CAFE: CLOSED-LOOP CLASSIFIER-AWARE FEATURE ENRICHMENT FOR LONG-TAILED RECOGNITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-tailed recognition remains challenging due to classifier bias induced by highly imbalanced training data. Existing information augmentation methods often overlook which tail samples actually require augmentation and whether generated features remain beneficial as the classifier evolves. To address these issues, we propose Classifier-Aware Feature Enrichment (CAFE), a two-stage classifier fine-tuning framework that establishes a closed-loop augmentation process. Guided by the evolving classifier, CAFE identifies challenging tail samples and selectively augments them using informative features transferred from data-rich classes. The generated features are continuously evaluated, and beneficial synthetic features are retained while redundant or detrimental ones are removed. The updated classifier subsequently guides the next round of sample identification and feature enrichment, allowing the augmentation process to adapt continuously to the evolving classifier. Extensive experiments on CIFAR10-LT, CIFAR100-LT, ImageNet-LT, and iNaturalist 2018 demonstrate the effectiveness of CAFE and show that it achieves competitive performance against state-of-the-art methods. Code will be made publicly available.

1 INTRODUCTION

Deep neural networks (DNNs) Sze et al. (2017); Samek et al. (2021) have achieved remarkable success across a wide range of applications, including medical image segmentation Liu et al. (2024a), object detection Lin et al. (2017), and visual recognition Chen et al. (2024). However, much of this success relies on relatively balanced distributions with sufficient training samples for different classes. Such an idealized assumption is often difficult to satisfy in real-world applications. Due to variations in data availability, collection difficulty, and class occurrence frequency, the number of training samples can vary substantially across different classes.

In practice, real-world data often follows a long-tailed distribution, where head classes contain abundant samples while tail classes are severely underrepresented. When DNNs are trained directly on such imbalanced data, the optimization process is inevitably dominated by the head classes. Consequently, the learned classifier becomes biased toward head classes Kang et al. (2020); Ren et al. (2020), resulting in less reliable decision boundaries for underrepresented tail classes. Such imbalance causes several challenges, including limited intra-class diversity, biased classifier weights, increased inter-class confusion, and degraded recognition performance on tail classes.

Existing long-tailed recognition methods can be broadly grouped into three categories Zhang et al. (2023). Class re-balancing methods aim to alleviate class imbalance by adjusting the training distribution or optimization objective. Typical strategies include re-sampling Wang et al. (2019), class-sensitive losses Cao et al. (2019), and logit adjustment Menon et al. (2021). A representative example is Focal Loss Lin et al. (2020), which focuses training on a sparse set of hard examples and prevents the vast number of easy negatives from overwhelming the detector during training.

The second line of research focuses on information augmentation, which mainly includes data augmentation and transfer learning. Data augmentation Shorten & Khoshgoftaar (2019); Kim et al. (2020) enriches the training data by generating or synthesizing additional samples, while transfer learning Pan & Yang (2010) leverages knowledge from a source domain to improve learning in a target domain. A representative method is Feature Transfer Learning Yin et al. (2019), which employs a center-based feature transfer framework to augment the feature space of underrepresented

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.

classes using information from data-rich classes with sufficiently diverse samples. In this way, the feature space of tail classes is enriched with greater diversity, thereby improving their representation and recognition.

The third line focuses module improvement methods that directly enhance specific components of the recognition framework. Representative methods include representation learning Cui et al. (2021), classifier design Wu et al. (2020), decoupled training Kang et al. (2020); Desai et al. (2021), and ensemble learning Cui et al. (2023); Cai et al. (2021). Notably, Kang et al. (2020) showed that a simple decoupling of representation learning and classifier learning can outperform more sophisticated approaches based on carefully designed losses or sampling strategies. This finding has motivated extensive research on decoupled long-tailed learning, which will be discussed in detail in the 2.2.

Although existing long-tailed recognition methods have achieved significant progress, existing information augmentation approaches still exhibit several limitations. First, they typically augment underrepresented classes according to class-level statistic information such as sample frequency, without considering the classification state of individual samples Shao et al. (2024). Consequently, augmentation may not be directed toward tail samples that are misclassified or poorly separated from competing classes. Second, high-quality generated features do not always benefit the classifier Li et al. (2021). Whether a generated feature actually improves classification performance on real data is rarely evaluated explicitly. More importantly, such utility is not static but may change as the classifier evolves. Features that are beneficial at one stage may later become redundant or even detrimental, whereas existing methods generally lack effective mechanisms to continuously re-evaluate and remove them.

These limitations reveal a fundamental issue in existing information augmentation: feature generation is largely performed in an open-loop manner, with limited feedback from the evolving classifier. To address this issue, we propose Classifier-Aware Feature Enrichment (CAFE), a two-stage classifier fine-tuning framework that transforms feature augmentation into a closed-loop process. Instead of treating generated features as static training data, CAFE couples sample identification, feature enrichment, and utility evaluation with the evolving classifier, allowing synthetic information to be continuously adapted throughout Stage II.

The main contributions of this work are summarized as follows:

(1) A closed-loop classifier-aware feature enrichment framework is proposed for decoupled long-tailed recognition. Unlike conventional augmentation that primarily relies on predefined class-level statistics, CAFE uses the evolving classifier to identify tail samples requiring enrichment and continuously feeds the updated classification state back into subsequent feature enrichment, establishing a closed loop between feature enrichment and classifier optimization.

(2) A synthetic feature evaluation and lifecycle management mechanism is designed to explicitly assess the semantic reliability and classifier utility of generated features before they are used for optimization. Accepted features are continuously re-evaluated as the classifier evolves, allowing redundant or detrimental features to be dynamically removed.

(3) Extensive experiments on four widely used long-tailed benchmarks demonstrate the effectiveness of CAFE across different imbalance settings, with comprehensive analyses validating its key components and dynamic feature lifecycle.

2 Related Work

2.1 Information Augmentation for Long-Tailed Recognition

Information augmentation alleviates the scarcity of tail-class data by enriching their available information. Kim et al. (2020) translates majority samples into synthetic minority samples to exploit the diversity of majority-class data. Li et al. (2021) further employs meta-learning to learn class-wise semantic augmentation with a balanced validation set. Zang et al. (2021) generates virtual features from online class-wise feature distributions and adaptively adjusts their sampling probabilities according to the training status. More recently, Shao et al. (2024) introduces diffusion models to generate diverse samples for underrepresented classes. While these methods enrich tail-class

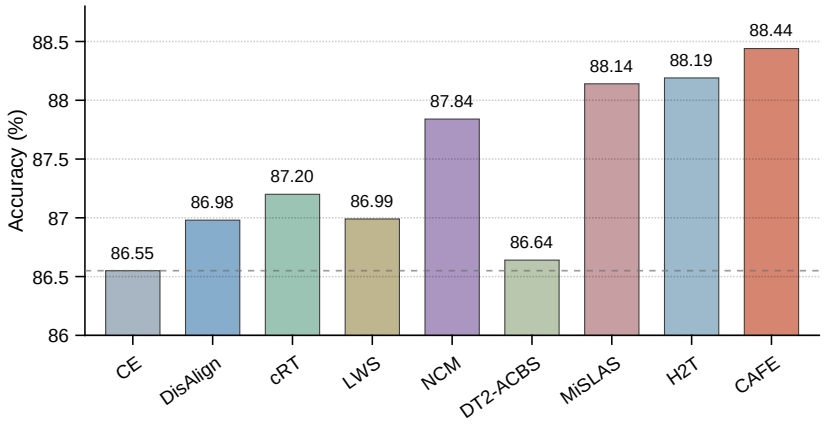


Figure 1: Comparison of CAFE with other decoupled Stage-II classifier fine-tuning methods on CIFAR10-LT with 100 IF. All methods use the same Stage-I backbone trained with CE loss.

information through different augmentation strategies, generated features are typically treated as additional training data once produced, without explicitly considering whether their utility changes as training progresses. CAFE instead introduces classifier-level utility evaluation and dynamic lifecycle management, allowing generated features to be retained or removed according to their evolving training utility.

2.2 Decoupled Learning for Long-Tailed Recognition

Decoupled learning Desai et al. (2021) has become an effective paradigm for long-tailed recognition by separating representation and classifier learning. Previous studies have shown that representations learned from imbalanced data can remain competitive, but classifier bias is an important source of performance degradation. This motivates a two-stage paradigm in which the representation is first learned from long-tailed data, followed by classifier adjustment with the backbone fixed.

Kang et al. (2020) systematically investigated classifier adjustment under the decoupled paradigm, including classifier re-training (cRT), nearest class mean (NCM), and classifier weight scaling (LWS). Zhong et al. (2021) further improves classifier learning through label-aware smoothing. More recently, Li et al. (2024) incorporated feature augmentation into the classifier adjustment stage by transferring partial semantic information from head classes to enrich tail-class features.

Despite their effectiveness, existing methods Kang et al. (2020); Desai et al. (2021); Zhong et al. (2021); Li et al. (2024); Zhang et al. (2021) mainly focus on adjusting the classifier or enriching tail-class representations. They provide limited mechanisms for determining which tail samples require enrichment according to the current classifier state and whether the introduced features remain useful as the classifier evolves.

In contrast, CAFE uses feedback from the evolving classifier to guide the identification of challenging samples and feature enrichment, while continuously evaluating the utility of generated features throughout Stage-II training. This establishes a closed-loop interaction between feature enrichment and classifier optimization. As shown in Figure 1, using the same CE-trained Stage-I model, CAFE achieves 88.44% accuracy, outperforming the compared second-stage adjustment methods.

3 Methodology

3.1 Framework Overview

Given a long-tailed training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, CAFE follows a two-stage training paradigm. In Stage I, a base model consisting of a feature extractor E and a classifier is trained on the original long-tailed data. In Stage II, the feature extractor E is frozen, and only the classifier is fine-tuned.

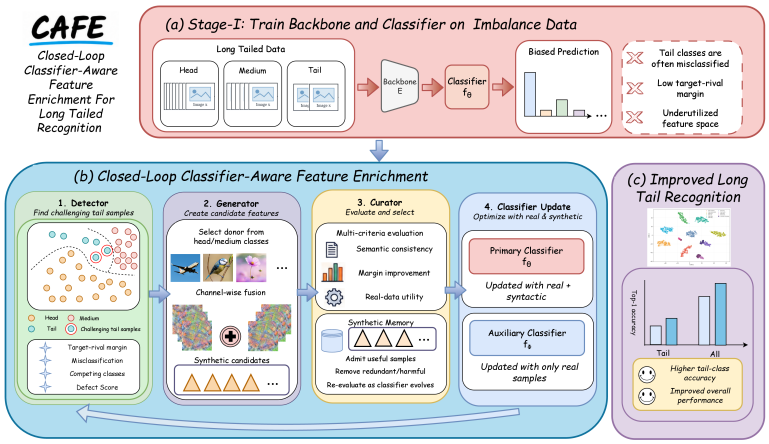


Figure 2: Framework of CAFE. CAFE consists of three core modules: the Detector, the Generator, and the Curator. The Detector identifies tail features that require enrichment, while the Generator incorporates semantic information from head/medium classes into these features. The Curator determines whether generated features should be accepted, retained, or removed. The updated classifier provides feedback to guide subsequent sample identification and feature enrichment, forming a closed loop.

For an input sample x_i, we denote its feature map and globally pooled representation as

$$M_i = E(x_i) \in \mathbb{R}^{K\times H\times W}, \qquad z_i = \mathrm{GAP}(M_i).$$

According to the number of training samples, the classes are divided into head, medium, and tail subsets, denoted by $\mathcal{H}$, $\mathcal{M}$, and $\mathcal{T}$.

As illustrated in Figure 2, CAFE performs iterative feature enrichment during Stage II. The current classifier identifies tail samples requiring enrichment, while the generated features are selectively used to update the classifier. The updated classifier then provides feedback for subsequent feature enrichment, forming a closed loop between feature enrichment and classifier optimization.

3.2 Classifier-aware Detector

Tail samples may exhibit different levels of classification risk under the current classifier. Therefore, CAFE identifies tail samples requiring enrichment according to their classification states, rather than augmenting all tail samples uniformly. To improve the reliability of this identification, we use a primary classifier f_θ and an auxiliary real-data classifier f_ϕ. The former is updated with both real features and accepted synthetic features, while the latter is updated only with real features.

For a tail feature $\boldsymbol{z}_i$ with label y_i, the averaged logits are defined as

$$\bar{f}(\boldsymbol{z}_i) = \frac{1}{2}\left(f_\theta(\boldsymbol{z}_i) + f_\phi(\boldsymbol{z}_i)\right). \tag{1}$$

The strongest rival class and the target–rival margin are respectively given by

$$r_i = \arg\max_{c\neq y_i} \bar{f}(\boldsymbol{z}_i)_c, \tag{2}$$

$$m_i = \bar{f}(\boldsymbol{z}_i)_{y_i} - \bar{f}(\boldsymbol{z}_i)_{r_i}. \tag{3}$$

A non-positive m_i indicates misclassification, while a small positive m_i indicates weak separation from the strongest rival class.

We compute an adaptive threshold from the margins of real tail features:

$$\tau = Q_q\left(\{m_i \mid y_i \in \mathcal{T}\}\right), \tag{4}$$

where $Q_q(\cdot)$ denotes the q-th quantile operator. Thus, τ serves as an adaptive margin threshold determined by the q-th percentile of real tail-sample margins. We then define the defect score as

$$s_i = \tau - m_i. \tag{5}$$

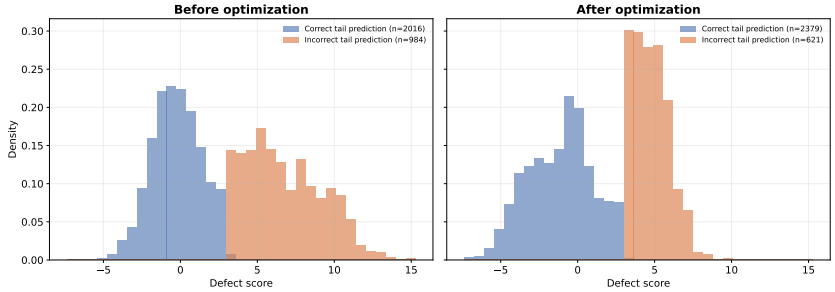


Figure 3: Distributions of defect scores. A large defect score indicates a sample that is closer to the decision boundary or more likely to be misclassified.

A larger s_i indicates a smaller margin and higher classification risk. The Detector selects misclassified samples with $m_i \leq 0$ and correctly classified but low-margin samples satisfying

$$0 < m_i \leq \tau^+, \qquad \tau^+ = \max(\tau, m_{\min}). \tag{6}$$

The selected samples, together with their target classes, rival classes, margins, and defect scores, are passed to the Generator.

As shown in Figure 3, incorrectly classified tail samples generally exhibit higher defect score than correctly classified ones. After optimization, the overall defect score distribution shifts toward lower values, indicating larger classification margins for tail-class samples and reduced classification risk.

3.3 Generator

For each tail feature identified by the Detector, the Generator constructs candidate enriched features by incorporating information from head and medium classes. The selected tail feature map is referred to as the target, while the feature map providing additional information is referred to as the donor.

3.3.1 Donor Selection

Given a target class t and its rival class r, the donor class d is selected from the head and medium classes according to

$$S(d \mid t, r) = \boldsymbol{p}_t^\top \boldsymbol{p}_d + \lambda \mathbb{I}(d = r), \tag{7}$$

where $\boldsymbol{p}_t$ and $\boldsymbol{p}_d$ are the normalized class prototypes, and λ controls the preference for the rival class. Since the rival class reflects the primary confusion direction of the current classifier, this preference encourages candidate generation to explore feature variations relevant to the target–rival confusion.

3.3.2 Channel-wise Soft Fusion

Let F_t and F_d denote the target and donor feature maps, respectively. A candidate feature map is generated as

$$\widetilde{F}_t = F_t + \alpha_t M \odot (F_d - F_t), \tag{8}$$

where M is the channel-selection mask, $\odot$ denotes element-wise multiplication, and $\alpha_t \in (0, 1]$ is the fusion strength. Unselected channels remain unchanged, while selected channels are shifted toward the corresponding donor channels according to α_t. The Generator produces multiple candidate features using different donor samples or channel masks, which are forwarded to the Curator for evaluation.

3.4 Curator

The Generator produces candidate features, but generation quality alone does not guarantee their utility to the classifier. The Curator therefore evaluates candidates based on semantic validity, real-data utility, and their effectiveness during subsequent training.

3.4.1 CANDIDATE VALIDATION

Let

$$\mathcal{C} = \{(\widetilde{\boldsymbol{z}}_i, z_i, y_i, r_i)\}_{i=1}^{N_c}$$

denote the candidate set, where $\widetilde{\boldsymbol{z}}_i$ is a generated candidate feature and $\boldsymbol{z}_i$ denotes its corresponding original tail feature identified by the Detector. For each candidate, we measure its cosine similarity c_i to the original tail feature and its relative feature drift d_i:

$$c_i = \frac{\widetilde{\boldsymbol{z}}_i^\top \boldsymbol{z}_i}{\|\widetilde{\boldsymbol{z}}_i\|_2 \|\boldsymbol{z}_i\|_2}, \qquad d_i = \frac{\|\widetilde{\boldsymbol{z}}_i - \boldsymbol{z}_i\|_2}{\|\boldsymbol{z}_i\|_2}. \tag{9}$$

Let $\boldsymbol{p}_c$ denote the normalized prototype of class c. The prototype margin of a candidate is defined as

$$g_i^{\text{proto}} = \widetilde{\boldsymbol{z}}_i^\top \boldsymbol{p}_{y_i} - \max_{c \neq y_i} \widetilde{\boldsymbol{z}}_i^\top \boldsymbol{p}_c. \tag{10}$$

A larger g_i^{proto} indicates stronger association with the target-class prototype compared with the other classes.

A candidate is considered valid if it preserves the original feature structure, remains associated with the target class, and improves the target–rival margin under both classifiers:

$$\mathcal{C}_{\text{valid}} = \left\{ (\widetilde{\boldsymbol{z}}_i, y_i, r_i) \in \mathcal{C} \;\middle|\; c_i \geq \tau_{\cos},\ d_i \leq \tau_{\text{drift}},\ g_i^{\text{proto}} \geq \tau_{\text{proto}},\ \text{Consistent}(\widetilde{\boldsymbol{z}}_i) \right\}. \tag{11}$$

Here, $\text{Consistent}(\widetilde{\boldsymbol{z}}_i)$ requires both classifiers to confirm a valid improvement in the target–rival margin of the candidate. This step removes candidates that are excessively distorted or lose their target-class identity.

3.4.2 REAL-DATA UTILITY EVALUATION

The valid candidates are further evaluated using the auxiliary classifier f_ϕ, which is trained only on real samples. For a candidate subset $\mathcal{C}' \subseteq \mathcal{C}_{\text{valid}}$, we first perform a virtual update:

$$\phi' = \phi - \eta_s \nabla_\phi \mathcal{L}_{\text{syn}}(\mathcal{C}'). \tag{12}$$

The effect of this update is evaluated on a support set $\mathcal{S}$ constructed by sampling real training features as evenly as possible across classes. The class-balanced classification risk is defined as

$$R_{\text{real}}(f; \mathcal{S}) = \frac{1}{|\mathcal{C}_\mathcal{S}|} \sum_{c \in \mathcal{C}_\mathcal{S}} \frac{1}{|\mathcal{S}_c|} \sum_{(\boldsymbol{z}_i, y_i) \in \mathcal{S}_c} \ell_{\text{CE}}(f(\boldsymbol{z}_i), y_i). \tag{13}$$

The resulting risk reduction is

$$\Delta R_{\text{real}} = R_{\text{real}}(f_\phi; \mathcal{S}) - R_{\text{real}}(f_{\phi'}; \mathcal{S}). \tag{14}$$

The risk reductions for head, medium, and tail classes, denoted by ΔR_{head}, ΔR_{medium}, and ΔR_{tail}, respectively, are computed analogously on the corresponding class subsets.

Candidate subsets satisfying the following conditions are included in the accepted set $\mathcal{A}_t$:

$$\Delta R_{\text{real}} > 0, \quad \Delta R_{\text{tail}} > 0, \quad \Delta R_{\text{head}} \geq -\epsilon_{\text{head}}, \quad \Delta R_{\text{medium}} \geq -\epsilon_{\text{medium}}. \tag{15}$$

The accepted candidates form the set $\mathcal{A}_t$, which is used to update the synthetic memory.

3.4.3 SYNTHETIC MEMORY MAINTENANCE

The Curator maintains a bounded memory of accepted synthetic features for each class. At t-th epoch, the newly accepted candidates $\mathcal{A}_t$ are merged with the memory $\mathcal{M}_{t-1}$ retained from the previous epoch:

$$\mathcal{M}_{t-1} \cup \mathcal{A}_t \xrightarrow{\text{Update}} \mathcal{M}_t, \tag{16}$$

where Update denotes the memory maintenance process. The merged features are first filtered by cosine similarity to remove highly similar features. The retained features are then continuously re-evaluated as the classifier evolves, allowing features that no longer provide sufficient utility to be removed from the memory.

Algorithm 1: Training Procedure of CAFE

Input: Long-tailed training set $\mathcal{D}$; Stage-I backbone E and classifier;
Output: Trained classifier f_θ;

Load the Stage-I model and freeze the backbone E;
Initialize the primary classifier f_θ and auxiliary classifier f_ϕ;
Initialize the synthetic memory $\mathcal{M}_0 \leftarrow \emptyset$;
for $t = 1$ **to** T **do**
    Update f_θ with real samples: $\theta \leftarrow \theta - \eta_r \nabla_\theta \mathcal{L}_{\text{real}}$;
    Update f_ϕ independently using real samples only;
    $\mathcal{D}_t^{\text{def}} \leftarrow \text{Detector}(f_\theta, f_\phi)$;
    $\mathcal{C}_t \leftarrow \text{Generator}(\mathcal{D}_t^{\text{def}})$;
    $\mathcal{M}_t \leftarrow \text{Curator}(\mathcal{M}_{t-1}, \mathcal{A}_t, f_\phi)$;
    Update f_θ with accepted synthetic features: $\theta \leftarrow \theta - \eta_s \nabla_\theta \mathcal{L}_{\text{syn}}(\mathcal{M}_t)$;
return f_θ;

3.5 Training Procedure

Algorithm 1 summarizes the Stage-II training procedure of CAFE, including classifier updating with real data, tail-sample detection, feature generation and evaluation, and classifier updating with accepted synthetic features.

During Stage II, the backbone E is frozen, while the primary classifier f_θ and auxiliary classifier f_ϕ are initialized from the Stage I. The primary classifier is optimized by using both real features and accepted synthetic features, whereas the auxiliary classifier is updated only with real features.

At each epoch, both classifiers are first updated independently on real data. The Detector identifies tail samples requiring enrichment, and the Generator constructs candidate features. The Curator then evaluates these candidates and updates the synthetic memory $\mathcal{M}_t$, which will be subsequently used to further optimize the primary classifier. The classifier update is given by

$$\theta_{t+\frac{1}{2}} = \theta_t - \eta_r \nabla_\theta \mathcal{L}_{\text{real}}, \tag{17}$$

$$\theta_{t+1} = \theta_{t+\frac{1}{2}} - \eta_s \nabla_\theta \mathcal{L}_{\text{syn}}(\mathcal{M}_t), \tag{18}$$

where η_r and η_s denote the learning rates for classifier updates using real and synthetic features, respectively. This process is repeated for T epochs to obtain the final classifier.

4 Experiment

4.1 Datasets and Metrics

We evaluate on four widely used benchmarks: CIFAR10-LT/CIFAR100-LT, ImageNet-LT Liu et al. (2024b), and iNaturalist2018 Van Horn et al. (2018). The CIFAR-LT datasets use imbalance factor (IF)=10,50,100, while ImageNet-LT and iNaturalist2018 have IF=256 and 500, respectively. Following Cui et al. (2021), we report top-1 accuracy for all classes and three frequency groups: head ($n_i > 100$), medium ($20 < n_i \leq 100$), and tail ($n_i \leq 20$). Full comparisons are provided in Appendix A.9.

4.2 Settings

For the training in Stage I, the models trained with CE and BCE3S Fan et al. (2026) on CIFAR10-LT and CIFAR100-LT are trained for 200 epochs. On ImageNet-LT and iNaturalist 2018, ProCo Du et al. (2024) is trained for 90 epochs according to its original training protocol, while GLMC Du et al. (2023) is trained for 180 epochs following the original paper. For decoupled-learning methods, the backbone is kept fixed during Stage II and the classifier is fine-tuned for 20 epochs under the unified protocol. Further implementation details and hyperparameter settings are provided in the Appendix A.3.

Table 1: Top-1 accuracy (%) on CIFAR10-LT and CIFAR100-LT with ResNet-32 He et al. (2016) under different imbalance factors.

Method	CIFAR10-LT			CIFAR100-LT		
	10	50	100	10	50	100
CE	86.55	77.89	70.57	56.73	44.70	40.24
MiSLAS Zhong et al. (2021)	89.35	83.85	80.26	62.20	51.89	47.02
BCL Zhu et al. (2022)	91.39	86.70	84.22	64.08	54.86	51.57
SBCL Hou et al. (2023)	–	–	–	57.90	48.70	44.90
WCADS Han (2023)	92.40	–	86.40	65.80	–	52.50
ProCo Du et al. (2024)	91.90	88.20	85.90	65.50	57.10	52.80
ECL Xu et al. (2024)	91.80	88.90	86.50	67.30	59.90	56.30
ICL Chen et al. (2025)	91.30	88.50	86.40	69.00	59.70	56.30
LOS Sun et al. (2025)	–	–	–	69.70	58.80	54.90
BCE3S Fan et al. (2026)	93.85	89.41	85.25	71.93	60.40	54.70
CE + CAFE	88.44	79.71	72.87	58.43	45.89	41.59
BCE3S + CAFE	**94.65**	**90.88**	**87.76**	**73.10**	**61.72**	**56.50**

Table 2: Top-1 accuracy (%) on ImageNet-LT using ResNeXt-50 Xie et al. (2017).

Method	Many	Med.	Few	All
RIDE(4Exp) Wang et al. (2021)	–	–	–	56.80
ACE(3Exp) Cai et al. (2021)	–	–	–	56.60
GML Suh & Seo (2023)	68.70	55.70	38.60	58.30
BCL Zhu et al. (2022)	67.90	54.20	36.60	57.10
ResLT Cui et al. (2023)	63.60	55.70	38.90	56.10
GLMC Du et al. (2023)	**70.10**	52.40	30.40	56.30
GLMC + BS Du et al. (2023)	64.76	55.67	**42.19**	57.21
ProCo Du et al. (2024)	–	–	–	58.00
GLMC + CAFE	65.94	**57.20**	37.46	57.88
ProCo + CAFE	68.60	56.18	39.72	**58.72**

4.3 Comparison with SOTA

4.3.1 CIFAR10-LT and CIFAR100-LT

As shown in Table 1, CAFE consistently improves different Stage-I baselines across imbalance settings, demonstrating its compatibility with different feature representations. In particular, BCE3S+CAFE achieves the best performance in all six settings, reaching 94.65%, 90.88%, and 87.76% on CIFAR10-LT and 73.10%, 61.72%, and 56.50% on CIFAR100-LT under IF=10,50,100, respectively. The improvement become particularly evident under severe imbalance. For example, BCE3S+CAFE improves BCE3S by 2.51 and 1.80 percentages on CIFAR10-LT and CIFAR100-LT at IF=100, respectively. These results demonstrate the effectiveness of CAFE in further refining the classifier under long-tailed distributions.

4.3.2 ImageNet-LT

As shown in Table 2, CAFE remains effective on large-scale ImageNet-LT. With GLMC, it improves medium/few-shot accuracy from 52.40%/30.40% to 57.20%/37.46%, while raising overall accuracy from 56.30% to 57.88%. ProCo+CAFE further achieves the best overall accuracy (58.72%), which demonstrates CAFE's effectiveness in improving underrepresented classes on large-scale datasets.

4.3.3 iNaturalist 2018

On iNaturalist 2018, ProCo + CAFE achieves the highest overall accuracy (74.19%), outperforming the original ProCo by 0.69%. It also obtains the best medium-shot accuracy (74.91%) while maintaining a competitive few-shot accuracy (73.74%). The consistent performance across different class groups further demonstrates the effectiveness of CAFE on large-scale long-tailed recognition.

Table 3: Top-1 accuracy (%) on iNaturalist 2018 using ResNet-50 He et al. (2016).

Method	Many	Med.	Few	All
BalPoE Aimar et al. (2023)	–	–	–	73.50
ResLT Cui et al. (2023)	73.00	72.60	73.10	72.90
DBM–GML Son et al. (2025)	66.90	71.90	73.60	72.00
RIDE + RL Zhao et al. (2024)	71.40	70.90	**74.80**	73.60
WCDAS Han (2023)	75.50	72.30	69.80	71.80
SBCL Hou et al. (2023)	73.30	71.90	68.60	70.80
BCE3S Fan et al. (2026)	**77.16**	74.45	72.78	73.99
ProCo Du et al. (2024)	–	–	–	73.50
ProCo + CAFE	72.41	**74.91**	73.74	**74.19**

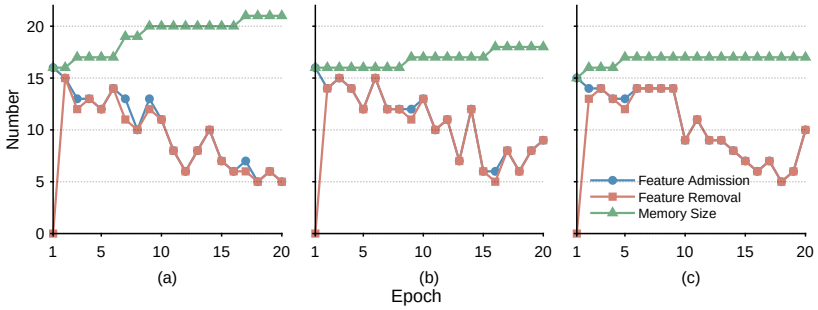


Figure 4: Dynamics of the synthetic feature lifecycle on CIFAR10-LT under (a) IF=10, (b) IF=50, and (c) IF=100.

4.4 Lifecycle

Figure 4 illustrates the dynamic lifecycle of synthetic features during the training on Stage II. Across all imbalance factors, the synthetic memory rapidly reaches a relatively stable size, whereas feature admission and removal remain active throughout the optimization process. In particular, even when the memory size remains unchanged, newly admitted features are continuously accompanied by the removal of previously retained ones. This behavior shows that CAFE does not simply accumulate synthetic features. Instead, the composition of the synthetic memory is continuously revised as the classifier evolves. Moreover, the decrease in feature admission and removal during later epochs indicates that the synthetic feature set becomes stable as training progresses.

5 Conclusion

In this paper, we proposed CAFE, a closed-loop classifier-aware feature enrichment framework for long-tailed recognition. CAFE uses the evolving classifier to identify tail samples requiring enrichment and performs feature augmentation using information transferred from data-rich classes. Rather than directly using generated features for classifier optimization, CAFE evaluates their semantic reliability and classifier utility and continuously re-evaluates accepted features to remove redundant or detrimental ones. The updated classifier further guides subsequent sample identification and feature enrichment, establishing a closed loop between feature enrichment and classifier optimization. Extensive experiments demonstrate that CAFE consistently improves different base classifiers and achieves competitive or superior performance on long-tailed benchmarks.

AI use statement

In this work, we used generative AI tools to assist with language editing, including improving grammar, clarity, and readability, and to provide limited assistance with code debugging and refinement. The authors take full responsibility for the final content of this work, including all text, claims, code, experimental results, and conclusions produced with the aid of generative AI.

Ethics statement

This work focuses on methodological research in long-tailed visual recognition and is evaluated on publicly available benchmark datasets. It does not involve human-subject studies, collection of private or sensitive information, or deployment in high-risk applications. We follow standard research practices regarding data usage, experimental reporting, and reproducibility. To the best of our knowledge, this work does not raise significant ethical, privacy, security, or societal concerns.

Reproducibility statement

We have made efforts to ensure the reproducibility of this work. The proposed method and its key components are described in detail in the main paper. Experimental settings, dataset configurations, implementation details, and hyperparameter choices are provided in the experimental section and appendix. Additional ablation studies and analyses are included to clarify the contribution and behavior of individual components.

Acknowledgments

The authors acknowledge the support of several national, provincial, municipal, and institutional research funding programs. Detailed funding information will be provided in the final version.

References

Emanuel Sanchez Aimar, Arvi Jonnarth, Michael Felsberg, and Marco Kuhlmann. Balanced Product of Calibrated Experts for Long-Tailed Recognition . In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19967–19977, 2023.

Yuxiang Bao, Guoliang Kang, Linlin Yang, Xiaoyue Duan, Bo Zhao, and Baochang Zhang. Normalizing batch normalization for long-tailed recognition. *IEEE Transactions on Image Processing*, 34:209–220, 2025.

Jiarui Cai, Yizhou Wang, and Jenq-Neng Hwang. Ace: Ally complementary experts for solving long-tailed recognition in one-shot. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 112–121, 2021.

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.

Tao Chen, Lijie Wang, Yang Liu, and Haisheng Yu. Dacbn: Dual attention convolutional broad network for fine-grained visual recognition. *Pattern Recognition*, 156:110749, 2024.

Yanhao Chen, Zhongquan Jian, Nianxin Ke, Shuhao Hu, Junjie Jiao, Qingqi Hong, and Qingqiang Wu. Enhancing mixture of experts with independent and collaborative learning for long-tail visual recognition. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 2025.

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.

Jiequan Cui, Shu Liu, Zhuotao Tian, Zhisheng Zhong, and Jiaya Jia. Reslt: Residual learning for long-tailed recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 3695–3706, 2023.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-Balanced Loss Based on Effective Number of Samples . In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9260–9269, 2019.

Alakh Desai, Tz-Ying Wu, Subarna Tripathi, and Nuno Vasconcelos. Learning of visual relations: The devil is in the tails. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

Chaoqun Du, Yulin Wang, Shiji Song, and Gao Huang. Probabilistic contrastive learning for long-tailed visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Fei Du, Peng Yang, Qi Jia, Fengtao Nan, Xiaoting Chen, and Yun Yang. Global and local mixture consistency cumulative learning for long-tailed visual recognitions. In *Conference on Computer Vision and Pattern Recognition 2023*, 2023.

Weijia Fan, Qiufu Li, Jiajun Wen, and Xiaoyang Peng. Bce3s: Binary cross-entropy based tripartite synergistic learning for long-tailed recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 3795–3803, 2026.

Boran Han. Wrapped cauchy distributed angular softmax for long-tailed visual recognition. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Pengxiao Han, Changkun Ye, Jieming Zhou, Jing Zhang, Jie Hong, and Xuesong Li. Latent-based Diffusion Model for Long-tailed Recognition . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2639–2648, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition . In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Chengkai Hou, Jieyu Zhang, Haonan Wang, and Tianyi Zhou. Subclass-balancing contrastive learning for long-tailed recognition. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5372–5384, 2023.

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.

Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13893–13902, 2020.

Jun Li, Zichang Tan, Jun Wan, Zhen Lei, and Guodong Guo. Nested collaborative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6949–6958, 2022.

Mengke Li, Zhikai Hu, Yang Lu, Weichao Lan, Yiu-ming Cheung, and Hui Huang. Feature fusion from head to tail for long-tailed visual recognition. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, 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 *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5208–5217, 2021.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, 2017.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020.

Jialun Liu, Wenhui Li, and Yifan Sun. Memory-based jitter: Improving visual recognition on long-tailed data with diversity in memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 1720–1728, 2022.

Taiheng Liu, Zhaoshui He, Zhijie Lin, Guang-Zhong Cao, Wenqing Su, and Shengli Xie. An adaptive image segmentation network for surface defect detection. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6):8510–8523, 2024a.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Open Long-Tailed Recognition in a Dynamic World . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 46(03):1836–1851, 2024b. ISSN 1939-3539.

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.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6877–6886, 2022.

Harsh Rangwani, Pradipto Mondal, Mayank Mishra, Ashish Ramayee Asokan, and R. Venkatesh Babu. Deit-lt: Distillation strikes back for vision transformer training on long-tailed datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23396–23406, June 2024.

Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020.

Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J. Anders, and Klaus-Robert Müller. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3):247–278, 2021.

Jie Shao, Ke Zhu, Hanxiao Zhang, and Jianxin Wu. Diffult: Diffusion for long-tail recognition without external knowledge. In *Advances in Neural Information Processing Systems*, volume 37, pp. 123007–123031, 2024.

Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.

Minseok Son, Inyong Koo, Jinyoung Park, and Changick Kim. Difficulty-aware balancing margin loss for long-tailed recognition. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, 2025.

Min-Kook Suh and Seung-Woo Seo. Long-tailed recognition by mutual information maximization between latent features and ground-truth labels. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

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

Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S. Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329, 2017.

Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*, 2020.

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 (CVPR)*, pp. 8769–8778, 2018.

Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations*, 2021.

Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. Dynamic curriculum learning for imbalanced data classification. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5016–5025, 2019.

Tz-Ying Wu, Pedro Morgado, Pei Wang, Chih-Hui Ho, and Nuno Vasconcelos. Solving long-tailed recognition with deep realistic taxonomic classifier. In *European Conference on Computer Vision (ECCV)*, 2020.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995, 2017.

Zhengzhuo Xu, Zenghao Chai, Chengyin Xu, Chun Yuan, and Haiqin Yang. Towards effective collaborative learning in long-tailed recognition. *IEEE Transactions on Multimedia*, 26:3754–3764, 2024.

Shiyu Xuan and Shiliang Zhang. Decoupled contrastive learning for long-tailed recognition. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, 2024.

Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Feature transfer learning for face recognition with under-represented data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5697–5706, 2019.

Yuhang Zang, Chen Huang, and Chen Change Loy. FASA: Feature Augmentation and Sampling Adaptation for Long-Tailed Instance Segmentation . In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3437–3446, 2021.

Enhao Zhang, Chuanxing Geng, and Songcan Chen. Class-aware universum inspired re-balance learning for long-tailed recognition. *Pattern Recognition*, 161:111337, 2025.

Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *CVPR*, 2021.

Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 45(09):10795–10816, 2023.

Qihao Zhao, Yalun Dai, Shen Lin, Wei Hu, Fan Zhang, and Jun Liu. Ltrl: Boosting long-tail recognition via reflective learning. In *ECCV 2024*, pp. 1–18, 2024.

Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

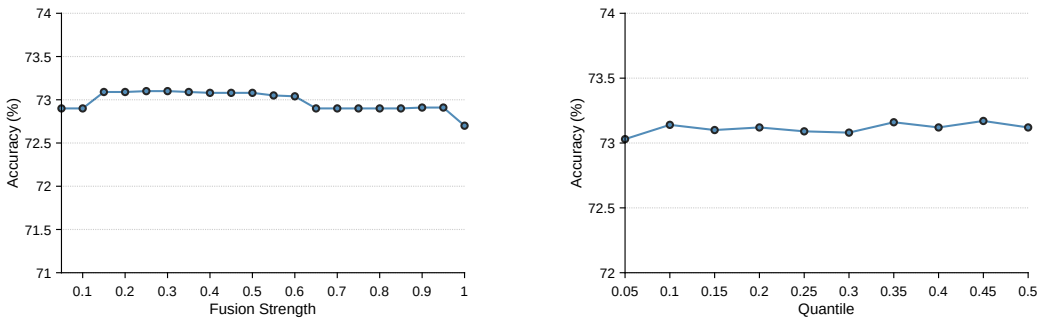
Jianggang Zhu, Zheng Wang, Jingjing Chen, Yi-Ping Phoebe Chen, and Yu-Gang Jiang. Balanced contrastive learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6908–6917, 2022.

A Appendix

A.1 Hyperparameter Sensitivity

We investigate the sensitivity of CAFE to two key hyperparameters: the fusion strength α in the Generator and the quantile q in the Detector. The former controls the amount of information transferred from the donor feature during feature generation, while the latter determines the margin threshold to identifying challenging tail samples.

As shown in Figure 5-(a), CAFE maintains relatively stable performance over a broad range of fusion strengths. Moderate values of α generally achieve better performance, whereas an excessively large fusion strength leads to a noticeable degradation. This suggests that moderate feature transfer can introduce useful feature variations while preserving the semantics of the target sample, whereas



(a) (b)

Figure 5: (a): Sensitivity of CAFE to the fusion strength α in the Generator. (b): Sensitivity of CAFE to the defect-selection quantile q in the Detector.

overly aggressive fusion may introduce excessive donor information and compromise class consistency.

Figure 5-(b) further reports the sensitivity to the defect quantile q. The accuracy varies slightly as q changes from 0.05 to 0.50, remaining within a narrow range between 73.03%–73.17%. This result indicates that CAFE is relatively insensitive to the precise choice of the defect-selection quantile and does not rely on a narrowly tuned detection threshold.

A.2 Ablation Study

We conduct a systematic ablation study to investigate the contribution of each component in the Stage-II classifier fine-tuning of CAFE. Stage I is trained with cross-entropy on CIFAR10-LT, and the experiments are conducted under imbalance factors of 50 and 100. We construct seven variants, denoted as V_0–V_6, as follows:

- V_0: **Stage-I baseline.** The original model trained by CE without Stage-II classifier fine-tuning.
- V_1: **Classifier fine-tuning.** The backbone is frozen and only the classifier is fine-tuned using real training features, without introducing any synthetic features.
- V_2: **Without classifier-aware detection.** The Generator and Curator are retained, while the classifier-aware Detector is removed. Real tail samples are randomly selected for synthetic feature generation.
- V_3: **Without feature generation.** In this variant, the detector selects defective tail-class samples, but the generator is disabled. The selected real features are directly replayed during classifier fine-tuning, while the curator remains active. Comparing this variant with the full CAFE model measures whether synthetic feature generation offers additional benefits beyond replaying difficult real samples.
- V_4: **Without feature curation.** The Detector identifies defective tail samples and the Generator produces synthetic features, but all generated features are unconditionally used for classifier fine-tuning without Curator screening or subsequent removal. This variant evaluates the effect of synthetic feature generation without controlling the quality and utility .
- V_5: **Without dynamic lifecycle management.** Newly generated features are still screened by the Curator before being added to the synthetic memory. Unlike the full CAFE, however, accepted features are retained throughout the remaining Stage-II epochs without further re-evaluation or removal. This variant evaluates the effect of dynamic re-evaluation as the classifier evolves.
- V_6: **Full CAFE.** The complete framework.

These variants allow the individual components to be compared. Specifically, V_0 versus V_1 measures the benefit of Stage-II classifier fine-tuning; V_2 versus V_6 evaluates the contribution of classifier-aware defect detection; V_3 versus V_6 evaluates the benefit of synthetic feature generation; V_4 versus

V_6 evaluates the contribution of feature curation; and V_5 versus V_6 isolates the effect of dynamic lifecycle management.

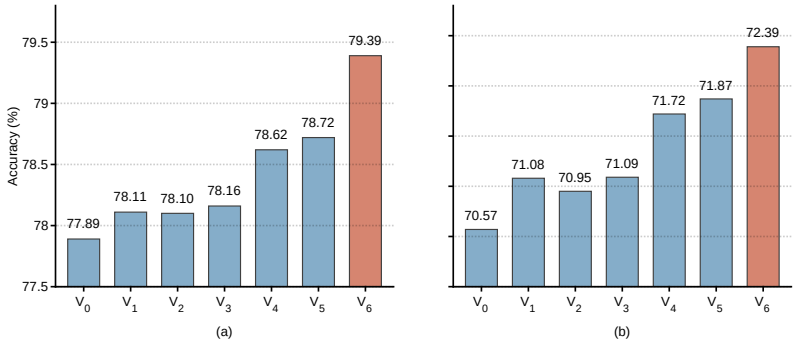


Figure 6: Ablation study of CAFE on CIFAR10-LT under different imbalance factors: (a) IF=50 and (b) IF=100. V_0–V_6 denote different variants of CAFE, with V_6 representing the complete framework, i.e. CAFE.

As shown in Figure 6, the complete CAFE (i.e.V_6) consistently achieves the best performance under both imbalance factors, indicating that the Detector, Generator, Curator, and dynamic lifecycle management provide complementary benefits. Notably, V_6 improves upon V_5 by 0.67% and 0.52% under IF=50 and IF=100, respectively, demonstrating that permanently retaining accepted synthetic features is suboptimal and that their utility should be continuously re-evaluated as the classifier evolves.

Moreover, replacing classifier-aware defect detection with random tail-sample selection (V_2) provides almost no improvement over classifier fine-tuning (V_1) at IF=50 and even slightly degrades the performance at IF=100, highlighting the importance of localizing samples according to the current classifier state rather than performing random tail feature augmentation.

A.3 Settings

For Stage-II training, the backbone is frozen and only the classifiers are updated. Both the primary classifier f_θ and the auxiliary real-data classifier f_ϕ are optimized using SGD with a momentum of 0.9. The initial real-data learning rate η_r is set to 10^{-3} for CIFAR10-LT and 10^{-4} for the other datasets, and is decayed using a cosine schedule over 20 epochs.

The primary classifier is additionally updated using the synthetic features retained in the memory. The learning rate for this update is defined as

$$\eta_s = \lambda_{\text{syn}} \max\left(\rho_{\text{syn}}\eta_r, \eta_{\text{floor}}\right),$$

where λ_{syn}, ρ_{syn}, and η_{floor} are set independently for each dataset. The auxiliary classifier is updated only with real features.

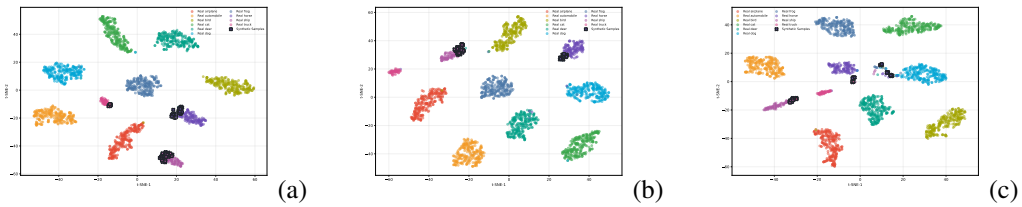
After each synthetic feature update, MaxNorm with a threshold of 0.1 is applied to the classifier. The fusion strength α_t in the Generator is set to 0.15. The quantile q in the Detector is set to 0.35.

Experiments on CIFAR10-LT and CIFAR100-LT are conducted using an NVIDIA RTX 4060 GPU, while experiments on ImageNet-LT and iNaturalist 2018 are conducted using an NVIDIA RTX 4090 GPU.

A.4 Visualization

To further investigate the behavior of CAFE, we conduct feature visualization. Specifically, t-SNE is employed to visualize the distribution of real and synthetic features.

The synthetic samples are mainly distributed within or near the real feature clusters rather than being randomly scattered in the embedding space, which indicates that the generated features largely



(a) (b) (c)

Figure 7: t-SNE visualization of the feature representations after CAFE optimization on CIFAR-10-LT under different imbalance factors. (a) IF $= 10$, (b) IF $= 50$, and (c) IF $= 100$. Each color represents one class.

preserve the semantic structures of their target classes. Meanwhile, some synthetic samples appear near cluster boundaries or relatively sparse regions, providing additional variations for tail samples. These observations suggest that CAFE can generate class-consistent features while enriching the feature distribution of tail classes.

A.5 Computational Cost Analysis

Table 4: Computational cost comparison of different second-stage classifier fine-tuning methods on CIFAR10-LT with IF=10. All methods use the same Stage-I backbone trained with cross-entropy loss.

Metric	**DisAlign**	**cRT**	**LWS**	**NCM**	**DT2-ACBS**	**MiSLAS**	**H2T**	**CAFE**
Time (s)	287.3	52.7	80.1	25.0	586.1	577.6	1071.5	240.38
GPU Peak Memory (MiB)	63.3	48.0	48.0	48.0	63.3	63.3	74.1	222.0
Stage II Params.	85	650	10	0	650	660	650	650
FLOPs/Update (M)	0.017696	0.492	0.005	0	0.017696	0.017696	0.035392	0.030720

We evaluate the computational cost of CAFE by comparing it with representative Stage-II classifier fine-tuning methods. All methods are evaluated on CIFAR10-LT with an imbalance factor of 10, using the same Stage-I backbone trained with cross entropy loss. We report the total training time in Stage II (20 epoches), peak GPU memory consumption, number of trainable parameters during classifier fine-tuning, and FLOPs per update.

As shown in Table 4, CAFE requires 240.38s for Stage-II training, which is lower than DisAlign, DT2-ACBS, MiSLAS, and H2T. CAFE exhibits higher peak GPU memory consumption, mainly due to the storage and evaluation of synthetic features. Nevertheless, its FLOPs for each update remain comparable to those of representative classifier fine-tuning methods, indicating that the additional computational cost of CAFE mainly arises from synthetic feature management rather than increased model parameterization.

A.6 Channel-wise Analysis of Feature Generation

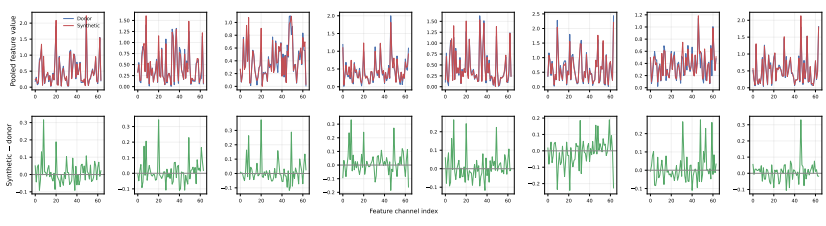


Figure 8: Channel-wise comparison between donor and synthetic features. The upper row presents the pooled feature values of donor and synthetic features, while the lower row shows their channel-wise differences.

Figure 8 visualizes the channel-wise values of the donor and synthetic features. The upper row compares their pooled feature values across different channels, while the lower row presents the corresponding channel-wise differences. The donor and synthetic features remain similar across many channels, while noticeable differences are concentrated in a subset of channels. This pattern is consistent with the channel-wise fusion mechanism of CAFE, which modifies selected channels while largely preserving the remaining feature representation. Overall, the visualization illustrates the localized changes introduced during synthetic feature generation.

A.7 Head–Tail Trade-off Analysis

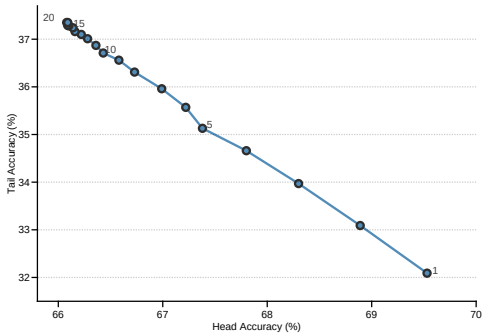


Figure 9: Head–tail accuracy trade-off during CAFE Stage-II classifier fine-tuning on ImageNet-LT. The Stage-I model is trained with GLMC using a ResNeXt-50 backbone. Each point corresponds to one Stage-II epoch, with selected epochs annotated.

We investigate the trade-off between head- and tail-class performance during classifier fine-tuning in Stage II. The analysis is conducted on ImageNet-LT using ResNeXt-50 as the backbone, where the model in Stage I is trained with GLMC and subsequently fine-tuned with CAFE in Stage II. Here, the head and tail groups correspond to the Many and Few groups, respectively.

As shown in Figure 9, each point represents the classifier state at one epoch in Stage II. As fine-tuning progresses, tail accuracy consistently increases while head accuracy gradually decreases, revealing a clear redistribution of classification performance from data-rich head classes toward underrepresented tail classes. The trajectory also becomes concentrated in later epochs, indicating that the head–tail trade-off gradually stabilizes as the classifier converges.

Starting from the GLMC model in Stage I, CAFE improves the tail-class accuracy from 31.01% to 37.35%, while the head-class accuracy changes from 69.88% to 66.09%, demonstrating that the substantial improvement on underrepresented classes is accompanied by a controlled sacrifice in head-class performance.

A.8 Utility Analysis of Accepted and Rejected Candidates

Table 5: Classifier-level utility of synthetic candidates accepted and rejected by the Curator. ΔAcc. denotes the mean accuracy change after the virtual classifier update in percentage points (pp), while PUR denotes the proportion of candidates producing a positive change in the corresponding metric.

Candidate Set	**Overall**		**Tail**		$\boldsymbol{\Delta m}$
	ΔAcc. (pp)	**PUR**	**ΔAcc. (pp)**	**PUR**	
Accepted	$+\mathbf{0.0188}$	$\mathbf{75.00}\%$	$+0.0458$	$\mathbf{75.00}\%$	$\mathbf{0.063402}$
Rejected	-0.0001	33.80%	$+\mathbf{0.0526}$	53.52%	0.055215

We investigate whether the Curator effectively distinguishes synthetic candidates according to their actual classifier-level utility. In addition to the mean change induced by each candidate, we report

the Positive Utility Ratio (PUR), which denotes the proportion of candidates that produce a positive change in the corresponding metric after the virtual classifier update.

As shown in Table 5, accepted candidates yield a positive mean overall accuracy change of +0.0188%, with 75.00% of them improving the overall accuracy. In contrast, rejected candidates produce an almost neutral but slightly negative mean change of -0.0001%, and only 33.80% exhibit positive overall utility. Although the rejected candidates show a slightly mean tail accuracy improvement (+0.0526% versus +0.0458%), their tail PUR is considerably lower (53.52% versus 75.00%) than accepted ones.

Interestingly, both accepted and rejected candidates produce positive target–rival margin changes in all evaluated cases, while their overall accuracy utilities differ substantially. This observation indicates that improving the target–rival margin or tail performance alone does not necessarily have reliable overall classifier improvement. Therefore, the results support the role of the Curator in identifying synthetic features that provide more consistent and globally beneficial training utility.

A.9 Complete Comparison

Table 6: Comparison with state-of-the-art methods on ImageNet-LT dataset.

Method	**Ref.**	**ImageNet-LT**			
		Many	**Med.**	**Few**	**All**
RIDE(4Exp) Wang et al. (2021)	ICLR'21	–	–	–	56.80
ACE(3Exp) Cai et al. (2021)	ICCV'21	–	–	–	56.60
GML Suh & Seo (2023)	ICML'23	68.70	55.70	38.60	58.30
BCL Zhu et al. (2022)	CVPR'22	67.90	54.20	36.60	57.10
NCL Li et al. (2022)	CVPR'22	–	–	–	58.40
ResLT Cui et al. (2023)	TPAMI'23	63.60	55.70	38.90	56.10
GLMC Du et al. (2023)	CVPR'23	**70.10**	52.40	30.40	56.30
GLMC + BS Du et al. (2023)	CVPR'23	64.76	55.67	**42.19**	57.21
ProCo Du et al. (2024)	TPAMI'24	–	–	–	58.00
DBM–BCL Son et al. (2025)	AAAI'25	–	–	–	57.40
DBM–GML Son et al. (2025)	AAAI'25	–	–	–	58.60
GLMC + CAFE	Ours	65.94	**57.20**	37.46	57.88
ProCo + CAFE	Ours	68.60	56.18	39.72	**58.72**

Table 7: Comparison with state-of-the-art methods on CIFAR10-LT and CIFAR100-LT under different imbalance factors. Top-1 accuracy of ResNet-32 He et al. (2016) is reported.

Method	**Ref.**	**CIFAR10-LT**			**CIFAR100-LT**		
		10	**50**	**100**	**10**	**50**	**100**
CE	Baseline	86.55	77.89	70.57	56.73	44.70	40.24
CB-Focal Cui et al. (2019)	CVPR'19	87.49	84.36	74.57	57.99	52.59	39.60
WCADS Han (2023)	ICML'23	92.40	–	86.40	65.80	–	52.50
De-confound-TDE Tang et al. (2020)	NeurIPS'20	88.50	83.60	80.60	59.60	50.30	44.10
BALMS Ren et al. (2020)	NeurIPS'20	91.30	84.90	81.50	63.00	50.80	45.50
MiSLAS Zhong et al. (2021)	CVPR'21	89.35	83.85	80.26	62.20	51.89	47.02
PaCo Cui et al. (2021)	ICCV'21	–	–	–	64.20	56.00	52.00
MiSLAS + H2T Li et al. (2024)	AAAI'24	89.66	85.48	81.97	62.33	53.17	48.55
SBCL Hou et al. (2023)	ICCV'23	–	–	–	57.90	48.70	44.90
BCL Zhu et al. (2022)	CVPR'22	91.39	86.70	84.22	64.08	54.86	51.57
ProCo Du et al. (2024)	TPAMI'24	91.90	88.20	85.90	65.50	57.10	52.80
WCADS + LDMLR Han et al. (2024)	CVPRW'24	92.58	–	86.29	62.67	–	49.76
ResLT Cui et al. (2023)	TPAMI'23	89.70	85.17	82.40	62.01	52.71	48.21
Balanced Softmax + NBN Bao et al. (2025)	TIP'25	–	87.00	84.10	–	58.90	53.70
DBM-BCL Son et al. (2025)	AAAI'25	91.69	87.16	84.60	65.25	55.98	51.66
ECL Xu et al. (2024)	TMM'24	91.80	88.90	86.50	67.30	59.90	56.30
BalPoE Aimar et al. (2023)	CVPR'23	–	–	–	65.10	56.30	52.00
ICL Chen et al. (2025)	IJCAI'25	91.30	88.50	86.40	69.00	59.70	56.30
MBJ + RIDE Liu et al. (2022)	AAAI'22	89.40	87.30	82.20	62.10	59.80	49.90
DiffuLT + RIDE Shao et al. (2024)	NeurIPS'24	90.90	87.30	85.30	64.20	56.90	52.40
BCE3S Fan et al. (2026)	AAAI'26	93.85	89.41	85.25	71.93	60.40	54.70
DeiT-LT (ViT) Rangwani et al. (2024)	CVPR'24	–	89.70	87.50	–	60.50	55.60
CaUIRL Zhang et al. (2025)	PR'25	92.50	89.10	86.50	72.10	57.40	52.60
LOS Sun et al. (2025)	ICLR'25	–	–	–	69.70	58.80	54.90
CE + CAFE	Ours	88.44	79.71	72.87	58.43	45.89	41.59
BCE3S + CAFE	Ours	**94.65**	**90.88**	**87.76**	**73.10**	**61.72**	**56.50**

Table 8: Three independent runs under different imbalance factors.

Method	**IF=10**			**IF=50**			**IF=100**		
	1	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
CIFAR10-LT									
CE + CAFE	88.41	88.45	88.46	79.72	79.71	79.71	72.86	72.88	72.86
BCE3S + CAFE	94.63	94.67	94.66	90.87	90.87	90.89	87.85	87.72	87.73
CIFAR100-LT									
CE + CAFE	58.42	58.41	58.45	45.90	45.88	45.89	41.59	41.58	41.59
BCE3S + CAFE	73.16	73.15	73.00	61.72	61.74	61.69	56.61	56.43	56.46

Table 9: Comparison with state-of-the-art methods on iNaturalist 2018 dataset. Top-1 accuracy of ResNet-50 He et al. (2016) is reported. "Many", "Med.", "Few", and "All" denote different groups. The best results are highlighted in bold.

Method	**Ref.**	**iNaturalist 2018**			
		Many	**Med.**	**Few**	**All**
ACE (3Exp) Cai et al. (2021)	ICCV'21	–	–	–	72.90
BalPoE Aimar et al. (2023)	CVPR'23	–	–	–	73.50
ResLT Cui et al. (2023)	TPAMI'23	73.00	72.60	73.10	72.90
DSCL Xuan & Zhang (2024)	AAAI'24	74.20	72.90	70.30	72.00
BCL Zhu et al. (2022)	CVPR'22	–	–	–	71.80
PaCo Cui et al. (2021)	ICCV'21	–	–	–	73.20
DBM–BCL Son et al. (2025)	AAAI'25	65.60	71.80	73.80	71.90
DBM–GML Son et al. (2025)	AAAI'25	66.90	71.90	73.60	72.00
RIDE + RL Zhao et al. (2024)	ECCV'24	71.40	70.90	**74.80**	73.60
WCDAS Han (2023)	ICML'23	75.50	72.30	69.80	71.80
RIDE (3Exp) + CMO Park et al. (2022)	CVPR'22	72.80	68.70	72.60	73.10
SBCL Hou et al. (2023)	ICCV'23	73.30	71.90	68.60	70.80
ProCo Du et al. (2024)	TPAMI'24	–	–	–	73.50
BCE3S Fan et al. (2026)	AAAI'26	**77.16**	74.45	72.78	73.99
ProCo + CAFE	Ours	72.41	**74.91**	73.74	**74.19**

For completeness, we provide the full comparisons with existing long-tailed recognition methods in Tables 6, 7, and 9. While the main paper reports representative methods for clarity and space efficiency, the tables in the Appendix include the complete set of baselines considered in our experiments. Table 8 further reports the results of three independent runs on the CIFAR-LT datasets to evaluate the stability of CAFE across different runs. The complete results further confirm the consistent effectiveness and stability of CAFE across different datasets, imbalance factors, and base models.