail samples, while making the augmentation sample-dependent rather than applying the same perturbation to all instances. In addition, contrastive and diversity-related objectives are introduced to encourage the transferred directions to be class-irrelevant and to expand the feature space of tail classes. As a result, SAFA improves tail representations without introducing additional inference-time computation.

GML. GML (Du & Wu, 2023) revisits long-tailed recognition from the perspective of worst-class performance. It observes that optimizing only average accuracy can still leave a small number of classes with extremely poor or even zero recall, and that few-shot classes are not necessarily always the worst-performing ones. GML therefore introduces an objective based on the geometric mean of class-wise recall and applies it during classifier re-training to encourage a more uniform distribution of performance across classes. This makes it particularly relevant to our setting, where classes with similar frequencies may still exhibit very different recognition behaviors.

ConCutMix. ConCutMix (Pan et al., 2024) extends CutMix to long-tailed recognition by addressing the semantic inconsistency of conventional area-based mixed labels. Rather than determining the target label solely from the spatial mixing ratio, ConCutMix measures semantic similarity between samples in a contrastively learned representation space and uses this information to refine the mixed labels. In this way, the augmented samples provide more semantically reliable supervision, which is especially useful when synthesizing additional variations for tail classes.

LOS. Label Over-Smoothing (LOS) (Sun et al., 2025) revisits the second-stage classifier re-training paradigm and analyzes long-tailed recognition through the magnitude and distribution of classifier logits. It shows that, once the logit magnitude is relatively balanced across classes, further reducing the overall magnitude can alleviate optimization disturbance and improve recognition. Based on this observation, LOS introduces a simple over-smoothed label distribution that regularizes classifier outputs without explicitly requiring the class-frequency prior, providing a strong classifier-level baseline.

CAR. Confusion-Aware Spectral Regularizer (CAR) (Zhu et al., 2026) approaches long-tailed recognition from the perspective of inter-class confusion rather than class frequency alone. It constructs a differentiable class confusion matrix during training and regularizes its spectral properties, with the goal of preventing a small number of highly confused class directions from dominating the prediction errors. This confusion-centric formulation improves the generalization of poorly recognized categories and can be incorporated into both training-from-scratch and pretrained fine-tuning pipelines, making it a recent and broadly applicable baseline.

D.2.2 Pretrained-Model-Based Long-Tailed Methods

BALLAD. BALLAD (Ma et al., 2021) is an early attempt to leverage large-scale vision-language pretraining for long-tailed recognition. It first continues training the vision-language backbone on the target long-tailed dataset and then freezes the backbone while learning an additional adapter using re-balanced samples. This two-stage design exploits the semantic knowledge acquired during pretraining while reducing the bias introduced by the imbalanced downstream distribution.

LPT. Long-tailed Prompt Tuning (LPT) (Dong et al., 2022) adopts parameter-efficient prompt tuning instead of updating the entire pretrained model. It keeps the pretrained backbone frozen and introduces a shared prompt to capture general target-domain information together with group-specific prompts for samples sharing similar characteristics. A two-stage optimization strategy is used to first adapt the shared prompt and then learn the group-specific prompts, providing an efficient alternative to full fine-tuning.

Decoder. Wang et al. (Wang et al., 2024) systematically investigate vision-language models under imbalanced data and show that directly applying pretrained models is still insufficient for long-tailed recognition. They introduce a lightweight decoder to capture finer-grained features for minority classes and study its combination with prompt tuning, fine-tuning, and conventional imbalance-aware objectives such as Focal Loss and Balanced Softmax. This method provides a simple way to improve tail-class recognition while retaining the advantages of large-scale vision-language pretraining.

PECEL. Parameter-Efficient Complementary Expert Learning (PECEL) (Ru et al., 2024) addresses the degradation of pretrained representations during long-tailed fine-tuning by introducing multiple parameter-efficient experts. Each expert is encouraged to specialize in complementary subsets of classes through a sample-aware logit adjustment objective, and their predictions are aggregated to obtain balanced recognition performance. To avoid the large parameter cost typically associated with multi-expert models, PECEL further introduces parameter sharing among experts, maintaining parameter efficiency while improving class-wise complementarity.

LIFT. LIFT (Shi et al., 2024) directly studies how the extent of fine-tuning affects long-tailed recognition with foundation models. It observes that heavy fine-tuning can substantially deteriorate tail-class performance and attributes this behavior to inconsistent class conditions during optimization. Based on this observation, LIFT adopts adaptive lightweight fine-tuning, where most pretrained parameters remain frozen and only a small number of task-specific parameters are updated, thereby reducing both training cost and the risk of damaging pretrained representations.

SRCD. Shortcut-Resistant CAM Distillation (SRCD) (Wan et al., 2026) revisits long-tailed recognition from the perspective of shortcut learning. It observes that tail classes are more likely to rely on spurious cues such as background and texture because their limited samples often share similar contexts. SRCD therefore transfers object-focused visual evidence from head classes to tail classes in the Class Activation Map (CAM) space, constructing a shortcut-resistant teacher and distilling its spatial evidence to tail-class predictions.

CAR. CAR (Zhu et al., 2026) can also be applied in the fine-tuning-from-pretrained setting. Rather than modifying the pretrained architecture itself, CAR regularizes the frequency-weighted class confusion matrix during fine-tuning and minimizes its spectral norm to reduce dominant inter-class confusion. This provides a complementary perspective to parameter-efficient fine-tuning methods, as it focuses on the structure of prediction errors rather than directly controlling the amount of parameter updating.

E ALGORITHM

Algorithm 1 Support-Aware Directional Preservation (SADP)

Require: Long-tailed dataset $\mathcal{D}$, frozen pretrained backbone f_{θ_0}, lightweight module g_ϕ, top-k dimension k, coefficient λ
Ensure: Fine-tuned parameters ϕ

Stage 1: Precompute support and pretrained subspaces
for each tail class c **do**
    Compute the pretrained class-wise AGOP A_c^{pre} from margin gradients
    Extract top-k eigenvectors U_c^{pre}
end for
Compute the head-class AGOP B_{head} using gradients of the downstream training loss
for each tail class c **do**
    Compute feature support

$$S_c = \frac{\mathrm{Tr}(P_c^{\mathrm{pre}} B_{\mathrm{head}})}{\mathrm{Tr}(B_{\mathrm{head}})}$$

    Convert S_c to preservation weight w_c
end for
Cache $\{U_c^{\mathrm{pre}}, w_c\}$

Stage 2: Support-aware fine-tuning
for each minibatch $\mathcal{B}$ **do**
    Compute the base long-tailed loss $\mathcal{L}_{\mathrm{LT}}$
    Obtain lightweight feature update $\Delta h(x)$
    Compute

$$\mathcal{L}_{\mathrm{SADP}} = \frac{1}{|\mathcal{B}_T|} \sum_{(x_i, y_i) \in \mathcal{B}_T} \frac{w_{y_i}}{k} \left\| (U_{y_i}^{\mathrm{pre}})^\top \Delta h(x_i) \right\|_2^2$$

    Optimize

$$\mathcal{L} = \mathcal{L}_{\mathrm{LT}} + \lambda \mathcal{L}_{\mathrm{SADP}}$$

    Update the lightweight module and classifier
end for
return ϕ