TAILCON: Mitigating Tail Signal Erosion through Memory Consolidation for Long-Tailed Recognition

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

Abstract

Long-tailed recognition remains challenging not only because rare classes provide limited supervision, but also because their evidence is difficult to accumulate during standard stochastic training. We characterize this failure as *tail signal erosion*: in conflict-dominant regimes, frequent-class updates can dilute sparse rare-class signals before they are sufficiently reinforced. To address this issue, we propose TAILCON, a memory consolidation framework that separates rare-evidence acquisition from parametric integration. During online acquisition, TAILCON stores uncertain or rare representations in detached episodic memory, while a parametric prediction pathway learns global decision structure. To avoid relying only on test-time retrieval, TAILCON introduces FOCUS, a periodic consolidation stage that distills stored memory evidence into the parametric prediction head through a low-capacity alignment surrogate. We provide a local optimization analysis that scopes when this separation is beneficial in sparse-recurrence, conflict-dominant regimes. Experiments on CIFAR-100-LT, CIFAR-10-LT, ImageNet-LT, and iNaturalist 2018 show consistent improvements over representative rebalancing, decoupled, expert-based, and prototype-oriented baselines under the evaluated protocols. Memory-disabled inference further shows that part of the stored evidence is transferred into the parametric prediction head rather than remaining only in the retrieval branch. An anonymized implementation is included in the **Supplementary Material**.

1 Introduction

Long-tailed recognition remains a central challenge in visual learning because real-world data rarely follow a balanced class distribution [1, 2]. A small number of frequent classes provide abundant supervision, while many rare classes appear only sparsely in the training set. Deep networks trained on such data often achieve strong performance on frequent classes but fail to form reliable decision boundaries for rare classes. Beyond biased class priors, this failure also reflects a persistent optimization difficulty: rare-class evidence is observed intermittently, has high variance, and must compete with frequent head-class updates during stochastic training.

Existing long-tailed learning methods address this challenge through re-sampling or re-weighting [3–9], decoupled representation and classifier learning [10–16], expert specialization [17–22], augmentation or contrastive representation learning [23–25], or memory-augmented recognition [26–28]. These directions have led to substantial progress across standard long-tailed benchmarks, but they mainly rebalance losses, calibrate decision boundaries, increase parametric specialization, improve representation learning, or improve retrieval. Our focus is the coupling of detached rare-evidence ac-

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

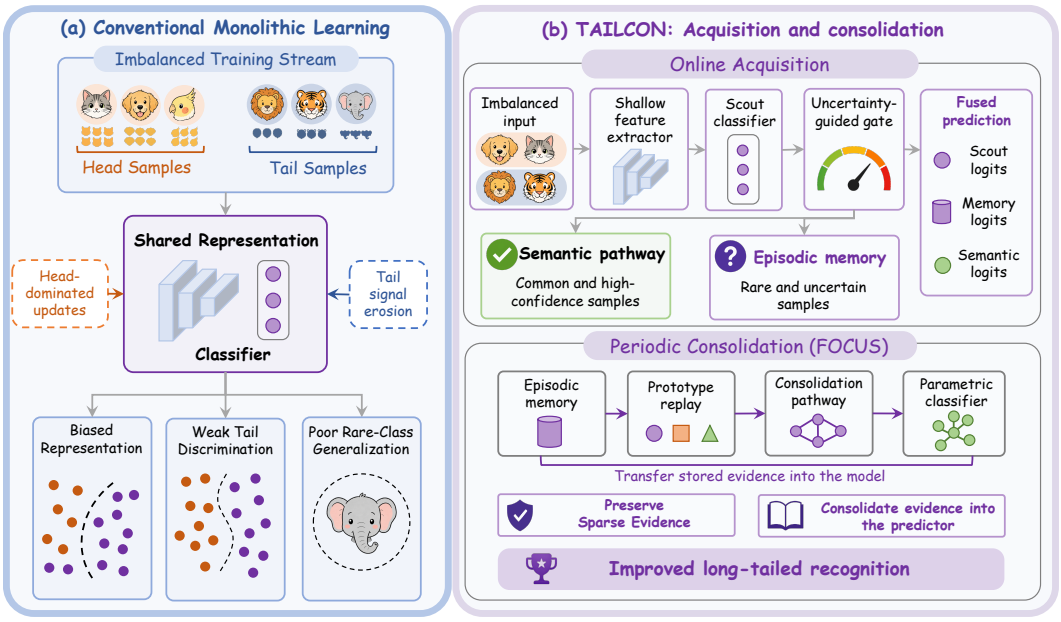

Figure 1: **Motivation and overview of TAILCON.** Conventional monolithic training mixes frequent and rare classes in the same online update stream, which can erode sparse tail evidence. TAILCON separates evidence acquisition from parametric consolidation by preserving uncertain or rare evidence in episodic memory and transferring it into the classifier head through FOCUS.

quisition with periodic memory-derived classifier-head consolidation; Appendix A details its relation to these approaches.

We characterize this failure mode as *tail signal erosion*: in conflict-dominant regimes, frequent-class updates can dilute rare-class signals before they are sufficiently reinforced. Related to stability-plasticity tension in continual and memory-based learning, this view is used only as an optimization-level analogy and motivates a practical design principle: rare evidence should be captured when it appears and integrated through controlled consolidation.

To address this, we propose TAILCON, a memory consolidation framework for long-tailed recognition. The core idea is to separate rare-evidence acquisition from parametric integration. During online acquisition, TAILCON uses a lightweight confidence gate to route uncertain samples and writes uncertain or rare representations into episodic memory. This memory provides a fast, local mechanism for preserving sparse evidence without forcing rare samples to immediately reshape the shared semantic pathway. In parallel, a parametric prediction pathway learns global decision structure from the training stream under confidence-controlled updates. During inference, predictions combine local memory evidence with global evidence from the parametric prediction pathway.

A key component of TAILCON is FOCUS, a periodic consolidation stage designed to avoid relying only on test-time retrieval. The episodic memory can preserve local evidence for rare and ambiguous samples, but a pure retrieval mechanism does not guarantee that this information becomes part of the parametric model. FOCUS addresses this limitation by replaying memory prototypes and distilling stored memory evidence into the classifier head through a low-capacity alignment surrogate during periodic training phases. This transfers part of the stored evidence into the parametric prediction component, improving memory-disabled inference without claiming that the full semantic representation is rewritten. Thus, the novelty of TAILCON lies in coupling uncertainty-driven memory acquisition with periodic parametric consolidation rather than using memory as retrieval alone.

Our analysis provides a local optimization account of why this separation can be useful. Under sparse recurrence and head-tail conflict, rare-class updates may be weakened by the accumulated effect of frequent-class updates between two rare observations. The memory branch mitigates this issue by preserving rare evidence through non-parametric writes, while the consolidation stage reintroduces

this evidence under a more balanced and controlled objective. This analysis is intentionally scoped to conflict-dominant regimes. It does not assume that sample imbalance always implies gradient dominance, nor does it claim that long-tailed training universally behaves like continual learning. We use the analysis to guide the design of TAILCON and validate the mechanism empirically through gradient diagnostics, ablations, and stress tests under weakened assumptions.

We evaluate TAILCON on CIFAR-100-LT, CIFAR-10-LT, ImageNet-LT, and iNaturalist 2018. Results show consistent improvements over representative long-tailed recognition baselines under the evaluated protocols. Ablations show that episodic memory, confidence routing, and FOCUS play complementary roles, while memory-disabled inference verifies that the gains are not explained by retrieval alone. Additional diagnostics show reduced harmful head-tail interactions with modest computational overhead.

In summary, our contributions are as follows:

❶ **New perspective.** We identify *tail signal erosion* as a practical optimization failure mode, emphasizing that rare-class evidence is not only scarce but also difficult to preserve under conflict-dominant stochastic training.

❷ **Framework design.** We propose TAILCON, which separates rare-evidence acquisition from parametric integration by combining detached episodic memory with FOCUS consolidation and confidence-based routing for stability and budget control.

❸ **Empirical evidence.** We evaluate TAILCON across synthetic and natural long-tailed benchmarks, and show through ablations, mechanism diagnostics, and memory-disabled inference that its gains go beyond test-time retrieval and reflect effective classifier-head consolidation.

2 Related Work

Long-tailed recognition. Long-tailed recognition is commonly addressed by rebalancing the effective training distribution, calibrating the classifier, or improving representation learning. Resampling and re-weighting methods adjust the exposure or loss contribution of rare classes [3–9], while decoupled training and logit-calibration methods separate representation learning from classifier calibration or correct label-prior bias [10–16]. Recent work further improves long-tailed recognition through augmentation, contrastive learning, logit adjustment, or distribution-aware regularization [23, 29, 30, 24, 25]. These methods mainly reduce class-prior bias or improve decision boundaries. In contrast, TAILCON focuses on preserving sparse rare-class evidence during training and consolidating it into the parametric predictor.

Specialization, memory, and optimization conflict. Multi-expert and branch-based methods increase parametric specialization for imbalanced data [17–22, 31]. Memory- and prototype-based recognition methods preserve or reconstruct representative evidence for rare or ambiguous samples [26–28, 32, 33]. TAILCON couples detached memory acquisition with periodic classifier-head consolidation from memory-derived soft targets; Appendix A compares this mechanism with closely related optimization, memory, and distillation methods. Related studies in federated learning address data imbalance through feature-space alignment [34] and mitigate representational interference through attention masking [35]. Our motivation is also related to gradient conflict and continual learning, where replay, regularization, gradient projection, or heterogeneity-aware scheduling preserves sparse evidence or reduces interference [36–42]. However, long-tailed recognition repeatedly observes a fixed imbalanced dataset rather than a sequence of tasks with explicit boundaries; we therefore use this connection only as an optimization-level analogy. Additional discussion is provided in Appendix A.

3 A Local View of Tail Signal Erosion

This section gives a local optimization view of why sparse tail evidence can be difficult to preserve. We do not claim a universal failure mode; instead, we characterize a conflict-dominant regime where frequent-class updates can weaken rare-class evidence before reinforcement, motivating the separation between evidence acquisition and parametric consolidation in TAILCON.

3.1 Setup and Diagnostic Quantities

For the local analysis, let $\mathcal{D} = \mathcal{H} \cup \mathcal{T}$ denote the head-tail diagnostic subset of the training data. Medium-frequency classes are excluded, and p_H, p_T are conditioned on this subset. For a parametric classifier $f(\cdot;\theta)$, the diagnostic objective is

$$\mathcal{L}_{\mathcal{D}}(\theta) = p_H \mathcal{L}_H(\theta) + p_T \mathcal{L}_T(\theta), \tag{1}$$

where p_H and p_T are the sampling probabilities of head and tail examples, with $p_H \gg p_T$ in the long-tailed regime. Let

$$\mathbf{g}_H = \nabla \mathcal{L}_H(\theta), \qquad \mathbf{g}_T = \nabla \mathcal{L}_T(\theta). \tag{2}$$

The expected update direction is

$$\bar{\mathbf{g}}(\theta) = p_H \mathbf{g}_H + p_T \mathbf{g}_T. \tag{3}$$

Class imbalance alone does not guarantee head-gradient dominance, since the effective contribution also depends on loss scale, optimization stage, and class difficulty. We therefore use two measurable diagnostics, beginning with the effective dominance ratio

$$\Delta(\theta) = \frac{p_H \|\mathbf{g}_H\|}{p_T \|\mathbf{g}_T\| + \epsilon}, \tag{4}$$

where $\epsilon > 0$ is a small numerical constant. The second is the head-tail gradient alignment

$$\Gamma(\theta) = \frac{\langle \mathbf{g}_H, \mathbf{g}_T \rangle}{\|\mathbf{g}_H\| \|\mathbf{g}_T\|}. \tag{5}$$

We call a regime conflict-dominant when $\Delta(\theta)$ is large and $\Gamma(\theta)$ is negative. These quantities are diagnostics rather than hard-thresholded assumptions.

3.2 Local Tail Drift

Consider one expected stochastic update

$$\theta^+ = \theta - \eta \bar{\mathbf{g}}(\theta), \tag{6}$$

where η is the learning rate. A first-order expansion of the tail loss gives

$$\mathcal{L}_T(\theta^+) - \mathcal{L}_T(\theta) = -\eta p_T \|\mathbf{g}_T\|^2 - \eta p_H \langle \mathbf{g}_T, \mathbf{g}_H \rangle + \mathcal{O}(\eta^2). \tag{7}$$

The first term is useful tail descent, while the second is the head-tail interaction. When $\langle \mathbf{g}_T, \mathbf{g}_H \rangle < 0$ and its magnitude offsets useful descent, the expected update locally moves against the tail objective.

Proposition 3.1 (Diagnostic condition for local tail drift)**.** *Assume $\mathcal{L}_T$ is locally smooth around θ. Under the first-order approximation in Eq. 7, if*

$$\langle \mathbf{g}_T, \mathbf{g}_H \rangle < 0 \quad \textit{and} \quad p_H \, |\langle \mathbf{g}_T, \mathbf{g}_H \rangle| > p_T \|\mathbf{g}_T\|^2, \tag{8}$$

then the expected update increases the tail loss under the first-order approximation.

The complete derivation, including the smoothness remainder, is in Appendix B. Proposition 3.1 is local: it identifies a measurable condition where frequent-class updates can oppose tail improvement, not a universal increase of tail loss.

3.3 Sparse Recurrence and Erosion

Long-tailed training also creates sparse recurrence: a tail sample may appear at step t, followed by many head-dominated updates before related tail evidence appears again. Let $\mathbf{q}_t$ be the tail gradient at step t, and let $\mathbf{h}_{t+i}$ denote the intervening head-dominated gradients. A local approximation gives

$$\theta_{t+k} \approx \theta_t - \eta \mathbf{q}_t - \eta \sum_{i=1}^{k} \mathbf{h}_{t+i}. \tag{9}$$

We measure the retained tail descent direction by

$$R_t = \frac{\langle -\mathbf{q}_t, \theta_{t+k} - \theta_t \rangle}{\eta \|\mathbf{q}_t\|^2} \approx 1 + \frac{\left\langle \mathbf{q}_t, \sum_{i=1}^{k} \mathbf{h}_{t+i} \right\rangle}{\|\mathbf{q}_t\|^2}. \tag{10}$$

A negative projection from intervening head updates reduces R_t; we call this attenuation before reinforcement *tail signal erosion*. Details are in Appendix B.

3.4 Implications for TAILCON

The analysis suggests two design requirements: tail evidence should be preserved when it appears, instead of relying only on potentially head-dominated parametric updates, and the preserved evidence should later improve the parametric predictor rather than remain a retrieval cache.

TAILCON follows these requirements through memory consolidation. During online acquisition, uncertain or rare samples are written into a non-parametric episodic memory through local prototype updates. This write operation does not backpropagate a gradient through the semantic pathway. If ϕ denotes the parameters of the semantic pathway and $\mathcal{U}_{\mathcal{M}}$ denotes the memory write operation, then

$$\nabla_{\phi}\mathcal{U}_{\mathcal{M}}(\mathbf{x}) = 0. \tag{11}$$

This provides functional separation during acquisition. It does not imply global geometric orthogonality, since the shallow feature extractor remains shared and the stored prototypes can be affected by representation drift over training. The claim is narrower: sparse evidence can be stored without forcing potentially conflicting online gradients through the semantic pathway. In practice, prototype staleness is mitigated by continual memory updates, strength-weighted merging, and utility-based replacement, but we do not claim that the memory is invariant to feature drift.

The second requirement is addressed by FOCUS. It periodically replays memory prototypes and distills their evidence into the classifier head through a controlled consolidation objective. This converts memory from a pure test-time retrieval aid into a teacher for improving the parametric predictor. In this way, TAILCON directly instantiates the two implications of the local analysis: fast preservation of sparse evidence through detached non-parametric writes, followed by slower integration into the prediction head through periodic consolidation. The full algorithmic details of memory writing, reading, and FOCUS consolidation are introduced in Section 4.

3.5 Scope of the Analysis

The analysis is local and diagnostic, most relevant under sparse tail observations and weak or negative head-tail alignment. It does not imply universal gradient dominance, equivalence to continual learning, or global orthogonality. These scope conditions motivate the diagnostics in Section 5.

4 TAILCON: Memory Consolidation for Long-Tailed Recognition

TAILCON combines a parametric prediction pathway, detached episodic memory, and FOCUS classifier-head consolidation. Here, *evidence* denotes class-discriminative information carried by features or prototypes and expressed through their induced class logits. *Online acquisition* is ordinary mini-batch training: eligible samples still contribute to the online objective, while a detached feature copy is stored in memory. Operational details are in Appendix D.11.

4.1 Overview

Given an input image $\mathbf{x}$, a shallow feature extractor F_s produces a representation

$$\mathbf{f}_s = F_s(\mathbf{x}). \tag{12}$$

A lightweight scout classifier H_{scout} predicts preliminary logits, while a deeper semantic pathway G_ϕ and classifier head H_{cls} produce semantic logits:

$$\mathbf{z}^{\text{scout}} = H_{\text{scout}}(\mathbf{f}_s), \qquad \mathbf{z}^{\text{sem}} = H_{\text{cls}}\left(G_\phi(\mathbf{f}_s)\right). \tag{13}$$

In parallel, an episodic memory $\mathcal{M}$ stores class-specific prototypes and returns memory logits $\mathbf{z}^{\text{mem}}$ through prototype retrieval. The final prediction combines scout, memory, and semantic evidence:

$$\mathbf{z}^{\text{final}} = \mathbf{z}^{\text{scout}} + \lambda_{\text{mem}}\frac{\mathbf{z}^{\text{mem}}}{\tau_{\text{mem}}} + \lambda_{\text{sem}}\tilde{g}(\mathbf{x})\frac{\mathbf{z}^{\text{sem}}}{\tau_{\text{sem}}}, \tag{14}$$

where $\tilde{g}(\mathbf{x}) = \text{stopgrad}(g(\mathbf{x}))$ during training and $\tilde{g}(\mathbf{x}) = g(\mathbf{x})$ at inference. Here, $g(\mathbf{x})$ is a confidence-based gate, τ_{mem} and τ_{sem} are temperature parameters, and λ_{mem} and λ_{sem} balance the memory and semantic pathways. Detaching the gate in the training loss prevents the scout classifier

from manipulating the routing coefficient through the classification objective. The activation rate is controlled separately by threshold feedback.

$$\mathcal{L}_{\text{online}} = \text{CE}\left(\mathbf{z}^{\text{final}}, y\right), \tag{15}$$

The target activation budget $\rho^\star$ is enforced by the adaptive threshold feedback in Eq. 17, rather than by backpropagating the classification loss through the routing coefficient.

4.2 Confidence-Based Routing

The confidence gate controls how strongly the semantic pathway contributes to the final prediction. We compute confidence from the scout prediction using the margin between the largest and second-largest softmax probabilities:

$$C(\mathbf{x}) = P_{(1)}(\mathbf{x}) - P_{(2)}(\mathbf{x}), \qquad g(\mathbf{x}) = \text{sigmoid}\left(\kappa(\tau - C(\mathbf{x}))\right). \tag{16}$$

Low confidence gives a larger gate value and increases the contribution of the semantic pathway. The threshold τ is adapted online to maintain the target activation rate:

$$\tau \leftarrow \tau - \zeta\left(\bar{g} - \rho^\star\right), \tag{17}$$

where $\bar{g}$ is the average gate value in the current mini-batch and ζ is the feedback step size. Since increasing τ increases the activation rate under Eq. 16, the negative-feedback sign decreases τ when the observed activation exceeds the target budget.

The gate is not a tail-class detector; it measures scout uncertainty and mainly serves stability and budget control during online acquisition. Tail awareness enters memory acquisition through Eq. 19. We use learning-rate warm-up, soft-to-sharp annealing of κ, and threshold feedback to stabilize early routing; schedules and analyses are in Appendices D and I.

4.3 Episodic Memory Acquisition

The episodic memory stores normalized class-specific prototypes. For class c,

$$\mathcal{P}_c = \{(\mathbf{p}_{c,j}, s_{c,j}, a_{c,j}, u_{c,j})\}_{j=1}^{m_c}, \qquad m_c \le B_c, \tag{18}$$

where m_c is the current occupancy and B_c the capacity. Strength $s_{c,j}$ controls merging and replacement utility, age $a_{c,j}$ penalizes stale entries, and uncertainty $u_{c,j}$ guides replay. Retrieval uses cosine similarity. Initialization and updates are specified in Appendix D.5.

A training sample is eligible for memory writing if it is uncertain or belongs to a rare class:

$$\omega(\mathbf{x}, y) = \mathbb{I}\left[C(\mathbf{x}) < \tau_{\text{write}} \text{ or } N_y < t\right], \tag{19}$$

where N_y is the number of training samples in class y and t is the rare-class cutoff. When $\omega(\mathbf{x}, y) = 1$, we detach and normalize the shallow feature:

$$\bar{\mathbf{f}}_s = \frac{\text{stopgrad}(\mathbf{f}_s)}{\| \text{stopgrad}(\mathbf{f}_s) \|_2}. \tag{20}$$

The detached feature is then written into the memory of class y through a local prototype update. This stop-gradient operation allows uncertain or rare evidence to be stored without forcing the same sample to immediately update the semantic pathway through the memory branch. For memory reading, we compare the query feature with stored prototypes and use the strongest class-level match:

$$z_c^{\text{mem}} = \max_{\mathbf{p} \in \mathcal{P}_c} \left(\frac{\text{stopgrad}(\mathbf{f}_s)}{\| \text{stopgrad}(\mathbf{f}_s) \|_2} \right)^\top \mathbf{p}. \tag{21}$$

If class c has no stored prototype, its memory logit is masked with a fixed low value. The resulting memory logits are fused with scout and semantic logits in Eq. 14. Since the query feature is detached for memory reading, the memory branch provides non-parametric evidence but does not create an additional gradient route through the semantic pathway during online acquisition. For large-scale datasets, retrieval is accelerated efficiently by shortlisting candidate classes using scout logits before prototype search, as detailed in Appendix D.

4.4 FOCUS Consolidation

The episodic memory preserves rare evidence, but memory retrieval alone does not guarantee it becomes part of the parametric predictor. FOCUS addresses this limitation by periodically replaying memory prototypes and distilling stored evidence into the parametric prediction head.

Every K epochs, FOCUS samples prototypes from memory using a replay distribution that upweights uncertain and rare prototypes:

$$q(\mathbf{p}_{c,j}) \propto u_{c,j}^{\beta} \left(\frac{N_{\max}}{N_c} \right)^{\gamma}, \tag{22}$$

where $u_{c,j}$ is the prototype uncertainty score, $N_{\max}$ is the largest class count, and β, γ control uncertainty and rarity. This replay distribution reduces head-class dominance during consolidation.

Since memory prototypes are stored in the shallow feature space, they cannot be directly passed through the deeper semantic pathway. We therefore use a low-capacity alignment module A as a consolidation surrogate that maps replayed prototypes into the input space of the classifier head:

$$\tilde{\mathbf{p}} = A(\mathbf{p}). \tag{23}$$

In the reported implementation, A is instantiated as a linear adapter. The adapter is used only during FOCUS to provide a low-capacity bridge between stored shallow prototypes and the classifier head. Thus, FOCUS should be interpreted as consolidating memory evidence primarily into the parametric prediction head, rather than rewriting the full semantic representation.

We construct teacher logits $\mathbf{z}^T(\mathbf{p})$ from class-level memory evidence by applying the memory read operation to the replayed prototype. When applicable, the replayed prototype itself is excluded from the memory read to avoid a trivial self-match. The resulting teacher distribution reflects class-level memory evidence rather than using the prototype label alone. Let $\mathbf{z}^S(\tilde{\mathbf{p}})$ denote the student logits predicted by the classifier head after the alignment module.

$$\mathcal{L}_{\text{focus}} = \mathbb{E}_{\mathbf{p} \sim q(\mathcal{M})} \left[\mathrm{KL} \left(\mathrm{softmax}(\mathbf{z}^T(\mathbf{p})/\tau_T) \middle\| \mathrm{softmax}(\mathbf{z}^S(\tilde{\mathbf{p}})/\tau_T) \right) \right]. \tag{24}$$

During FOCUS, F_s and G_ϕ are frozen, and only A and H_{cls} are updated. Since A is a training-time surrogate, its persistent inference-time effect is mainly through the classifier head. Attribution controls in Appendix G separate memory-evidence consolidation from extra head updates, label-only replay, and corrupted memory teachers under matched consolidation schedules.

4.5 Training and Inference Protocol

During training, TAILCON alternates online acquisition with periodic FOCUS consolidation. Online acquisition updates the scout classifier, semantic pathway, and classifier head through Eq. 15, while memory is updated through detached local writes. FOCUS updates only the alignment module and classifier head through Eq. 24. A compact algorithm is provided in Appendix C.

By default, TAILCON uses both the parametric prediction pathway and episodic memory at inference. The memory is an internal component built only from the training stream, not an external database. No validation or test samples are written into memory or used to update prototypes. To verify gains are not caused only by test-time retrieval, we evaluate a memory-disabled protocol ($\lambda_{\text{mem}} = 0$) in Section 5.2. The memory-disabled predictor still retains substantial improvements after FOCUS, showing that memory evidence is partially transferred into the classifier head rather than remaining only as a retrieval cache at inference time. The alignment module serves as a training-time surrogate for mapping shallow memory prototypes to the classifier-head input space during consolidation.

5 Experiments

Datasets and metrics. We evaluate whether TAILCON improves long-tailed recognition beyond retrieval-only memory usage on CIFAR-100-LT [9], CIFAR-10-LT [9], ImageNet-LT [26] and iNaturalist 2018 [2]. CIFAR-LT datasets use exponential imbalance, while ImageNet-LT and iNaturalist 2018 are natural long-tailed benchmarks. We report Top-1 accuracy and Many/Medium/Few-shot accuracy on CIFAR-100-LT following standard protocols [26, 10]; dataset statistics are in Appendix E.

Table 1: Detailed performance breakdown on CIFAR-100-LT. All numbers are Top-1 accuracy in percent. Best results are in **bold**, and second-best results are underlined.

Method	Type	IR=100				IR=50				IR=10			
		Many	Medium	Few	All	Many	Medium	Few	All	Many	Medium	Few	All
CE-DRW* [3]	*Re-bal.*	63.4	41.2	15.7	41.4	60.6	39.0	22.9	45.0	62.5	48.6	–	58.2
LDAM-DRW* [3]	*Re-bal.*	61.2	39.5	14.4	42.0	64.9	45.1	28.7	46.6	70.3	57.0	–	58.7
Balanced Softmax* [4]	*Re-bal.*	60.8	44.6	20.4	46.1	66.0	47.2	30.1	50.9	68.0	56.0	–	61.0
LADE* [5]	*Re-bal.*	60.2	43.9	19.6	45.6	65.5	46.3	29.7	50.1	69.0	57.8	–	61.6
MiSLAS* [11]	*Re-bal.*	62.1	45.3	21.1	46.8	66.8	48.4	31.2	51.5	70.0	58.5	–	62.5
BCL* [6]	*Re-bal.*	63.1	42.9	23.9	44.2	61.6	43.1	34.3	49.1	62.2	51.8	–	58.9
LSC* [13]	*Decoup.*	66.0	50.0	27.0	51.8	69.0	55.0	36.0	56.5	73.6	63.8	–	65.0
LOS† [12]	*Decoup.*	68.6	53.8	30.2	54.9	69.1	55.8	38.4	58.8	74.0	64.3	–	69.7
IP-TSL* [30]	*Decoup.*	62.4	59.7	31.9	52.4	–	–	–	55.9	–	–	–	62.6
RIDE* [17]	*Expert*	64.7	46.5	22.3	48.0	67.5	50.8	27.9	51.7	66.4	49.4	–	61.8
SADE* [18]	*Expert*	63.4	49.2	25.1	49.8	67.2	53.5	31.8	53.8	72.9	62.5	–	63.6
BalPoE* [44]	*Expert*	65.3	51.1	28.0	52.0	69.2	55.9	34.5	56.3	73.0	63.2	–	64.8
PRL* [19]	*Expert*	68.7	55.3	31.2	52.8	67.5	53.5	39.0	56.6	74.1	64.5	–	65.6
ICL* [20]	*Expert*	–	–	–	57.6	–	–	–	61.3	–	–	–	69.3
ProCo† [45]	*Proto.*	69.0	52.7	34.1	52.8	69.1	55.2	37.0	57.1	73.9	64.0	–	65.5
ConCutMix† [23]	*Aug.*	67.0	52.0	29.2	53.2	69.0	55.6	38.0	57.4	73.0	63.0	–	64.5
FeatRecon† [46]	*Recon.*	68.2	53.3	34.0	52.5	68.8	54.3	37.2	57.0	73.4	63.5	–	65.3
TAILCON	*Memory consol.*	**76.1**±0.3	**60.1**±0.4	**35.4**±0.5	**58.5**±0.4	**74.8**±0.4	**57.7**±0.3	**47.2**±0.5	**62.8**±0.3	**74.5**±0.2	**66.7**±0.4	–	**72.1**±0.3

Table 2: Overall Top-1 accuracy on CIFAR-10-LT, ImageNet-LT, and iNaturalist 2018.

Method	CIFAR-10-LT	ImageNet-LT	iNat 2018
LDAM-DRW* [3]	77.0	48.8	64.6
RIDE* [17]	78.1	56.3	71.8
SADE* [18]	78.8	58.8	72.7
ConCutMix† [23]	81.7	58.5	72.1
FeatRecon† [46]	85.2	56.8	72.9
PRL* [19]	80.5	60.8	75.1
BalPoE* [44]	85.2	58.5	75.0
IP-TSL* [30]	76.4	51.7	74.0
ProCo† [45]	85.9	60.2	73.5
NCL++* [47]	87.2	59.6	75.2
ICL* [20]	87.9	60.2	75.9
TAILCON	**88.4**±0.3	**63.2**±0.4	**76.9**±0.3

Table 3: Component ablation on CIFAR-100-LT with IR= 100.

Variant	Many	Medium	Few	All
Baseline	61.2±0.4	39.5±0.5	14.4±0.4	42.0±0.3
Memory only	65.1±0.5	45.3±0.4	25.8±0.6	46.5±0.4
Semantic only	69.3±0.3	48.2±0.5	19.5±0.5	48.8±0.4
Dual path w/o FOCUS	74.9±0.4	56.5±0.3	31.3±0.5	55.9±0.4
Full w/o gate	74.5±0.3	58.6±0.4	34.4±0.4	57.2±0.3
TAILCON	**76.1**±0.3	**60.1**±0.4	**35.4**±0.5	**58.5**±0.4

Table 4: Memory-disabled inference on CIFAR-100-LT with IR= 100.

Training setting	Many	Medium	Few	All
Without FOCUS	71.8±0.4	51.0±0.5	24.6±0.6	50.7±0.4
With FOCUS	**73.9**±0.3	**55.4**±0.4	**28.7**±0.5	**53.3**±0.3

Implementation and inference. We use ResNet-32 for CIFAR and ResNet-50 [43] for ImageNet-LT and iNaturalist 2018, trained with SGD; full recipes are in Appendix D. We use a fixed default hyperparameter configuration and report sensitivity in Appendix K. All TAILCON results and ablations are averaged over five runs. In comparison tables, * denotes officially reported results and † denotes protocol-aligned reproductions. The memory is constructed only from training samples; validation and test images are never written into memory or used to update prototypes.

Because Tables 1–2 mix officially reported and protocol-aligned reproduced results, they should be interpreted as broad benchmark comparisons. Controlled attribution is provided by the component, memory-disabled, and mechanism analyses in Sections 5.2–5.3 and by the appendix controls.

5.1 Main Results

CIFAR-100-LT. Table 1 reports CIFAR-100-LT results under IR= 100, IR= 50, and IR= 10. Under the reported protocol, TAILCON obtains the strongest overall accuracy among the listed methods, with gains distributed across Many-, Medium-, and Few-shot groups when breakdowns are available. Appendix F details the backbone, augmentation, schedule, and source for each entry, while the following ablations isolate memory acquisition, FOCUS, and confidence routing.

Other benchmarks. Table 2 reports overall Top-1 accuracy on CIFAR-10-LT, ImageNet-LT, and iNaturalist 2018. TAILCON obtains competitive or best performance among the compared methods under the reported protocols, including recent expert-based methods such as NCL++ and ICL.

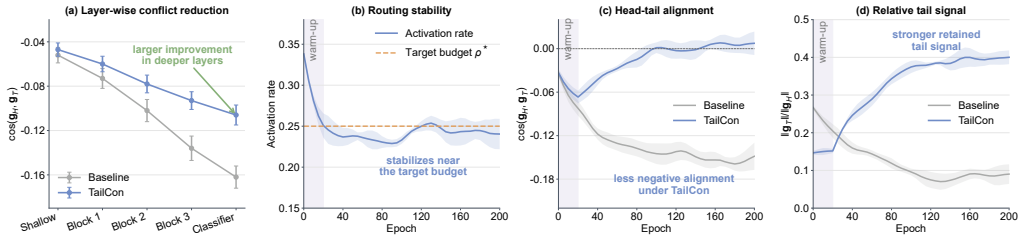

Figure 2: **Mechanism analysis of TAILCON. (a)** Layer-wise gradient alignment shows stronger conflict reduction in deeper semantic layers and the classifier. **(b)** The adaptive gate keeps activation close to the target budget after warm-up. **(c)** TAILCON reduces negative head-tail alignment during training. **(d)** TAILCON maintains a larger tail-to-head gradient magnitude ratio after warm-up.

Table 5: Inference cost on CIFAR-100-LT using the same backbone and input resolution. The three-branch ensemble is included only as a high-cost reference.

Method	GFLOPs	Latency	Rel. latency	Peak VRAM	Inference mode	Main overhead
Baseline	0.14	4.4 ms	1.00×	1.5 GB	Parametric	–
Three-branch ensemble	0.42	13.2 ms	3.00×	4.5 GB	Parametric ensemble	Multiple full branches
TAILCON	0.19	6.1 ms	1.39×	1.8 GB	Memory + parametric	Prototype retrieval and fusion

5.2 Ablation and Consolidation Effects

Table 3 shows that episodic memory, the semantic pathway, FOCUS, and the confidence gate are complementary. FOCUS improves the dual-path model by 2.6 overall points and 4.1 Few-shot points, while the gate further improves the full model by 1.3 overall points and 1.0 Few-shot points. Table 4 isolates consolidation from test-time retrieval: after setting $\lambda_{\text{mem}} = 0$, the FOCUS-trained predictor still retains a clear improvement. This indicates that stored evidence has been transferred into the classifier head rather than remaining only in the retrieval branch.

5.3 Mechanism Analysis

Figure 2 summarizes mechanism diagnostics. TAILCON reduces head-tail conflict most clearly in deeper semantic layers and the classifier head, keeps activation close to the target budget during online acquisition, and maintains a larger tail-to-head gradient magnitude ratio. These diagnostics instantiate the local quantities in Section 3: less negative head-tail alignment corresponds to weaker destructive interaction, while a larger relative tail-signal ratio indicates that rare evidence remains more visible during training. These results support the intended mechanism without implying global orthogonality or a universal guarantee across all regimes.

Additional mechanism details are in the appendix. Appendix H reports layer-wise gradient diagnostics, with the strongest reduction in deeper semantic blocks and the classifier head. Appendix I analyzes early-routing warm-up, and further compares confidence metrics and activation budgets.

5.4 Cost and Practicality

Table 5 reports inference cost on CIFAR-100-LT under the same input resolution. Compared with the baseline, TAILCON increases GFLOPs from 0.14 to 0.19 and latency from 4.4 ms to 6.1 ms, while remaining far lighter than a three-branch ensemble. CIFAR uses exact cosine search; ImageNet-LT and iNaturalist 2018 use scout-based shortlisting over training-memory prototypes. Appendix K reports shortlist and memory-budget sensitivity.

Appendix J reports noisy-label robustness and altered-dominance stress tests. Appendix K reports large-scale hyperparameter sensitivity. Appendix L includes a limited continual-learning diagnostic, testing only whether the preservation and consolidation mechanism remains useful under a no-revisit protocol. Overall, experiments show that TAILCON improves long-tailed recognition, transfers part of the memory evidence into the memory-disabled predictor through FOCUS, and adds only modest inference overhead across evaluated settings.

6 Conclusion

We introduced TAILCON, a memory consolidation framework for long-tailed recognition. TAILCON addresses tail signal erosion by preserving rare and uncertain evidence in episodic memory and transferring part of this evidence into the parametric prediction head through FOCUS. A local optimization analysis motivates this separation in conflict-dominant regimes, while experiments across synthetic and natural long-tailed benchmarks show strong performance and improved memory-disabled inference. These results suggest that preserving sparse evidence and consolidating it into a parametric predictor is a promising direction for robust long-tailed recognition.

Acknowledgments

The authors gratefully acknowledge the support from the National Natural Science Foundation of China (NSFC) under Grant Nos. 62402472, 12227901, and 62476109. This work was also supported by the Natural Science Foundation of Jiangsu Province (No. BK20240461), the Project of Stable Support for Youth Team in Basic Research Field, CAS (No. YSBR-005), and the Academic Leaders Cultivation Program at USTC. The AI-driven experiments, simulations, and model training were performed on the robotic AI-Scientist platform of the Chinese Academy of Sciences.

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

References

[1] 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*, pages 248–255. Ieee, 2009.

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

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

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

[5] 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*, pages 6626–6636, 2021.

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

[7] 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.

[8] 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*, pages 2980–2988, 2017.

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

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

[11] 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*, pages 16489–16498, 2021.

[12] 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.

[13] Tong Wei, Zhen Mao, Zi-Hao Zhou, Yuanyu Wan, and Min-Ling Zhang. Learning label shift correction for test-agnostic long-tailed recognition. In *Forty-first International Conference on Machine Learning*, 2024.

[14] 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 (ICLR)*, 2021.

[15] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2361–2370, 2021.

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

[17] 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.

[18] Yifan Zhang, Bryan Hooi, Lanqing Hong, and Jiashi Feng. Self-supervised aggregation of diverse experts for test-agnostic long-tailed recognition. *Advances in neural information processing systems*, 35:34077–34090, 2022.

[19] Zhe Zhao, HaiBin Wen, Zikang Wang, Pengkun Wang, Fanfu Wang, Song Lai, Qingfu Zhang, and Yang Wang. Breaking long-tailed learning bottlenecks: A controllable paradigm with hypernetwork-generated diverse experts. *Advances in Neural Information Processing Systems*, 37:7493–7520, 2024.

[20] 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*, pages 828–836, 2025.

[21] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9719–9728, 2020.

[22] Liuyu Xiang, Guiguang Ding, and Jungong Han. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In *European conference on computer vision*, pages 247–263. Springer, 2020.

[23] 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.

[24] Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13896–13905, 2020.

[25] 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*, pages 715–724, 2021.

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

[27] 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, pages 1720–1728, 2022.

[28] 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)*, pages 6949–6959. IEEE, 2022.

[29] Pengxiao Han, Changkun Ye, Jieming Zhou, Jing Zhang, Jie Hong, and Xuesong Li. Latent-based diffusion model for long-tailed recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2639–2648, 2024.

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

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

[32] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

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

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

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

[36] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[37] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

[38] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, Christoph H Lampert, et al. Incremental classifier and representation learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5533–5542, 2024.

[39] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 831–839, 2019.

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

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

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

[43] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[44] Emanuel Sanchez Aimar, Arvi Jonnarth, Michael Felsberg, and Marco Kuhlmann. Balanced product of calibrated experts for long-tailed recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19967–19977, 2023.

[45] 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*, 46(9):5890–5904, 2024.

[46] Lingjie Yi, Jiachen Yao, Weimin Lyu, Haibin Ling, Raphael Douady, and Chao Chen. Geometry of long-tailed representation learning: Rebalancing features for skewed distributions. In *The Thirteenth International Conference on Learning Representations*, 2025.

[47] 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*, pages 6949–6958, 2022.

[48] Cong Cong, Shiyu Xuan, Sidong Liu, Shiliang Zhang, Maurice Pagnucco, and Yang Song. Decoupled optimisation for long-tailed visual recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 1380–1388, 2024.

[49] Ahmet Iscen, André Araujo, Boqing Gong, and Cordelia Schmid. Class-balanced distillation for long-tailed visual recognition. *arXiv preprint arXiv:2104.05279*, 2021.

[50] Yin-Yin He, Jianxin Wu, and Xiu-Shen Wei. Distilling virtual examples for long-tailed recognition. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 235–244. IEEE, 2021.

[51] Sarah Parisot, Pedro M Esperança, Steven McDonagh, Tamas J Madarasz, Yongxin Yang, and Zhenguo Li. Long-tail recognition via compositional knowledge transfer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6929–6938. IEEE, 2022.

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

[53] 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, 2022.

[54] Yan Jin, Mengke Li, Yang Lu, Yiu-ming Cheung, and Hanzi Wang. Long-tailed visual recognition via self-heterogeneous integration with knowledge excavation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23695–23704. IEEE, 2023.

Appendix

TAILCON: Mitigating Tail Signal Erosion through Memory Consolidation for Long-Tailed Recognition

The content of this Appendix is summarized as follows:

- **Section A: Extended Related Work.** This section provides additional discussion of related work and clarifies the positioning of TAILCON relative to long-tailed recognition, expert-based models, memory-based recognition, and optimization-conflict studies.
  - Section A.1: Rebalancing and Decoupled Training.
  - Section A.2: Multi-Expert and Branch-Based Methods.
  - Section A.3: Memory-Based Recognition.
  - Section A.4: Optimization Conflict and Continual Learning.
  - Section A.5: Conceptual Comparison.
  - Section A.6: Logit Adjustment and Effective-Prior Correction.
- **Section B: Additional Theory and Derivations.** This section provides the full derivations and additional discussion for the local optimization view of tail signal erosion.
  - Section B.1: Notation and Setup.
  - Section B.2: Local Tail Drift.
  - Section B.3: Smoothness Remainder.
  - Section B.4: Dominance and Alignment Interpretation.
  - Section B.5: Sparse Recurrence and Tail Signal Erosion.
  - Section B.6: Why Non-Parametric Acquisition Helps.
  - Section B.7: Why Consolidation Is Needed.
  - Section B.8: Relation to Continual Learning.
  - Section B.9: Scope and Non-Claims.
  - Section B.10: Summary.
- **Section C: Algorithm and Training Procedure.** This section provides the compact training algorithm and details the online acquisition, memory read/write, FOCUS consolidation, and inference protocols.
  - Section C.1: Training Overview.
  - Section C.2: Memory Read and Write Operations.
  - Section C.3: FOCUS Consolidation Procedure.
  - Section C.4: Inference Protocols.
  - Section C.5: Hyperparameter and Validation Usage.
  - Section C.6: Discussion of Algorithmic Choices.
- **Section D: Implementation Details.** This section reports architecture, training recipes, confidence-gate details, memory construction, large-scale retrieval, FOCUS implementation, default hyperparameters, and reproducibility notes.
  - Section D.1: Network Architecture.
  - Section D.2: Training Recipes.
  - Section D.3: Confidence Gate and Threshold Feedback.
  - Section D.4: Episodic Memory Budget.
  - Section D.5: Memory Write Rule.
  - Section D.6: Memory Read and Empty-Class Handling.
  - Section D.7: Candidate-Class Shortlisting for Large-Scale Datasets.
  - Section D.8: FOCUS Implementation Details.
  - Section D.9: Default Hyperparameters.
  - Section D.10: Reproducibility Notes.
  - Section D.11: Operational Evidence and Module Roles.
  - Section D.12: Prototype Memory, Instance Storage, and Feature Drift.

- **Section E: Dataset and Evaluation Protocols.** This section describes dataset statistics, group definitions, evaluation metrics, inference protocol, and missing-entry handling.
  - Section E.1: CIFAR-100-LT and CIFAR-10-LT.
  - Section E.2: ImageNet-LT.
  - Section E.3: iNaturalist 2018.
  - Section E.4: Many-, Medium-, and Few-Shot Splits.
  - Section E.5: Evaluation Metrics.
  - Section E.6: Inference Protocol.
  - Section E.7: Dataset Statistics.
  - Section E.8: Reporting Missing or Incomparable Results.
  - Section E.9: Reproducibility Notes.
- **Section F: Full Comparison Tables and Source Annotations.** This section provides source-annotated comparison tables and protocol notes for reported and protocol-aligned reproduced baselines.
  - Section F.1: Comparison Protocol.
  - Section F.2: CIFAR-100-LT Source-Annotated Results.
  - Section F.3: Other Benchmarks.
  - Section F.4: Source Status and Protocol Notes.
  - Section F.5: Protocol Alignment.
  - Section F.6: Missing and Incomparable Entries.
  - Section F.7: Reproducibility and Fairness Considerations.
  - Section F.8: Additional Standard Long-Tailed Comparisons.
  - Section F.9: Group-Wise Effects across Datasets.
- **Section G: FOCUS Attribution Controls.** This section isolates whether FOCUS improves the parametric predictor through memory-evidence consolidation rather than through extra classifier-head updates, label-only replay, or corrupted memory-teacher signals.
  - Section G.1: Attribution-Control Protocol.
  - Section G.2: Extra-Update and Teacher-Signal Controls.
  - Section G.3: Default and Memory-Disabled Evaluation.
  - Section G.4: Interpretation of FOCUS Attribution.
  - Section G.5: Memory-Disabled Inference across Datasets.
  - Section G.6: Compatibility with Logit Adjustment.
- **Section H: Additional Mechanism Diagnostics.** This section provides additional gradient diagnostics and connects the empirical behavior of TAILCON to the local optimization view in the main text.
  - Section H.1: Diagnostic Setup.
  - Section H.2: Whole-Model Gradient Diagnostics.
  - Section H.3: Gradient Alignment Dynamics.
  - Section H.4: Tail-to-Head Gradient Magnitude Dynamics.
  - Section H.5: Layer-Wise Gradient Diagnostics.
  - Section H.6: Routing Stability.
  - Section H.7: Per-Group Accuracy Dynamics.
  - Section H.8: Connection to the Main Mechanism Analysis.
  - Section H.9: Interpretation and Limitations.
- **Section I: Gate Warm-Up and Routing Analysis.** This section analyzes early-routing warm-up, confidence metrics, activation budgets, activation dynamics, and the interaction between the gate and memory writing.
  - Section I.1: Role of the Confidence Gate.
  - Section I.2: Early-Routing Instability.
  - Section I.3: Confidence Metrics.

- Section I.4: Activation Budget.
- Section I.5: Activation Dynamics.
- Section I.6: Memory Write Consistency.
- Section I.7: Interaction Between Gate and Memory.
- Section I.8: Summary.

- **Section J: Robustness and Stress Tests under Weakened Assumptions.** This section evaluates TAILCON under noisy labels, altered-dominance settings, calibration diagnostics, and memory-corruption stress tests.
  - Section J.1: Purpose of the Stress Tests.
  - Section J.2: Performance Impact under Weakened Assumptions.
  - Section J.3: Noisy-Head and Clean-Tail Stress Test.
  - Section J.4: Symmetric Label Noise Robustness.
  - Section J.5: Calibration and Reliability.
  - Section J.6: Reliability Diagrams.
  - Section J.7: Robustness to Memory Write Corruption.
  - Section J.8: Interpretation of Robustness Results.
  - Section J.9: Non-Claims.
  - Section J.10: Summary.
- **Section K: Hyperparameter Sensitivity and Scalability.** This section reports default settings, sensitivity to routing and memory hyperparameters, large-scale retrieval behavior, and training-time overhead.
  - Section K.1: Default Configuration.
  - Section K.2: Sensitivity to Activation Budget.
  - Section K.3: Sensitivity to Memory Merge Threshold.
  - Section K.4: Sensitivity to Rare-Class Cutoff.
  - Section K.5: Sensitivity to FOCUS Interval.
  - Section K.6: Sensitivity to Alignment-Module Capacity.
  - Section K.7: Memory Size and Prototype Budget.
  - Section K.8: Candidate-Class Shortlist Size.
  - Section K.9: Large-Scale Hyperparameter Stability.
  - Section K.10: Computational Scalability.
  - Section K.11: Training-Time Overhead.
  - Section K.12: Practical Sensitivity and Untested Replay Choices.
  - Section K.13: Summary.
- **Section L: Continual-Learning Diagnostic.** This section provides a limited diagnostic under a no-revisit protocol, testing whether preservation and consolidation remain useful when samples are not revisited.
  - Section L.1: Protocol.
  - Section L.2: Compared Methods.
  - Section L.3: Results.
  - Section L.4: Complementary Forgetting Metrics.
  - Section L.5: Interpretation.
  - Section L.6: Non-Claims.
  - Section L.7: Summary.
- **Section M: Limitations and Broader Impact.** This section summarizes the intended scope, future extensions, deployment considerations, and broader impact of TAILCON.
  - Section M.1: Limitations.
  - Section M.2: Broader Impact.

A Extended Related Work

This appendix provides additional discussion of related work and clarifies the positioning of TAILCON relative to existing long-tailed recognition methods, expert-based models, memory-based recognition, and optimization-conflict studies. The central distinction of TAILCON is the coupling between uncertainty-driven memory acquisition and periodic parametric consolidation: memory is not used only as a retrieval component, and specialization is not implemented only through multiple homogeneous parametric experts.

A.1 Rebalancing and Decoupled Training

Long-tailed recognition is often formulated as a class-imbalance problem. A common family of methods modifies the effective training distribution or changes the loss contribution of each class. Re-sampling methods increase the exposure of rare classes or reduce the dominance of frequent classes, while re-weighting and logit-adjustment methods assign larger losses or corrected decision logits to underrepresented classes [3–6]. These methods are simple, effective, and widely used as strong baselines. However, they mainly operate at the objective or logit level and do not explicitly model how rare-class evidence is preserved after it appears sparsely in the stochastic training stream.

Another important direction decouples representation learning from classifier learning. Classifier re-training and two-stage calibration methods first learn a representation and then adjust the classifier to reduce class-prior bias [10–13]. These methods show that the representation and classifier can suffer from different imbalance effects, and they often improve tail accuracy by recalibrating the final decision layer. Nevertheless, the representation is still learned through a shared parametric pathway during the main training stage, so sparse tail evidence can be weakly accumulated before classifier calibration.

Recent work further improves long-tailed recognition with stronger augmentation, contrastive objectives, logit adjustment, or distribution-aware regularization [23, 29, 30]. These techniques reduce bias from different perspectives and can be complementary to TAILCON. In contrast, TAILCON targets a different failure mode: rare-class evidence is treated as a sparse signal that should be preserved when it appears and later consolidated into the parametric predictor. This emphasizes the temporal accumulation of rare evidence under stochastic updates, rather than only correcting the final class prior or decision boundary.

Decoupled optimization. Decoupled Optimisation (DO) [48] shares the motivation of reducing harmful coupling between frequent- and rare-class learning. It intervenes in parametric optimization by separating parameter groups and using different training objectives with collaborative optimization. TAILCON additionally changes where sparse observations are retained: a detached feature can remain in bounded episodic memory between its occurrence and a later consolidation phase. Its subsequent teacher distribution is constructed from memory, and FOCUS updates the alignment module and classifier head. Thus, the distinction concerns the preservation mechanism and the source of the consolidation target, not an exclusive claim to decoupled learning.

A.2 Multi-Expert and Branch-Based Methods

Multi-expert learning is a strong paradigm for long-tailed recognition. Methods such as RIDE, SADE, NCL, and PRL introduce multiple parametric experts to improve specialization across class groups or data regions [17, 18, 47, 19]. By increasing parametric capacity and encouraging diverse expert behavior, these methods reduce the burden on a single shared model. Recent expert-collaboration methods further study how independent and collaborative branches can improve long-tailed recognition, with ICL as a representative example [20].

TAILCON is related to this line of work in that it also separates learning roles, but the form of specialization is different. Multi-expert methods typically train several parametric branches, all updated through stochastic gradient descent. Tail samples are still incorporated through parametric optimization, although the optimization space is enlarged by additional experts. In contrast, TAILCON uses heterogeneous specialization: the semantic pathway is parametric and learns global decision structure, while the episodic memory is non-parametric and stores uncertain or rare evidence through

detached local writes. This allows sparse evidence to be preserved without forcing every rare sample to immediately reshape the shared semantic pathway.

The consolidation mechanism further distinguishes TAILCON from standard expert ensembles. In many multi-expert methods, the final prediction is obtained by combining expert outputs. In TAILCON, the memory branch is not only an additional predictor. Through FOCUS, memory evidence is periodically distilled into the classifier head through the alignment surrogate, making TAILCON a memory consolidation framework rather than a conventional multi-branch ensemble.

A.3 Memory-Based Recognition

Memory-based long-tailed recognition has several direct precedents. OLTR [26] combines a direct feature with an associated memory feature through dynamic meta-embedding. Memory-Based Jitter (MBJ) [27] collects historical feature or model variations to improve the within-class diversity available to tail classes. Retrieval Augmented Classification (RAC) [28] combines a base encoder with a parallel retrieval branch over pre-encoded image and text memory. These methods establish that memory and retrieval can supply useful information beyond a single parametric prediction pathway.

A limitation of pure retrieval is that stored evidence may remain external to the parametric predictor. If memory is mainly accessed at inference time, predictions can depend strongly on memory size, memory quality, and retrieval efficiency. This dependence may be acceptable for small datasets but can become more challenging as the number of classes and stored prototypes grows.

TAILCON addresses this limitation by treating memory as both an evidence store and a consolidation teacher. During online training, the episodic memory preserves uncertain or rare evidence through detached local writes. During FOCUS, memory prototypes are replayed and their evidence is distilled into the classifier head through the alignment surrogate. This design allows TAILCON to benefit from memory at inference while also improving the memory-disabled predictor. The memory-disabled inference experiment in the main text directly tests whether stored evidence has been transferred into the prediction head rather than remaining only as a retrieval cache.

Distillation and knowledge transfer. Class-Balanced Distillation (CBD) [49] uses parametric teachers to guide class-balanced feature learning in a second training stage. DiVE [50] transfers a flattened teacher distribution interpreted as virtual examples. Compositional Knowledge Transfer (CKT) [51] uses class prototypes and common-class classifier information to construct rare-class classifiers. TAILCON does not introduce distillation or prototype-based transfer in isolation. Its teacher is induced by the episodic memory that preserves eligible training observations, and the teacher is used periodically to update a low-capacity alignment module and the classifier head. The label-only and shuffled-teacher controls in Appendix G isolate the role of the memory-derived soft targets.

A.4 Optimization Conflict and Continual Learning

Gradient conflict has been studied in multi-task learning, continual learning, and imbalanced learning. Projection-based methods reduce interference by modifying gradients within a shared parametric space. Continual learning studies how models preserve previously acquired knowledge when data arrive in changing task sequences. Replay methods store past examples and revisit them during later training, while regularization methods constrain parameter updates to reduce forgetting.

Our work is connected to these ideas at the level of optimization dynamics. In long-tailed recognition, rare classes are not separated into explicit tasks, and the model repeatedly observes a fixed imbalanced dataset across epochs. Therefore, we do not claim that long-tailed recognition is equivalent to continual learning. Instead, we use a local optimization perspective: rare-class observations are sparse, and their influence can be weakened by many intervening frequent-class updates before sufficient reinforcement. This is the tail signal erosion effect studied in the main text.

TAILCON uses this perspective to motivate a practical architecture. The memory branch preserves rare evidence when it appears, and FOCUS consolidates this evidence under a controlled objective. Unlike generic gradient projection, TAILCON does not only modify update directions inside a shared

Table 6: Mechanistic comparison with related optimization, memory, and knowledge-transfer methods. The final column identifies the distinction from detached acquisition followed by periodic memory-derived classifier-head consolidation.

Method	Information mechanism	Relation to TAILCON
DO [48]	Parameter-group decoupling and collaborative optimization	Intervenes within parametric learning without a dedicated episodic-memory teacher.
OLTR [26]	Dynamic meta-embedding with associated memory features	Uses memory-assisted features without periodic memory-to-head consolidation.
MBJ [27]	Historical feature or model jitters collected in memory	Increases tail diversity without a dedicated memory-derived soft-teacher stage.
RAC [28]	Parallel retrieval over pre-encoded image/text memory	Uses a retrieval branch without periodically transferring its memory into the parametric head.
CBD [49]	Class-balanced distillation from parametric teachers	The teacher comes from trained models, not a detached episodic acquisition path.
DiVE [50]	Flattened teacher distributions as virtual examples	Transfers teacher predictions without maintaining the acquisition-to-consolidation memory lifecycle.
CKT [51]	Composition using prototypes and common-class classifiers	Transfers classifier information without online episodic preservation and periodic replay into the head.
TAILCON	Detached prototypes, memory-derived soft targets, and periodic head updates	Couples preservation, retrieval, and classifier-head consolidation.

parametric model. It separates fast evidence preservation from slower parametric consolidation, then reconnects the two through periodic distillation.

A.5 Conceptual Comparison

Table 6 compares the concrete use of preserved information or teacher knowledge in closely related approaches. The comparison concerns the cited mechanisms; it does not imply that memory, distillation, or decoupled optimization is unique to TAILCON.

The defining mechanism is the complete sequence of detached acquisition, bounded preservation, memory-derived soft evidence, and classifier-head consolidation. The memory supports default inference and also supplies the consolidation target. Memory-disabled inference measures the improvement that remains without direct prototype retrieval.

A.6 Logit Adjustment and Effective-Prior Correction

Logit adjustment [14] corrects the decision rule using class-prior information. Prior2Posterior [52] estimates the model's effective prior from posterior predictions and applies a post-hoc correction. These methods operate on a different aspect of long-tailed learning from retaining sparse observations during training. In particular, empirical class frequency need not equal the effective prior reflected by a learned classifier.

TAILCON changes how eligible observations are preserved and subsequently integrated into the prediction head. Its local analysis therefore uses measured gradient magnitude and alignment in addition to class frequency. Appendix G.6 evaluates compatibility with logit adjustment using a shared correction rule and coefficient. The discussion of Prior2Posterior is conceptual; no protocol-matched empirical comparison with that method is claimed.

B Additional Theory and Derivations

This appendix provides derivations and additional discussion for Section 3. The analysis is intended to explain a local optimization mechanism that can arise in conflict-dominant long-tailed regimes. It is not a universal theory of all imbalanced training dynamics, nor does it imply that long-tailed recognition is equivalent to continual learning. We use the analysis to motivate TAILCON and to define measurable diagnostics for the experiments.

B.1 Notation and Setup

Let $\mathcal{D} = \mathcal{H} \cup \mathcal{T}$ denote the diagnostic subset containing head and tail examples. Medium-frequency examples are excluded from this diagnostic, as specified in Appendix H.1. The probabilities p_H and p_T are conditioned on membership in $\mathcal{D}$ and satisfy $p_H + p_T = 1$. This conditional objective is decomposed as

$$\mathcal{L}_{\mathcal{D}}(\theta) = p_H \mathcal{L}_H(\theta) + p_T \mathcal{L}_T(\theta), \tag{25}$$

where p_H and p_T are the sampling probabilities of head and tail examples. Let $\mathbf{g}_H = \nabla \mathcal{L}_H(\theta)$ and $\mathbf{g}_T = \nabla \mathcal{L}_T(\theta)$. Conditioned on the current parameters θ, the expected update direction is

$$\bar{\mathbf{g}}(\theta) = p_H \mathbf{g}_H + p_T \mathbf{g}_T, \tag{26}$$

and the corresponding expected update with learning rate $\eta > 0$ is

$$\theta^+ = \theta - \eta \bar{\mathbf{g}}(\theta). \tag{27}$$

The main text introduces two diagnostic quantities:

$$\begin{aligned} \Delta(\theta) &= \frac{p_H \|\mathbf{g}_H\|}{p_T \|\mathbf{g}_T\| + \epsilon}, \\ \Gamma(\theta) &= \frac{\langle \mathbf{g}_H, \mathbf{g}_T \rangle}{\|\mathbf{g}_H\| \|\mathbf{g}_T\|}, \end{aligned} \tag{28}$$

where $\epsilon > 0$ is a small numerical constant. $\Delta(\theta)$ measures effective head-tail gradient dominance, while $\Gamma(\theta)$ measures head-tail gradient alignment. Negative alignment indicates direct conflict, whereas weak alignment indicates that frequent-class updates provide little useful descent for the tail objective. These quantities are diagnostics rather than hard-thresholded assumptions. Class imbalance alone does not guarantee this regime, because the relative gradient magnitudes also depend on loss scale, optimization stage, and class difficulty.

B.2 Local Tail Drift

We derive the one-step drift of the tail objective after the expected update in Eq. 27. Let

$$\boldsymbol{\delta} = \theta^+ - \theta = -\eta \left(p_H \mathbf{g}_H + p_T \mathbf{g}_T \right). \tag{29}$$

A first-order Taylor expansion gives

$$\mathcal{L}_T(\theta + \boldsymbol{\delta}) - \mathcal{L}_T(\theta) = \langle \mathbf{g}_T, \boldsymbol{\delta} \rangle + \mathcal{O}\left(\|\boldsymbol{\delta}\|^2 \right). \tag{30}$$

Substituting Eq. 29 yields

$$\begin{aligned} \mathcal{L}_T(\theta^+) - \mathcal{L}_T(\theta) &= -\eta \langle \mathbf{g}_T, p_H \mathbf{g}_H + p_T \mathbf{g}_T \rangle + \mathcal{O}(\eta^2) \\ &= -\eta p_T \|\mathbf{g}_T\|^2 - \eta p_H \langle \mathbf{g}_T, \mathbf{g}_H \rangle + \mathcal{O}(\eta^2). \end{aligned} \tag{31}$$

The first term in Eq. 31 is useful descent for the tail objective, while the second term is the head-tail interaction. When

$$\langle \mathbf{g}_T, \mathbf{g}_H \rangle < 0, \tag{32}$$

the interaction term becomes positive and can increase the tail loss. Under the first-order approximation, the expected update increases the tail loss when

$$p_H \left| \langle \mathbf{g}_T, \mathbf{g}_H \rangle \right| > p_T \|\mathbf{g}_T\|^2. \tag{33}$$

This is the diagnostic condition stated in Proposition 3.1.

B.3 Smoothness Remainder

We next make the local approximation explicit. Assume $\mathcal{L}_T$ is L_T-smooth in a neighborhood of θ. The Taylor remainder satisfies

$$\left| \mathcal{L}_T(\theta + \boldsymbol{\delta}) - \mathcal{L}_T(\theta) - \langle \mathbf{g}_T, \boldsymbol{\delta} \rangle \right| \leq \frac{L_T}{2} \|\boldsymbol{\delta}\|^2. \tag{34}$$

Using Eq. 29,

$$\|\boldsymbol{\delta}\|^2 = \eta^2 \left\|p_H \mathbf{g}_H + p_T \mathbf{g}_T\right\|^2. \tag{35}$$

Therefore,

$$\begin{aligned}\mathcal{L}_T(\theta^+) - \mathcal{L}_T(\theta) \geq & -\eta p_T \|\mathbf{g}_T\|^2 - \eta p_H \langle \mathbf{g}_T, \mathbf{g}_H \rangle \\ & - \frac{L_T \eta^2}{2} \left\|p_H \mathbf{g}_H + p_T \mathbf{g}_T\right\|^2.\end{aligned} \tag{36}$$

When $\langle \mathbf{g}_T, \mathbf{g}_H \rangle < 0$, a sufficient condition for actual positive drift of the tail loss is

$$p_H \left|\langle \mathbf{g}_T, \mathbf{g}_H \rangle\right| > p_T \|\mathbf{g}_T\|^2 + \frac{L_T \eta}{2} \left\|p_H \mathbf{g}_H + p_T \mathbf{g}_T\right\|^2. \tag{37}$$

Eq. 37 is stronger than the first-order condition in Eq. 33. It shows why the analysis is local: when the learning rate is small and the update remains within a locally smooth region, the first-order interaction term is informative; when the learning rate is large or curvature is high, higher-order effects may dominate.

B.4 Dominance and Alignment Interpretation

The first-order condition can be expressed using the diagnostics in Eq. 28. Suppose $\|\mathbf{g}_T\| > 0$ and $\Gamma(\theta) < 0$. Then Eq. 33 becomes

$$p_H \|\mathbf{g}_H\| \|\mathbf{g}_T\| |\Gamma(\theta)| > p_T \|\mathbf{g}_T\|^2. \tag{38}$$

Dividing both sides by $p_T \|\mathbf{g}_T\|^2$ gives

$$\frac{p_H \|\mathbf{g}_H\|}{p_T \|\mathbf{g}_T\|} |\Gamma(\theta)| > 1. \tag{39}$$

Thus, local tail drift depends on both dominance and conflict. A large head-gradient contribution alone is not sufficient if head and tail gradients are well aligned. Negative alignment alone is also not sufficient if the tail-gradient contribution is strong enough. This is why the main text uses both $\Delta(\theta)$ and $\Gamma(\theta)$ as empirical diagnostics.

B.5 Sparse Recurrence and Tail Signal Erosion

The previous derivation studies one update. Long-tailed training also creates sparse recurrence: a tail observation at step t may be followed by many head-dominated updates before related tail evidence appears again. Let $\mathbf{q}_t$ denote the gradient induced by the tail observation, and let $\mathbf{h}_{t+i}$ denote the intervening head-dominated gradients for $i = 1, \ldots, k$. Ignoring higher-order curvature effects along this short trajectory,

$$\theta_{t+k} \approx \theta_t - \eta \mathbf{q}_t - \eta \sum_{i=1}^{k} \mathbf{h}_{t+i}. \tag{40}$$

The total displacement is therefore

$$\theta_{t+k} - \theta_t \approx -\eta \mathbf{q}_t - \eta \sum_{i=1}^{k} \mathbf{h}_{t+i}. \tag{41}$$

We measure how much of the original tail descent direction is retained by

$$R_t = \frac{\langle -\mathbf{q}_t, \theta_{t+k} - \theta_t \rangle}{\eta \|\mathbf{q}_t\|^2}. \tag{42}$$

Substituting Eq. 41 into Eq. 42 gives

$$\begin{aligned}R_t &\approx \frac{\left\langle -\mathbf{q}_t, -\eta \mathbf{q}_t - \eta \sum_{i=1}^{k} \mathbf{h}_{t+i} \right\rangle}{\eta \|\mathbf{q}_t\|^2} \\ &= 1 + \frac{\left\langle \mathbf{q}_t, \sum_{i=1}^{k} \mathbf{h}_{t+i} \right\rangle}{\|\mathbf{q}_t\|^2}.\end{aligned} \tag{43}$$

If the accumulated head-dominated updates have a negative projection on the original tail descent direction, then $R_t < 1$. More generally, if

$$\left\langle \mathbf{q}_t, \sum_{i=1}^{k} \mathbf{h}_{t+i} \right\rangle \leq -\alpha \|\mathbf{q}_t\|^2 \tag{44}$$

for some $\alpha > 0$, then

$$R_t \leq 1 - \alpha. \tag{45}$$

When $\alpha \geq 1$, the retained component along the original tail descent direction becomes non-positive in this local measure. We refer to this attenuation of sparse tail evidence before sufficient reinforcement as *tail signal erosion*.

This recurrence analysis is also local. It does not imply that every tail update is erased, nor does it imply that long-tailed training cannot eventually learn tail classes. It only identifies one mechanism by which sparse tail evidence can be weakened between rare observations.

Relation to other imbalance diagnoses. Tail signal erosion describes a trajectory-level event, not a new source of class imbalance. Representation underfitting concerns whether the final features discriminate tail classes; classifier or posterior bias concerns the final decision rule; gradient imbalance concerns the relative contributions within a current update. The retention score instead asks how much of a useful tail-update component survives intervening updates before reinforcement. These diagnoses can coexist. Under empirical risk minimization and comparable per-sample gradient norms, the sampling frequency creates an expected head-contribution advantage, while the direction of its effect on the tail objective additionally depends on gradient alignment. The local analysis distinguishes these quantities without assuming that the structural effect of frequency disappears.

B.6 Why Non-Parametric Acquisition Helps

The recurrence analysis suggests that rare evidence should be preserved when it appears, rather than relying only on immediate parametric updates. In a purely parametric model, the effect of a rare update can be weakened by subsequent head-dominated updates. TAILCON introduces a non-parametric episodic memory as an additional acquisition path.

Let ϕ denote the parameters of the semantic pathway and let $\mathcal{U}_{\mathcal{M}}$ denote the memory write operation. During memory acquisition, the stored feature is detached from the computational graph. Therefore,

$$\nabla_{\phi} \mathcal{U}_{\mathcal{M}}(\mathbf{x}) = 0. \tag{46}$$

This equation should be interpreted narrowly. It does not imply global geometric orthogonality across the entire model: the shallow feature extractor remains shared, the semantic pathway is still trained by the online objective, and stored prototypes may become stale as the shallow representation changes during training. The claim is functional. The memory write itself does not create an additional gradient route through the semantic pathway, allowing uncertain or rare evidence to be stored locally even when the online parametric update is dominated by frequent-class gradients. The method does not assume that stored prototypes are immutable representations; their quality is maintained empirically through repeated writes, prototype merging, and utility-based replacement.

B.7 Why Consolidation Is Needed

A memory-only strategy can preserve local evidence, but it may leave that evidence external to the parametric predictor. If memory is used only at inference, performance can depend heavily on memory size, retrieval quality, and prototype coverage. This motivates the second design requirement: preserved evidence should later improve the memory-disabled predictor through the classifier head.

FOCUS implements this requirement by replaying memory prototypes and distilling their evidence into the classifier head through a low-capacity alignment surrogate. The feature extractor and semantic representation modules are frozen during this phase, which keeps consolidation controlled and reduces the risk of overfitting sparse prototypes. The replay distribution upweights rare and uncertain prototypes, changing the effective training distribution from the original long-tailed stream to a memory-induced consolidation distribution. In this sense, FOCUS provides a controlled route for preserved tail evidence to influence the parametric predictor without claiming to rewrite the full semantic representation.

This interpretation is consistent with the memory-disabled inference experiment in Section 5.2: after FOCUS, part of the memory evidence remains useful even when the memory branch is removed at inference, suggesting that the classifier head has absorbed part of the stored evidence.

B.8 Relation to Continual Learning

Tail signal erosion is related to interference phenomena studied in continual learning, but the protocols are different. Continual learning usually studies a sequence of tasks or distributions, often with restricted access to earlier data. Long-tailed recognition uses a static imbalanced dataset that is repeatedly observed across epochs. Therefore, our analysis does not claim an equivalence between the two settings.

The connection is only at the level of local optimization dynamics. In both cases, sparse or weakly reinforced evidence can be weakened by subsequent updates. In long-tailed recognition, however, this effect occurs within a fixed imbalanced dataset rather than across an explicit task sequence.

B.9 Scope and Non-Claims

The analysis relies on local first-order approximations and smoothness assumptions. It is most informative when the following conditions approximately hold:

1. Tail observations are sparse relative to head observations.
2. The effective head-gradient contribution is large relative to the tail-gradient contribution.
3. Head and tail gradients are weakly aligned or negatively aligned.

The analysis does not imply that sample imbalance always causes gradient dominance. Tail classes can have large gradients, especially in early training, and head classes can have small gradients after they are well learned. This is why $\Delta(\theta)$ and $\Gamma(\theta)$ are treated as diagnostics rather than fixed assumptions.

The analysis also does not imply that TAILCON guarantees accuracy gains under every distribution, imbalance ratio, or optimization stage. When the above conditions weaken, the expected advantage of separating evidence acquisition from parametric consolidation may shrink. This motivates the weakened-assumption and robustness analyses reported in the main text and appendix.

Finally, Eq. 46 is not a global orthogonality guarantee. It only states that the detached memory write does not backpropagate through the semantic pathway. The full model still contains shared components, and FOCUS intentionally transfers part of the stored memory evidence back into the parametric prediction pathway.

B.10 Summary

The derivations support the mechanism-level interpretation used in the main text. First, under negative head-tail alignment, frequent-class updates can locally oppose tail improvement when the interaction term dominates useful tail descent. Second, sparse recurrence can attenuate the effect of a tail update before it is reinforced. Third, TAILCON mitigates this issue by preserving rare evidence through detached non-parametric writes and later consolidating part of that evidence into the classifier head through FOCUS. These claims are local and diagnostic, and should not be interpreted as a universal theory of all long-tailed training regimes.

C Algorithm and Training Procedure

This appendix provides the training and inference procedures of TAILCON. Training alternates between online acquisition and periodic FOCUS consolidation. During online acquisition, the scout classifier, semantic pathway, and classifier head are updated with the fused prediction objective, while the episodic memory is updated through detached local writes. During FOCUS, memory prototypes are replayed and distilled into the classifier head through the alignment module.

Algorithm 1 Training procedure of TAILCON

Require: Training set $\mathcal{D}$, epochs E, consolidation interval K, target activation rate $\rho^\star$
Require: Components F_s, H_{scout}, G_ϕ, H_{cls}, alignment module A, episodic memory $\mathcal{M}$
Require: Hyperparameters λ_{mem}, λ_{sem}, τ_{mem}, τ_{sem}, τ_T

Initialize $\mathcal{M} \leftarrow \emptyset$
Initialize threshold τ and gate sharpness schedule $\kappa(e)$
for $e = 1$ to E **do**
  Set learning rate and gate sharpness $\kappa \leftarrow \kappa(e)$
  for mini-batch $\mathcal{B} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{|\mathcal{B}|}$ **do**
    Compute shallow features $\mathbf{F}_s \leftarrow F_s(\mathcal{B})$
    Compute scout logits $\mathbf{Z}^{\text{scout}} \leftarrow H_{\text{scout}}(\mathbf{F}_s)$
    Compute confidence margins $C_i = P_{i,(1)} - P_{i,(2)}$ from $\text{softmax}(\mathbf{Z}^{\text{scout}})$
    Compute gates $g_i \leftarrow \text{sigmoid}(\kappa(\tau - C_i))$
    Compute memory logits $\mathbf{Z}^{\text{mem}} \leftarrow \text{ReadMemory}(\mathcal{M}, \text{stopgrad}(\mathbf{F}_s))$
    Compute semantic logits $\mathbf{Z}^{\text{sem}} \leftarrow H_{\text{cls}}(G_\phi(\mathbf{F}_s))$
    Fuse logits for each sample using a detached routing coefficient:

$$\mathbf{z}_i^{\text{final}} = \mathbf{z}_i^{\text{scout}} + \lambda_{\text{mem}} \mathbf{z}_i^{\text{mem}} / \tau_{\text{mem}} + \lambda_{\text{sem}} \, \text{stopgrad}(g_i) \, \mathbf{z}_i^{\text{sem}} / \tau_{\text{sem}}.$$

    Compute online objective:

$$\mathcal{L}_{\text{online}} = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \text{CE}\left(\mathbf{z}_i^{\text{final}}, y_i\right).$$

    Update F_s, H_{scout}, G_ϕ, and H_{cls} using $\mathcal{L}_{\text{online}}$
    Update threshold $\tau \leftarrow \tau - \zeta\left(\frac{1}{|\mathcal{B}|}\sum_i g_i - \rho^\star\right)$
    for all $(\mathbf{x}_i, y_i) \in \mathcal{B}$ satisfying $C_i < \tau_{\text{write}}$ **or** $N_{y_i} < t$ **do**
      $\bar{\mathbf{f}}_{s,i} \leftarrow \text{stopgrad}(\mathbf{f}_{s,i}) \,/\, \| \,\text{stopgrad}(\mathbf{f}_{s,i}) \,\|_2$
      $\text{WriteMemory}(\mathcal{M}, y_i, \bar{\mathbf{f}}_{s,i}, C_i)$
    end for
  end for
  if $e \bmod K = 0$ **and** $\mathcal{M}$ is not empty **then**
    Run FOCUS consolidation using Algorithm 3
  end if
end for
return Trained components and episodic memory $\mathcal{M}$

C.1 Training Overview

Online acquisition denotes the ordinary mini-batch training phase on the fixed training set. A sample satisfying the memory-write condition remains in the online classification objective; a separate stop-gradient feature copy is used for memory maintenance. It is neither a test-time adaptation step nor a requirement to process an unbounded stream.

Let F_s denote the shallow feature extractor, H_{scout} the scout classifier, G_ϕ the semantic pathway, H_{cls} the classifier head, $\mathcal{M}$ the episodic memory, and A the alignment module used in FOCUS. The memory is initialized as empty before training. The adaptive threshold τ and gate sharpness κ follow the schedules described in Appendix D. At each mini-batch, TAILCON computes scout, memory, and semantic logits, fuses them for online optimization, and writes eligible detached shallow features into memory.

C.2 Memory Read and Write Operations

Algorithm 2 summarizes the memory operations used during online acquisition and inference. The read operation returns one memory logit per class using maximum cosine similarity. If a class has no stored prototype, its memory logit is masked with a fixed low value. The write operation uses

Algorithm 2 Memory read and write operations

function READMEMORY($\mathcal{M}, \mathbf{f}$)
  $\bar{\mathbf{f}} \leftarrow \mathbf{f}/\|\mathbf{f}\|_2$
  for all class c **do**
    if $\mathcal{P}_c$ is empty **then**
      $z_c^{\mathrm{mem}} \leftarrow m_{\mathrm{mask}}$
    else
      $z_c^{\mathrm{mem}} \leftarrow \max_{\mathbf{p}\in\mathcal{P}_c} \bar{\mathbf{f}}^\top \mathbf{p}$
    end if
  end for
  return $\mathbf{z}^{\mathrm{mem}}$
end function

function WRITEMEMORY($\mathcal{M}, y, \bar{\mathbf{f}}, C$)
  if $\mathcal{P}_y$ is empty **then**
    Insert $\bar{\mathbf{f}}$ as a new prototype in $\mathcal{P}_y$
    Initialize $s \leftarrow 1$, $a \leftarrow 0$, and $u \leftarrow 1 - C$
  else
    $\mathbf{p}^\star \leftarrow \arg\max_{\mathbf{p}\in\mathcal{P}_y} \bar{\mathbf{f}}^\top \mathbf{p}$
    if $\bar{\mathbf{f}}^\top \mathbf{p}^\star \geq \theta_{\mathrm{merge}}$ **then**
      Merge $\bar{\mathbf{f}}$ into $\mathbf{p}^\star$ using the update rule in Appendix D
      Update strength and uncertainty; reset the age of $\mathbf{p}^\star$ using Appendix D.5
    else
      if $|\mathcal{P}_y| < B_y$ **then**
        Insert $\bar{\mathbf{f}}$ as a new prototype in $\mathcal{P}_y$
      else
        Replace the lowest-utility prototype in $\mathcal{P}_y$
      end if
    end if
  end if
end function

detached features and therefore does not create an additional gradient route through the semantic pathway.

For large-scale datasets, READMEMORY can be accelerated by selecting candidate classes from the scout logits and searching only within the prototypes of those classes. Classes outside the candidate set receive the masked memory logit. This acceleration reduces retrieval cost and does not change the memory-write protocol. No test samples are written into memory.

Every insertion or replacement initializes the stored strength, age, and uncertainty in the same way. After a memory update, the ages of non-updated entries advance on the memory-update clock. These state variables affect merging, replacement, and replay as detailed in Appendix D.5; they do not directly reweight the cosine retrieval logit.

C.3 FOCUS Consolidation Procedure

FOCUS is performed every K epochs to transfer part of the stored memory evidence into the parametric prediction pathway. During this phase, the feature extractor is frozen and only the alignment module A and classifier head H_{cls} are updated. This keeps consolidation low-capacity and prevents sparse prototypes from overfitting the full representation.

Here, $\mathrm{ReadMemory}_{-i}$ denotes the same memory-read operation as Algorithm 2, except that the queried prototype is excluded from its own read set when applicable. This avoids a trivial self-match when constructing teacher logits for a replayed prototype. The teacher logits $\mathbf{z}^T$ are induced by memory evidence, while the student logits $\mathbf{z}^S$ are produced by the alignment module and classifier head. Since the replay distribution upweights uncertain and rare prototypes, FOCUS changes the consolidation distribution from the original long-tailed stream to a memory-induced replay distribution.

Algorithm 3 FOCUS consolidation

Require: Episodic memory $\mathcal{M}$, alignment module A, classifier head H_{cls}, temperature τ_T
Require: Replay distribution $q(\mathbf{p}_{c,j}) \propto u_{c,j}^{\beta}(N_{\max}/N_c)^{\gamma}$
Freeze F_s and G_ϕ; update only A and H_{cls}
for consolidation step $s = 1$ to S_{focus} **do**
  Sample prototype mini-batch $\mathcal{B}_{\mathcal{M}} = \{(\mathbf{p}_i, y_i)\}$ from $q(\mathcal{M})$
  Map prototypes with the alignment module: $\tilde{\mathbf{p}}_i \leftarrow A(\mathbf{p}_i)$
  Compute teacher logits $\mathbf{z}_i^T \leftarrow \text{ReadMemory}_{-i}(\mathcal{M}, \mathbf{p}_i)$
  Compute student logits $\mathbf{z}_i^S \leftarrow H_{\text{cls}}(\tilde{\mathbf{p}}_i)$
  Compute consolidation loss:

$$\mathcal{L}_{\text{focus}} = \frac{1}{|\mathcal{B}_{\mathcal{M}}|} \sum_i \text{KL}\left(\text{softmax}(\mathbf{z}_i^T/\tau_T) \middle\| \text{softmax}(\mathbf{z}_i^S/\tau_T)\right).$$

  Update only A and H_{cls} using $\mathcal{L}_{\text{focus}}$
end for
Unfreeze modules needed for the next online acquisition stage

Algorithm 4 Inference protocols of TAILCON

Require: Test sample $\mathbf{x}$, trained components, training memory $\mathcal{M}$
Require: Protocol flag $\text{disable_memory} \in \{\text{true}, \text{false}\}$
$\mathbf{f}_s \leftarrow F_s(\mathbf{x})$
$\mathbf{z}^{\text{scout}} \leftarrow H_{\text{scout}}(\mathbf{f}_s)$
$\mathbf{P} \leftarrow \text{softmax}(\mathbf{z}^{\text{scout}})$
$C(\mathbf{x}) \leftarrow P_{(1)} - P_{(2)}$
$g(\mathbf{x}) \leftarrow \text{sigmoid}\left(\kappa(\tau - C(\mathbf{x}))\right)$
$\mathbf{z}^{\text{sem}} \leftarrow H_{\text{cls}}(G_\phi(\mathbf{f}_s))$
if disable_memory is **true then**
  $\mathbf{z}^{\text{final}} \leftarrow \mathbf{z}^{\text{scout}} + \lambda_{\text{sem}} g(\mathbf{x}) \mathbf{z}^{\text{sem}}/\tau_{\text{sem}}$
else
  $\mathbf{z}^{\text{mem}} \leftarrow \text{ReadMemory}(\mathcal{M}, \text{stopgrad}(\mathbf{f}_s))$
  $\mathbf{z}^{\text{final}} \leftarrow \mathbf{z}^{\text{scout}} + \lambda_{\text{mem}} \mathbf{z}^{\text{mem}}/\tau_{\text{mem}} + \lambda_{\text{sem}} g(\mathbf{x}) \mathbf{z}^{\text{sem}}/\tau_{\text{sem}}$
end if
return $\arg\max_c z_c^{\text{final}}$

C.4 Inference Protocols

At inference time, TAILCON uses the trained scout classifier, semantic pathway, classifier head, and the episodic memory constructed from the training stream. No test samples are written into memory. The default prediction follows the same fusion rule as Eq. 14. For analysis, we also evaluate a memory-disabled protocol, where the model is trained normally with online acquisition and FOCUS, but the memory branch is removed only at inference by setting $\lambda_{\text{mem}} = 0$.

The memory-disabled protocol is used only for mechanism analysis. Improvements under this protocol indicate that part of the memory evidence has been partially absorbed by the memory-disabled predictor, mainly through the classifier head, rather than remaining only as a test-time retrieval cache.

C.5 Hyperparameter and Validation Usage

We use a fixed default hyperparameter configuration for the main results and report sensitivity analyses in Appendix K. When a validation split is used for model selection or temperature selection, it is separated from the final evaluation split and is never written into episodic memory. The reported Top-1 accuracy is computed only on the official or standard evaluation split for each dataset.

C.6 Discussion of Algorithmic Choices

The procedures above reflect three design choices. First, memory writes use stop-gradient features, so storing uncertain or rare evidence does not introduce an additional online gradient route through the semantic pathway. Second, FOCUS updates only the alignment module and classifier head, keeping consolidation low-capacity and controlled. Third, memory-disabled inference removes the memory branch only at inference, not during training. Therefore, gains under this protocol test whether memory evidence has been partially consolidated into the memory-disabled predictor, mainly through the classifier head.

All hyperparameters, including memory budgets, merge thresholds, consolidation frequency, retrieval settings, and training schedules, are provided in Appendix D.

D Implementation Details

This appendix provides implementation details for TAILCON. The goal is to make the architecture, memory operations, consolidation procedure, and evaluation protocol reproducible. Unless otherwise stated, the same implementation principles are used across all datasets, with dataset-specific backbones and training schedules described below.

D.1 Network Architecture

Backbones and model decomposition. For CIFAR-100-LT and CIFAR-10-LT, we use ResNet-32 as the base architecture. For ImageNet-LT and iNaturalist 2018, we use ResNet-50. The network is decomposed into a shallow feature extractor F_s, a scout classifier H_{scout}, a semantic pathway G_ϕ, and a classifier head H_{cls}.

Given an input image $\mathbf{x}$, the shallow feature extractor produces

$$\mathbf{f}_s = F_s(\mathbf{x}). \tag{47}$$

The scout classifier predicts preliminary logits

$$\mathbf{z}^{\text{scout}} = H_{\text{scout}}(\mathbf{f}_s), \tag{48}$$

which are used to estimate confidence and guide routing. The semantic pathway maps the shallow feature to a higher-level representation:

$$\mathbf{f}_{\text{sem}} = G_\phi(\mathbf{f}_s), \qquad \mathbf{z}^{\text{sem}} = H_{\text{cls}}(\mathbf{f}_{\text{sem}}). \tag{49}$$

The memory branch stores detached and normalized shallow features. This design allows uncertain or rare evidence to be preserved without introducing an additional online gradient route through the semantic pathway during memory acquisition.

Split point. For CIFAR ResNet-32, F_s consists of the stem and the first residual stage, while G_ϕ contains the remaining residual stages before the classifier head. For ImageNet-LT and iNaturalist 2018 with ResNet-50, F_s consists of the convolutional stem and the first residual block group, and G_ϕ contains the remaining residual block groups. The memory stores normalized features produced by F_s, while H_{cls} operates on the output of G_ϕ. The alignment module A maps shallow memory prototypes to the input dimension of H_{cls} during FOCUS.

Alignment module for FOCUS. Memory prototypes are stored in the shallow feature space. During FOCUS, replayed prototypes are mapped into the parametric prediction space through an alignment module:

$$\tilde{\mathbf{p}} = A(\mathbf{p}). \tag{50}$$

In the reported implementation, A is implemented as a lightweight linear alignment module. During FOCUS, the feature extractor and semantic representation modules are frozen. Only the alignment module A and classifier head H_{cls} are updated. This restricts consolidation updates to a low-capacity pathway while allowing memory evidence to influence the classifier head. Therefore, FOCUS should be interpreted as low-capacity prediction-head consolidation rather than full representation retraining.

Table 7: Training recipes used in the experiments.

Setting	CIFAR-100-LT	CIFAR-10-LT	ImageNet-LT	iNaturalist 2018
Backbone	ResNet-32	ResNet-32	ResNet-50	ResNet-50
Optimizer	SGD	SGD	SGD	SGD
Momentum	0.9	0.9	0.9	0.9
Nesterov	Yes	Yes	Yes	Yes
Initial learning rate	0.1	0.1	0.025	0.1
Weight decay	5×10^{-4}	5×10^{-4}	5×10^{-4}	2×10^{-4}
Batch size	128	128	64	256
Training epochs	200	200	180	100
Learning-rate schedule	Step	Step	Cosine	Cosine
Schedule details	160/180, $\gamma = 0.1$	160/180, $\gamma = 0.1$	$T_{\max} = 180$	$T_{\max} = 100$
Warm-up epochs	5	5	5	5
RandAugment	Yes	Yes	Yes	No
Mixup alpha	–	–	0.3	0.2
Hyperparameter selection	Fixed	Fixed	Fixed	Fixed

D.2 Training Recipes

All models are trained with stochastic gradient descent using momentum 0.9. The default control hyperparameters of TAILCON are shared across datasets unless otherwise stated:

$$\rho^\star = 0.25, \qquad \zeta = 0.01, \qquad \kappa : 1 \to 50, \qquad t = 20, \qquad \theta_{\text{merge}} = 0.95, \qquad K = 5. \tag{51}$$

Table 7 summarizes the dataset-specific training setup.

We use the same fixed default control hyperparameters unless otherwise stated. Sensitivity analyses are reported in Appendix K; the official evaluation splits are used only for final reporting.

D.3 Confidence Gate and Threshold Feedback

The scout classifier produces a confidence score based on the margin between the largest and second-largest softmax probabilities:

$$C(\mathbf{x}) = P_{(1)}(\mathbf{x}) - P_{(2)}(\mathbf{x}). \tag{52}$$

The semantic gate is computed as

$$g(\mathbf{x}) = \text{sigmoid}\left(\kappa(\tau - C(\mathbf{x}))\right), \tag{53}$$

where κ controls gate sharpness and τ is an adaptive threshold. A lower confidence score yields a larger gate value and increases the contribution of the semantic pathway.

The threshold is updated by negative feedback control:

$$\tau \leftarrow \tau - \zeta\left(\bar{g} - \rho^\star\right), \tag{54}$$

where $\bar{g}$ is the mini-batch average of $g(\mathbf{x})$. The default target activation rate is $\rho^\star = 0.25$, and the feedback step size is $\zeta = 0.01$. This is a negative-feedback update: when the observed activation rate exceeds the target, the threshold is decreased to reduce future activation.

Gate sharpness schedule. Early routing is unreliable because the scout classifier is not yet well calibrated. We therefore anneal κ from 1 to 50 during early training. This makes routing soft at the beginning and gradually sharpens the gate after the scout classifier becomes more reliable. The warm-up length and gate parameters are summarized in Table 8.

D.4 Episodic Memory Budget

For each class c, the episodic memory stores a bounded set of normalized prototypes:

$$\mathcal{P}_c = \{(\mathbf{p}_{c,j}, s_{c,j}, a_{c,j}, u_{c,j})\}_{j=1}^{m_c}, \qquad m_c \le B_c, \tag{55}$$

Here, m_c is the current occupancy and B_c the capacity. Each entry contains a normalized prototype $\mathbf{p}_{c,j}$, strength $s_{c,j}$, age $a_{c,j}$, and uncertainty $u_{c,j}$. The bank may be partially occupied or empty.

In the reported implementation, we use a fixed per-class cap:

$$B_c = B_{\max}, \tag{56}$$

where $B_{\max}$ is the maximum number of prototypes per class. This keeps memory usage bounded and avoids linear growth with the number of training samples. The default value is reported in Table 8.

D.5 Memory Write Rule

Each new memory entry is initialized as $\mathbf{p} = \bar{\mathbf{f}}_s$, $s = 1$, $a = 0$, and $u = 1 - C(\mathbf{x})$. Strength counts compatible observations incorporated into the entry. Age measures memory-update steps since the entry was last updated. Uncertainty summarizes the scout uncertainty of its incorporated observations.

A sample is eligible for memory writing if it is uncertain or belongs to a rare class:

$$\omega(\mathbf{x}, y) = \mathbb{I}\left[C(\mathbf{x}) < \tau_{\text{write}} \text{ or } N_y < t\right], \tag{57}$$

where N_y is the number of training samples in class y and t is the rare-class cutoff. The default cutoff is $t = 20$.

When $\omega(\mathbf{x}, y) = 1$, the shallow feature is detached and normalized:

$$\bar{\mathbf{f}}_s = \frac{\text{stopgrad}(\mathbf{f}_s)}{\| \text{stopgrad}(\mathbf{f}_s) \|_2}. \tag{58}$$

The stop-gradient operation prevents the memory write itself from introducing an additional gradient route through the semantic pathway.

Prototype insertion and merge. Let $\mathbf{p}_y^\star$ be the most similar prototype in class y:

$$\mathbf{p}_y^\star = \arg\max_{\mathbf{p} \in \mathcal{P}_y} \bar{\mathbf{f}}_s^\top \mathbf{p}. \tag{59}$$

If $\mathcal{P}_y$ is empty, the feature is inserted as the first prototype of class y. Otherwise, if

$$\bar{\mathbf{f}}_s^\top \mathbf{p}_y^\star \geq \theta_{\text{merge}}, \tag{60}$$

we merge the feature into the existing prototype by a strength-weighted normalized update:

$$\mathbf{p}_y^\star \leftarrow \frac{s_y^\star \mathbf{p}_y^\star + \bar{\mathbf{f}}_s}{\left\| s_y^\star \mathbf{p}_y^\star + \bar{\mathbf{f}}_s \right\|_2}, \tag{61}$$

and update the prototype strength:

$$s_y^\star \leftarrow s_y^\star + 1. \tag{62}$$

The prototype uncertainty score is updated as a strength-weighted average:

$$u_y^\star \leftarrow \frac{(s_y^\star - 1)u_y^\star + (1 - C(\mathbf{x}))}{s_y^\star}. \tag{63}$$

The age of the merged prototype is reset to zero, while the ages of non-updated prototypes are increased by one memory-update step.

Prototype replacement. If the most similar prototype is below the merge threshold, the feature is inserted as a new prototype when the class memory is not full. If $|\mathcal{P}_y| = B_y$, we replace the prototype with the lowest utility score:

$$U_{c,j} = \frac{s_{c,j}}{1 + a_{c,j}}. \tag{64}$$

This utility favors prototypes that are both frequently reinforced and recently updated. The prototype with the smallest $U_{c,j}$ is replaced by the new detached feature. The new prototype is initialized with strength 1, age 0, and uncertainty score $1 - C(\mathbf{x})$.

D.6 Memory Read and Empty-Class Handling

For memory reading, the shallow feature is detached and normalized. The memory logit for class c is computed as the strongest cosine match:

$$z_c^{\text{mem}} = \max_{\mathbf{p} \in \mathcal{P}_c} \left(\frac{\text{stopgrad}(\mathbf{f}_s)}{\| \text{stopgrad}(\mathbf{f}_s) \|_2} \right)^\top \mathbf{p}. \tag{65}$$

If $\mathcal{P}_c$ is empty, the memory logit is set to a fixed low value:

$$z_c^{\text{mem}} = m_{\text{mask}}, \qquad \text{if } \mathcal{P}_c = \emptyset. \tag{66}$$

The mask is a logit-space suppression value, not a neutral zero contribution. Under the additive fusion below, it penalizes classes without available memory matches. Its value is reported in Table 8.

The memory logits are fused with scout and semantic logits:

$$\mathbf{z}^{\text{final}} = \mathbf{z}^{\text{scout}} + \lambda_{\text{mem}} \frac{\mathbf{z}^{\text{mem}}}{\tau_{\text{mem}}} + \lambda_{\text{sem}} g(\mathbf{x}) \frac{\mathbf{z}^{\text{sem}}}{\tau_{\text{sem}}}. \tag{67}$$

D.7 Candidate-Class Shortlisting for Large-Scale Datasets

For CIFAR-scale datasets, memory retrieval is implemented with exact cosine search over all stored prototypes. For ImageNet-LT and iNaturalist 2018, we reduce retrieval cost by using the scout classifier to shortlist candidate classes. Specifically, we select the top-k_{cand} classes according to $\mathbf{z}^{\text{scout}}$:

$$\mathcal{C}_{\text{cand}} = \text{TopK}\left(\mathbf{z}^{\text{scout}}, k_{\text{cand}}\right). \tag{68}$$

Memory search is then restricted to prototypes belonging to classes in $\mathcal{C}_{\text{cand}}$. For classes outside $\mathcal{C}_{\text{cand}}$, the memory logit is set to m_{mask}. This operation reduces the retrieval search space and suppresses the memory logits of non-candidate classes in the fused prediction. Thus, shortlist size affects both search cost and the available class evidence; it is not an exactly equivalent implementation of all-class retrieval. The memory-write protocol is unchanged, and no test samples are written into memory.

The shortlist size k_{cand} is reported in Table 8. Sensitivity to k_{cand} is analyzed in Appendix K.

D.8 FOCUS Implementation Details

FOCUS is performed every K epochs, with default $K = 5$. It samples prototypes from memory using a rarity- and uncertainty-aware replay distribution:

$$q(\mathbf{p}_{c,j}) \propto (u_{c,j} + \epsilon_u)^\beta \left(\frac{N_{\max}}{N_c} \right)^\gamma, \tag{69}$$

where $u_{c,j}$ is the prototype uncertainty score, $N_{\max}$ is the largest class count, N_c is the number of training samples in class c, and β, γ control uncertainty and rarity prioritization. The small constant ϵ_u prevents zero-probability sampling.

Prototype uncertainty. The uncertainty score $u_{c,j}$ is maintained during memory writing. When a prototype is inserted, its uncertainty is initialized as $1 - C(\mathbf{x})$. When a sample is merged into an existing prototype, the uncertainty score is updated by Eq. 63. Thus, prototypes that consistently receive low-confidence samples obtain larger uncertainty scores and are sampled more often during FOCUS.

Teacher and student logits. For a replayed prototype $\mathbf{p}$, teacher logits $\mathbf{z}^T(\mathbf{p})$ are obtained by applying the memory read operation to $\mathbf{p}$. To avoid a trivial self-match, the replayed prototype itself is excluded from the read set when applicable. If excluding the replayed prototype leaves its class with no remaining prototype, we set the teacher logit of its ground-truth class to the maximum non-masked memory logit before temperature scaling. Other class logits are still obtained from the remaining memory prototypes. This fallback avoids a degenerate teacher distribution caused by masking the only prototype of the replayed class.

Table 8: Default hyperparameters of TAILCON.

Hyperparameter	CIFAR-100-LT	CIFAR-10-LT	ImageNet-LT	iNaturalist 2018
Target activation rate $\rho^\star$	0.25	0.25	0.25	0.25
Threshold feedback rate ζ	0.01	0.01	0.01	0.01
Gate sharpness κ	$1 \to 50$	$1 \to 50$	$1 \to 50$	$1 \to 50$
Rare-class cutoff t	20	20	20	20
Memory write threshold τ_{write}	0.3	0.3	0.3	0.3
Memory merge threshold θ_{merge}	0.95	0.95	0.95	0.95
Maximum prototypes per class $B_{\max}$	10	10	10	10
Memory mask value m_{mask}	-10^4	-10^4	-10^4	-10^4
Semantic weight λ_{sem}	1.0	1.0	1.0	1.0
Memory weight λ_{mem}	0.5	0.5	0.5	0.5
Semantic temperature τ_{sem}	1.0	1.0	1.0	1.0
Memory temperature τ_{mem}	2.0	2.0	2.0	2.0
Teacher temperature τ_T	3.0	3.0	3.0	3.0
FOCUS interval K	5	5	5	5
FOCUS steps	200	200	200	200
Replay uncertainty exponent β	1.5	1.5	1.5	1.5
Replay rarity exponent γ	0.5	0.5	0.5	0.5
Replay smoothing constant ϵ_u	10^{-6}	10^{-6}	10^{-6}	10^{-6}
Candidate classes k_{cand}	All	All	50	50

Student logits are obtained by applying the alignment module and classifier head:

$$\tilde{\mathbf{p}} = A(\mathbf{p}), \qquad \mathbf{z}^S(\tilde{\mathbf{p}}) = H_{\text{cls}}(\tilde{\mathbf{p}}). \tag{70}$$

The consolidation objective is

$$\mathcal{L}_{\text{focus}} = \mathbb{E}_{\mathbf{p}\sim q(\mathcal{M})} \left[\text{KL}\left(\text{softmax}(\mathbf{z}^T(\mathbf{p})/\tau_T) \middle\| \text{softmax}(\mathbf{z}^S(\tilde{\mathbf{p}})/\tau_T) \right) \right]. \tag{71}$$

During FOCUS, the feature extractor and semantic representation modules are frozen. Only the alignment module A and classifier head H_{cls} are updated. This restricts consolidation updates to a low-capacity pathway while allowing memory evidence to influence the classifier head. Therefore, FOCUS should be interpreted as low-capacity prediction-head consolidation rather than full representation retraining.

D.9 Default Hyperparameters

Table 8 summarizes the default hyperparameters used by TAILCON in the reported experiments.

D.10 Reproducibility Notes

Dataset splits, training schedules, and baseline source annotations are reported in Appendix F. All reported TAILCON results are averaged over five runs. For the memory module, we emphasize three implementation details:

1. Memory writes are always detached from the computational graph.
2. The episodic memory is constructed only from the training stream; validation and test samples are never written into memory or used to update prototypes.
3. In the memory-disabled protocol, the model is trained normally and only the memory branch is removed at inference by setting $\lambda_{\text{mem}} = 0$.

These details distinguish TAILCON from methods that rely only on test-time retrieval or external databases.

D.11 Operational Evidence and Module Roles

In TAILCON, evidence denotes class-discriminative information carried by an observed feature or stored prototype and operationalized through the class-logit distribution it induces. It is not an additional latent variable or a causal quantity. Table 9 identifies its concrete form at each stage.

Table 9: Operational forms of evidence and the separate role of confidence.

Stage	Concrete object
Observation	Detached, normalized feature copy of an eligible training example.
Preservation	Class-specific prototype summarizing compatible observations.
Memory read	Class logits induced by maximum cosine similarity to stored prototypes.
FOCUS	Soft teacher distribution obtained from memory-derived class logits.
Confidence	Scout margin controlling routing and uncertainty-based write eligibility; not evidence itself.

Table 10: Memory design on CIFAR-100-LT with IR$= 100$, under equal caps on stored feature vectors. Accuracy and same-class consistency are percentages, reported as mean $\pm$ standard deviation over five runs.

Memory design	Many	Medium	Few	All	Same-class consistency
Instance queue, equal storage	75.5 ± 0.2	59.1 ± 0.3	34.5 ± 0.4	57.8 ± 0.3	91.3 ± 0.3
Frozen prototypes	74.6 ± 0.3	57.4 ± 0.3	32.1 ± 0.4	56.3 ± 0.3	83.6 ± 0.4
Online prototypes without strength/age	75.3 ± 0.3	59.0 ± 0.3	34.2 ± 0.4	57.6 ± 0.3	90.4 ± 0.3
Full prototype memory	$\mathbf{76.1} \pm 0.3$	$\mathbf{60.1} \pm 0.4$	$\mathbf{35.4} \pm 0.5$	$\mathbf{58.5} \pm 0.4$	$\mathbf{94.2} \pm 0.2$

Routing, writing, and replay are different decisions. The gate $g(\mathbf{x})$ weights the semantic prediction according to scout uncertainty. The write condition $\omega(\mathbf{x}, y)$ decides whether a detached copy enters memory and additionally uses the training count N_y. The replay distribution prioritizes already stored entries using class rarity and prototype uncertainty. Low confidence can increase semantic contribution and permit writing, but a rare example is also write-eligible through $N_y < t$. Consequently, memory acquisition is not an exclusive partition that removes rare examples from ordinary learning.

Gradient flow. During online learning, F_s, H_{scout}, G_ϕ, and H_{cls} receive gradients through the online objective. The routing coefficient and memory read/write features are detached as specified in the main method. Detached memory logits can still change the fused prediction and hence the classification residual; detachment removes the extra differentiable memory route, not the influence of memory evidence on the loss. During FOCUS, F_s and G_ϕ are frozen and only A and H_{cls} are updated. The alignment module is not used during inference. Its persistent effect is carried through the updated classifier head.

State variables. Strength $s_{c,j}$ is incremented after a compatible merge and supplies the weight in Eq. 61. Age $a_{c,j}$ resets when an entry is updated and enters the replacement utility $s_{c,j}/(1 + a_{c,j})$. Uncertainty $u_{c,j}$ is initialized from $1 - C(\mathbf{x})$ and updated by the strength-weighted average in Eq. 63. Neither strength nor age directly modifies cosine retrieval; the replay distribution uses rarity and uncertainty.

D.12 Prototype Memory, Instance Storage, and Feature Drift

A controlled comparison on CIFAR-100-LT with IR$= 100$ tests whether class prototypes offer a useful trade-off against individual feature storage. All variants share the backbone, write eligibility, cap on stored feature vectors, memory weight, FOCUS schedule, replay steps, and optimizer. The instance queue stores individual stop-gradient feature copies with utility-based replacement. Frozen prototypes are formed early and are never refreshed. The online-prototype control removes strength/age-based maintenance, while the full memory uses the update rules in Appendix D.5.

Same-class consistency. Let $\mathcal{V}$ be the fixed set of diagnostic probes, and let $\widehat{F}_s$ be the final shallow extractor. For a probe $(\mathbf{x}, y)$, let $c^\star(\mathbf{x})$ denote the class of the stored vector with highest cosine similarity to its normalized final feature. Same-class consistency is

$$\mathrm{SC} = \frac{100}{|\mathcal{V}|} \sum_{(\mathbf{x},y)\in\mathcal{V}} \mathbb{I}[c^\star(\mathbf{x}) = y], \tag{72}$$

where the search uses the final stored memory. This diagnostic measures label agreement of nearest-memory retrieval under the final representation. It differs from the write-event consistency in Appendix I.6, which evaluates the relation of a feature to the memory structure during acquisition. The two quantities therefore have different evaluation objects and are not interchangeable.

Full prototype memory exceeds the equal-storage instance queue by 0.7 overall and 0.9 Few-shot percentage points. Frozen prototypes lose 2.2 overall points relative to the full memory; online maintenance recovers much of this loss. Adding strength/age-based maintenance to the online-prototype control yields another 0.9 overall and 1.2 Few-shot points. These results support updated prototypes as an effective memory representation in this setting, without establishing universal superiority over instance queues. Detached storage does not make a prototype invariant to representation drift: continual writes, weighted merging, and age-aware replacement mitigate, but do not eliminate, that risk.

E Dataset and Evaluation Protocols

This appendix describes the datasets, imbalance construction, evaluation metrics, group definitions, and inference protocols used in the experiments. We evaluate TAILCON on CIFAR-100-LT, CIFAR-10-LT, ImageNet-LT, and iNaturalist 2018. CIFAR-100-LT and CIFAR-10-LT are synthetically imbalanced benchmarks constructed from balanced CIFAR datasets, while ImageNet-LT and iNaturalist 2018 provide large-scale natural long-tailed recognition settings.

E.1 CIFAR-100-LT and CIFAR-10-LT

CIFAR-100-LT and CIFAR-10-LT are constructed by exponentially reducing the number of training examples per class while keeping the test set balanced. Following standard long-tailed recognition protocols [3, 10], classes are ordered by their original class index, and the target number of training samples in class c is defined as

$$n_c = n_{\max} \cdot \mathrm{IR}^{-\frac{c}{C-1}}, \qquad c = 0, \dots, C-1, \tag{73}$$

where C is the number of classes, $n_{\max}$ is the number of samples in the most frequent class, and IR is the imbalance ratio. In implementation, n_c is rounded to an integer following the standard CIFAR-LT construction used by prior long-tailed recognition work. All group assignments are computed from the resulting integer class counts. The imbalance ratio is

$$\mathrm{IR} = \frac{\max_c n_c}{\min_c n_c}. \tag{74}$$

For CIFAR-100-LT, we evaluate $\mathrm{IR} = 100$, $\mathrm{IR} = 50$, and $\mathrm{IR} = 10$. For CIFAR-10-LT, we report the standard $\mathrm{IR} = 100$ setting in the main text. The test sets remain balanced and are not modified.

E.2 ImageNet-LT

ImageNet-LT is a large-scale long-tailed benchmark derived from ImageNet [26]. It contains 1,000 classes with a naturally imbalanced training distribution. The number of training images per class varies substantially, creating a large gap between frequent and rare categories. We follow the standard ImageNet-LT training, validation, and test splits used in long-tailed recognition.

For ImageNet-LT, the main text reports overall Top-1 accuracy using ResNet-50. Group-wise results are reported in Appendix F when they are available under compatible protocols. We avoid mixing group-wise numbers from incompatible training recipes or evaluation splits.

E.3 iNaturalist 2018

iNaturalist 2018 is a naturally long-tailed fine-grained recognition benchmark. It contains a large number of biological species with a highly imbalanced class distribution. Compared with CIFAR-LT, iNaturalist 2018 is more challenging because rare classes are not only underrepresented but also fine-grained and visually similar to related species.

We follow the standard training and validation protocol used in long-tailed recognition. The main text reports overall Top-1 accuracy using ResNet-50. The cross-dataset analysis in Appendix F.9 additionally uses the count-based Many/Medium/Few definitions below.

E.4 Many-, Medium-, and Few-Shot Splits

For CIFAR-100-LT, we report Many-shot, Medium-shot, Few-shot, and overall accuracy in the main comparison table. The groups are defined by the number of training examples per class. Following common long-tailed recognition practice, we use

$$\begin{aligned} \text{Many} &:= \{c \mid n_c > 100\}, \\ \text{Medium} &:= \{c \mid 20 \le n_c \le 100\}, \\ \text{Few} &:= \{c \mid n_c < 20\}. \end{aligned} \tag{75}$$

For CIFAR-100-LT with different imbalance ratios, the number of classes in each group can change because the class counts are generated by Eq. 73. When a group is not applicable or not reported for a baseline, we use a dash in the table.

For ImageNet-LT, the Many/Medium/Few convention is also commonly used in the literature. However, to keep the main paper compact and to avoid mixing incompatible reporting protocols from different papers, the main text reports overall Top-1 accuracy. Compatible group-wise results are provided in Appendix F when available.

The cross-dataset component comparison in Appendix F.9 applies these same count-based definitions to ImageNet-LT, iNaturalist 2018, and CIFAR-10-LT. At IR$= 100$, CIFAR-10-LT has no class with fewer than 20 training examples. Its Few-shot entry is therefore shown as a dash; the grouping threshold is not changed to manufacture a Few-shot group.

E.5 Evaluation Metrics

We report Top-1 accuracy throughout the paper. For a dataset with test set or validation set $\mathcal{D}_{\text{eval}}$, Top-1 accuracy is computed as

$$\text{Acc} = \frac{1}{|\mathcal{D}_{\text{eval}}|} \sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{eval}}} \mathbb{I}\left[\arg\max_c z_c(\mathbf{x}) = y\right]. \tag{76}$$

For group-wise accuracy, the same metric is computed on the subset of evaluation examples whose labels belong to the corresponding group:

$$\text{Acc}_{\mathcal{G}} = \frac{1}{|\mathcal{D}^{\mathcal{G}}_{\text{eval}}|} \sum_{(\mathbf{x},y)\in\mathcal{D}^{\mathcal{G}}_{\text{eval}}} \mathbb{I}\left[\arg\max_c z_c(\mathbf{x}) = y\right], \tag{77}$$

where $\mathcal{G} \in \{\text{Many}, \text{Medium}, \text{Few}\}$ when group-wise evaluation is used.

E.6 Inference Protocol

The default inference protocol uses the trained scout classifier, semantic pathway, classifier head, and episodic memory constructed only from the training stream. No validation or test samples are written into memory, and no evaluation sample is used to update prototypes. Thus, the memory is an internal model component rather than an external test-time database.

For memory-disabled inference, the model is trained normally with online acquisition and FOCUS consolidation. At inference time, the memory branch is removed by setting $\lambda_{\text{mem}} = 0$. This protocol is used only for mechanism analysis and tests whether memory evidence has been partially absorbed by the memory-disabled predictor, mainly through the classifier head, rather than remaining only as a test-time retrieval cache.

E.7 Dataset Statistics

Table 11 summarizes the datasets and evaluation protocols. CIFAR-LT datasets are constructed by exponential imbalance, while ImageNet-LT and iNaturalist 2018 use standard long-tailed recognition splits.

E.8 Reporting Missing or Incomparable Results

In the comparison tables, a dash indicates that a result is unavailable, not reported by the original paper, or not directly comparable under the same protocol. We do not fill missing group-wise

Table 11: Dataset statistics and evaluation protocols. CIFAR-LT datasets are constructed by exponential imbalance. ImageNet-LT and iNaturalist 2018 use standard long-tailed recognition splits.

Dataset	Classes	Imbalance setting	Backbone	Main metric	Additional group reporting	Split protocol
CIFAR-100-LT	100	IR=100, 50, 10	ResNet-32	Top-1	Many/Medium/Few	Exponential imbalance
CIFAR-10-LT	10	IR=100	ResNet-32	Top-1	Many/Medium	Exponential imbalance
ImageNet-LT	1000	Natural LT	ResNet-50	Top-1	When compatible	Standard split
iNaturalist 2018	8142	Natural LT	ResNet-50	Top-1	Many/Medium/Few	Standard split

values by interpolation, nor do we mix numbers from incompatible settings. Full source annotations, including whether each number is officially reported or protocol-aligned reproduced, are provided in Appendix F.

E.9 Reproducibility Notes

All reported results follow the same evaluation convention: no test-time memory writes, no external database, and Top-1 accuracy computed on the official or standard evaluation split of each benchmark. For reproduced baselines, we follow the training recipes specified in Appendix F. For officially reported baselines, we list the original source, backbone, and protocol whenever available.

F Full Comparison Tables and Source Annotations

This appendix provides source annotations and protocol details for the comparison tables in the main text. The goal is to make clear which numbers are officially reported, which numbers are protocol-aligned reproductions, and which entries are unavailable or not directly comparable. A dash denotes unavailable results, unreported group-wise metrics, or results that are not comparable under the same evaluation protocol. We do not fill missing values by interpolation or by mixing incompatible settings.

F.1 Comparison Protocol

Long-tailed recognition results are sensitive to backbone, augmentation, training schedule, evaluation split, and reporting convention. To make the comparison transparent, we use the following notation. * denotes officially reported results taken from the corresponding paper or official release under a standard long-tailed recognition protocol. † denotes protocol-aligned reproduced results. For reproduced baselines, we follow the official implementation whenever available and align the dataset, backbone, and evaluation convention with the protocol described in Appendix E and Appendix D.

The purpose of this appendix is not to claim that every published baseline was trained with an identical recipe. Instead, it makes explicit which values are reported, which values are reproduced, and which entries are unavailable. When a method does not report a required metric, or when the protocol differs substantially from the one used in our experiments, we leave the corresponding entry as a dash.

We emphasize that officially reported numbers are not always obtained under an identical recipe to ours. They are included because they are standard reference results in the long-tailed recognition literature. Protocol-aligned reproduced results are used when implementation and resource constraints allow closer control over backbone, schedule, augmentation, and evaluation convention.

F.2 CIFAR-100-LT Source-Annotated Results

Table 12 provides the source-annotated version of the CIFAR-100-LT comparison. It expands the main comparison by retaining the full Many/Medium/Few/All breakdown under IR$=100$, IR$=50$, and IR$=10$, while explicitly marking whether each method is officially reported or reproduced. Best results are in **bold**, and second-best results are underlined.

F.3 Other Benchmarks

Table 13 provides the source-annotated overall Top-1 comparison on CIFAR-10-LT, ImageNet-LT, and iNaturalist 2018. CIFAR-10-LT uses IR$=100$. For ImageNet-LT and iNaturalist 2018, we follow

Table 12: Source-annotated performance breakdown on CIFAR-100-LT. All numbers are Top-1 accuracy in percent. * denotes officially reported results and † denotes protocol-aligned reproduced results.

Method	Type	Src.	IR=100				IR=50				IR=10			
			Many	Medium	Few	All	Many	Medium	Few	All	Many	Medium	Few	All
CE-DRW [3]	Re-bal.	*	63.4	41.2	15.7	41.4	60.6	39.0	22.9	45.0	62.5	48.6	–	58.2
LDAM-DRW [3]	Re-bal.	*	61.2	39.5	14.4	42.0	64.9	45.1	28.7	46.6	70.3	57.0	–	58.7
Balanced Softmax [4]	Re-bal.	*	60.8	44.6	20.4	46.1	66.0	47.2	30.1	50.9	68.0	56.0	–	61.0
LADE [5]	Re-bal.	*	60.2	43.9	19.6	45.6	65.5	46.3	29.7	50.1	69.0	57.8	–	61.6
MiSLAS [11]	Re-bal.	*	62.1	45.3	21.1	46.8	66.8	48.4	31.2	51.5	70.0	58.5	–	62.5
BCL [6]	Re-bal.	*	63.1	42.9	23.9	44.2	61.6	43.1	34.3	49.1	62.2	51.8	–	58.9
LSC [13]	Decoup.	*	66.0	50.0	27.0	51.8	69.0	55.0	36.0	56.5	73.6	63.8	–	65.0
LOS [12]	Decoup.	†	68.6	53.8	30.2	54.9	69.1	55.8	38.4	58.8	74.0	64.3	–	69.7
IP-TSL [30]	Decoup.	*	62.4	59.7	31.9	52.4	–	–	–	55.9	–	–	–	62.6
RIDE [17]	Expert	*	64.7	46.5	22.3	48.0	67.5	50.8	27.9	51.7	66.4	49.4	–	61.8
SADE [18]	Expert	*	63.4	49.2	25.1	49.8	67.2	53.5	31.8	53.8	72.9	62.5	–	63.6
BalPoE [44]	Expert	*	65.3	51.1	28.0	52.0	69.2	55.9	34.5	56.3	73.0	63.2	–	64.8
PRL [19]	Expert	*	68.7	55.3	31.2	52.8	67.5	53.5	39.0	56.6	74.1	64.5	–	65.6
ICL [20]	Expert	*	–	–	–	57.6	–	–	–	61.3	–	–	–	69.3
ProCo [45]	Proto.	†	69.0	52.7	34.1	52.8	69.1	55.2	37.0	57.1	73.9	64.0	–	65.5
ConCutMix [23]	Aug.	†	67.0	52.0	29.2	53.2	69.0	55.6	38.0	57.4	73.0	63.0	–	64.5
FeatRecon [46]	Recon.	†	68.2	53.3	34.0	52.5	68.8	54.3	37.2	57.0	73.4	63.5	–	65.3
TAILCON	Memory	Ours	**76.1**±0.3	**60.1**±0.4	**35.4**±0.5	**58.5**±0.4	**74.8**±0.4	**57.7**±0.3	**47.2**±0.5	**62.8**±0.3	**74.5**±0.2	**66.7**±0.4	–	**72.1**±0.3

Table 13: Source-annotated overall Top-1 accuracy comparison on CIFAR-10-LT, ImageNet-LT, and iNaturalist 2018. CIFAR-10-LT uses IR$=100$. * denotes officially reported results and † denotes protocol-aligned reproduced results.

Method	Src.	CIFAR-10-LT	ImageNet-LT	iNaturalist 2018
LDAM-DRW [3]	*	77.0	48.8	64.6
RIDE [17]	*	78.1	56.3	71.8
SADE [18]	*	78.8	58.8	72.7
ConCutMix [23]	†	81.7	58.5	72.1
FeatRecon [46]	†	85.2	56.8	72.9
PRL [19]	*	80.5	60.8	75.1
BalPoE [44]	*	85.2	58.5	75.0
IP-TSL [30]	*	76.4	51.7	74.0
ProCo [45]	†	85.9	60.2	73.5
NCL++ [47]	*	87.2	59.6	75.2
ICL [20]	*	87.9	60.2	75.9
TAILCON	Ours	**88.4**±0.3	**63.2**±0.4	**76.9**±0.3

standard long-tailed recognition splits. We include recent expert-based methods such as NCL++ and ICL to make the comparison with parametric expert specialization explicit.

F.4 Source Status and Protocol Notes

Table 14 summarizes the source status of each comparison method. *Reported* means that the value is taken from the original paper or an official released result under the corresponding dataset protocol. *Reproduced* means that the value is obtained from our reproduction or a protocol-aligned implementation. For reproduced baselines, we use the same dataset split, backbone family, and evaluation metric as TAILCON whenever possible. Deviations are listed in the protocol note.

F.5 Protocol Alignment

Table 15 summarizes the evaluation convention used for each dataset in our experiments. For CIFAR-100-LT and CIFAR-10-LT, the standard backbone is ResNet-32. For ImageNet-LT and iNaturalist 2018, our experiments use ResNet-50. When a reported baseline uses a substantially different backbone or training setup, we do not include that number in the main comparison unless it is commonly accepted as part of the standard long-tailed recognition protocol.

Table 14: Source annotations for comparison methods. Reported numbers are taken from original papers or official released results. Reproduced numbers are obtained under protocol-aligned reproduction.

Method	Citation key	Type	Source status	Protocol note
CE-DRW	[3]	Re-balancing	Reported	Standard CIFAR-LT protocol
LDAM-DRW	[3]	Re-balancing	Reported	Standard CIFAR-LT protocol
Balanced Softmax	[4]	Re-balancing	Reported	Standard CIFAR-LT protocol
LADE	[5]	Re-balancing	Reported	Standard CIFAR-LT protocol
MiSLAS	[11]	Re-balancing	Reported	Standard CIFAR-LT protocol
BCL	[6]	Re-balancing	Reported	Standard CIFAR-LT protocol
LSC	[13]	Decoupled	Reported	Standard CIFAR-LT protocol
LOS	[12]	Decoupled	Reproduced	Protocol-aligned reproduction
IP-TSL	[30]	Decoupled	Reported	Reported overall and available breakdowns
RIDE	[17]	Expert	Reported	Standard long-tailed protocol
SADE	[18]	Expert	Reported	Standard long-tailed protocol
BalPoE	[44]	Expert	Reported	Standard long-tailed protocol
PRL	[19]	Expert	Reported	Standard long-tailed protocol
ICL	[20]	Expert	Reported	Available reported metrics only
NCL++	[47]	Expert	Reported	Available reported metrics only
ProCo	[45]	Prototype	Reproduced	Protocol-aligned reproduction
ConCutMix	[23]	Augmentation	Reproduced	Protocol-aligned reproduction
FeatRecon	[46]	Reconstruction	Reproduced	Protocol-aligned reproduction
TAILCON	Ours	Memory consolidation	Ours	Same protocol as Appendix E

Table 15: Protocol alignment summary for the main comparison.

Dataset	Backbone in our experiments	Main metric	Additional group reporting	Notes
CIFAR-100-LT	ResNet-32	Top-1	Many/Medium/Few	IR=100, 50, 10
CIFAR-10-LT	ResNet-32	Top-1	Many/Medium	IR=100
ImageNet-LT	ResNet-50	Top-1	When compatible	Standard long-tailed split
iNaturalist 2018	ResNet-50	Top-1	Many/Medium/Few	Standard split

F.6 Missing and Incomparable Entries

Missing entries arise for three main reasons. First, some papers report only overall accuracy and do not provide Many/Medium/Few breakdowns. Second, some methods report results only for a subset of imbalance ratios or datasets. Third, some results are not directly comparable because they use different backbones, training recipes, data processing pipelines, or evaluation splits.

For these cases, we use a dash rather than estimating missing values. This avoids overstating the strength or weakness of a baseline. For example, when ICL and NCL++ report only selected overall results, we include those reported overall values under the corresponding imbalance ratios or datasets, but leave unavailable group-wise entries blank.

F.7 Reproducibility and Fairness Considerations

We make the following choices to improve fairness and reproducibility:

1. We report the source status of each baseline as reported, reproduced, or ours.
2. We avoid filling missing results with inferred or incompatible values.
3. We keep comparisons under standard dataset protocols and standard backbone choices whenever possible.
4. For reproduced baselines, we follow the official implementation or a protocol-aligned reproduction.
5. For TAILCON, no test or validation samples are written into memory; the memory is built only from the training stream.

These details are important because long-tailed recognition results can be sensitive to backbone, augmentation, schedule, and evaluation split. The main text reports compact comparisons, while this appendix provides the source and protocol information needed to interpret those comparisons.

Table 16: Protocol-aligned standard-LTR comparisons on CIFAR-100-LT with IR$=100$. † denotes a reproduction. Top-1 accuracy is reported as mean $\pm$ standard deviation over five runs.

Method	Many	Medium	Few	All
ResLT†	69.7 ± 0.2	52.8 ± 0.3	29.5 ± 0.4	52.4 ± 0.2
SHIKE†	73.1 ± 0.1	57.6 ± 0.2	34.2 ± 0.3	56.0 ± 0.2
TAILCON	$\mathbf{76.1} \pm 0.3$	$\mathbf{60.1} \pm 0.4$	$\mathbf{35.4} \pm 0.5$	$\mathbf{58.5} \pm 0.4$

Table 17: Group-wise component comparison across datasets. Top-1 accuracy is reported as mean $\pm$ standard deviation over five runs. CIFAR-10-LT uses IR$=100$; its Few-shot group is empty under the original count-based definition.

Dataset	Method	Many	Medium	Few	All
ImageNet-LT	Protocol-matched baseline	65.0 ± 0.3	53.7 ± 0.4	35.1 ± 0.4	55.4 ± 0.4
	Dual path w/o FOCUS	68.9 ± 0.3	59.8 ± 0.4	43.0 ± 0.4	60.9 ± 0.4
	TAILCON	$\mathbf{70.4} \pm 0.3$	$\mathbf{62.4} \pm 0.4$	$\mathbf{46.5} \pm 0.4$	$\mathbf{63.2} \pm 0.4$
iNaturalist 2018	Protocol-matched baseline	76.3 ± 0.3	72.8 ± 0.3	69.6 ± 0.4	71.9 ± 0.3
	Dual path w/o FOCUS	78.4 ± 0.2	75.6 ± 0.3	74.6 ± 0.4	75.5 ± 0.3
	TAILCON	$\mathbf{79.6} \pm 0.2$	$\mathbf{77.1} \pm 0.3$	$\mathbf{75.9} \pm 0.4$	$\mathbf{76.9} \pm 0.3$
CIFAR-10-LT	Protocol-matched baseline	82.0 ± 0.3	71.5 ± 0.4	–	79.9 ± 0.3
	Dual path w/o FOCUS	87.6 ± 0.3	81.1 ± 0.4	–	86.3 ± 0.3
	TAILCON	$\mathbf{89.6} \pm 0.3$	$\mathbf{83.6} \pm 0.4$	–	$\mathbf{88.4} \pm 0.3$

F.8 Additional Standard Long-Tailed Comparisons

ResLT [53] and SHIKE [54] are reproduced on CIFAR-100-LT with IR$=100$, using released implementations where available and matching the ResNet-32 backbone, imbalance construction, augmentation, and evaluation protocol. These comparisons complement the broader benchmark tables with methods directly targeting standard long-tailed recognition.

Under this protocol, TAILCON improves overall accuracy over ResLT and SHIKE by 6.1 and 2.5 percentage points, respectively. ResLT preserves class-specific capacity through residual parameter-space specialization, and SHIKE integrates self-heterogeneous experts. In comparison, TAILCON retains eligible observations in a non-parametric memory and periodically transfers memory-derived evidence into the classifier head. These reproduced results concern IR$=100$; no missing results for other imbalance ratios are inferred.

F.9 Group-Wise Effects across Datasets

Table 17 compares a protocol-matched monolithic baseline, the dual-path model without FOCUS, and full TAILCON on ImageNet-LT, iNaturalist 2018, and CIFAR-10-LT. Within each dataset, the monolithic baseline uses the same backbone, data augmentation, optimizer, and training schedule as TAILCON, with episodic memory, routing, and FOCUS removed. The dual-path control retains the online architecture and memory but omits consolidation. Default inference is used for each applicable variant.

On ImageNet-LT, the gains over the matched baseline are 5.4, 8.7, and 11.4 percentage points for Many, Medium, and Few, respectively. Relative to the dual-path model without FOCUS, consolidation increases the three group accuracies by 1.5, 2.6, and 3.5 points. On iNaturalist 2018, the corresponding full-system gains over the baseline are 3.3, 4.3, and 6.3 points. On CIFAR-10-LT, Medium improves by 12.1 points, compared with 7.6 points for Many. Thus, the full-system improvements extend beyond frequent classes and are larger for the less frequent groups in these comparisons.

Table 18: FOCUS attribution controls on CIFAR-100-LT with IR$= 100$. All variants start from the same dual-path model and use the same consolidation schedule. The memory-disabled columns evaluate the predictor after removing the memory branch at inference. Values are reported as mean $\pm$ standard deviation over five runs.

Variant	Default inference				Memory-disabled inference			
	Many	Medium	Few	All	Many	Medium	Few	All
Dual path w/o FOCUS	74.9±0.4	56.5±0.3	31.3±0.5	55.9±0.4	71.8±0.4	51.0±0.5	24.6±0.6	50.7±0.4
Equal extra head updates, no memory teacher	75.1±0.3	56.9±0.4	31.9±0.5	56.2±0.3	72.0±0.4	51.4±0.4	25.2±0.5	51.0±0.4
FOCUS with label-only prototype targets	75.4±0.3	57.7±0.3	32.8±0.4	56.9±0.3	72.5±0.4	52.2±0.4	26.1±0.5	51.7±0.3
FOCUS with shuffled memory teacher	74.3±0.5	55.7±0.4	30.4±0.6	55.0±0.5	71.2±0.5	49.9±0.5	23.7±0.6	49.6±0.5
Full TAILCON	76.1±0.3	60.1±0.4	35.4±0.5	58.5±0.4	73.9±0.3	55.4±0.4	28.7±0.5	53.3±0.3

G FOCUS Attribution Controls

This appendix provides additional attribution controls for the periodic consolidation stage. The purpose is to separate memory-evidence consolidation from three alternative explanations: extra classifier-head optimization, label-only prototype replay, and arbitrary or corrupted memory-teacher signals. These controls complement the main memory-disabled inference analysis by testing whether the improvement persists in the parametric prediction head rather than only through test-time memory retrieval.

G.1 Attribution-Control Protocol

All variants start from the same trained dual-path model before consolidation and are evaluated on CIFAR-100-LT with IR$= 100$. Unless otherwise stated, the same consolidation interval, replay budget, optimizer, and number of consolidation updates are used across variants. The default inference protocol uses both the parametric pathway and the episodic memory branch, while the memory-disabled protocol removes the memory branch at inference by setting $\lambda_{\mathrm{mem}} = 0$. Thus, improvements under memory-disabled inference must persist in the parametric prediction head rather than being explained by test-time prototype retrieval.

The controls are diagnostic and are reported as mean $\pm$ standard deviation over five independent runs, matching the reporting protocol used for the main results and core ablations.

G.2 Extra-Update and Teacher-Signal Controls

We compare the full FOCUS objective with three controlled alternatives. The first control, *equal extra head updates*, performs the same number of additional classifier-head update steps but removes the memory teacher. This tests whether the gain is simply caused by extra optimization. The second control, *label-only prototype targets*, replays memory prototypes but replaces the soft memory-evidence teacher with the prototype class label. This is the simpler prototype-label retraining control under the same replay budget and schedule; it does not introduce a different replay sampler. The third control, *shuffled memory teacher*, preserves the consolidation schedule and teacher-logit format but breaks the correspondence between replayed prototypes and their memory-derived teacher distributions. This tests whether arbitrary teacher signals act merely as regularization.

G.3 Default and Memory-Disabled Evaluation

Table 18 reports both default and memory-disabled inference. Default inference measures the complete prediction system, including the memory branch. In contrast, memory-disabled inference evaluates the same trained model after removing memory retrieval at test time. This setting is the stricter attribution test: any remaining improvement must be carried by the parametric prediction head.

The equal-update control produces only a small gain over the unconsolidated dual-path model. This indicates that the benefit of FOCUS cannot be explained by extra classifier-head optimization steps alone. The label-only control improves over the equal-update control, especially on medium- and few-shot classes, but remains clearly below the full method. This suggests that replaying class labels

Table 19: Memory-disabled Top-1 accuracy across datasets, reported as mean ± standard deviation over five runs. Retained gain is the difference between the two memory-disabled means, in percentage points. No uncertainty estimate is inferred for the gain column.

Dataset	No FOCUS, memory off	With FOCUS, memory off	Full TAILCON	Retained gain
CIFAR-100-LT, IR= 100	50.7 ± 0.4	53.3 ± 0.3	58.5 ± 0.4	$+2.6$
CIFAR-10-LT, IR= 100	82.8 ± 0.3	85.1 ± 0.3	88.4 ± 0.3	$+2.3$
ImageNet-LT	57.9 ± 0.4	60.3 ± 0.4	63.2 ± 0.4	$+2.4$
iNaturalist 2018	72.9 ± 0.3	74.8 ± 0.3	76.9 ± 0.3	$+1.9$

from memory is helpful but insufficient to transfer the richer class-level evidence encoded by the memory teacher.

The shuffled-teacher control performs worse than the unconsolidated dual-path model, especially under memory-disabled inference. This result is important because the control preserves the consolidation schedule, teacher-logit format, and additional optimization steps, but destroys the semantic correspondence between prototypes and teacher distributions. Therefore, arbitrary or corrupted teacher signals do not provide useful regularization in this setting.

G.4 Interpretation of FOCUS Attribution

The full TAILCON consistently achieves the best performance under both inference protocols. Under memory-disabled inference, it improves overall accuracy from 50.7% to 53.3%, and few-shot accuracy from 24.6% to 28.7%. Since the memory branch is removed at test time in this protocol, these gains cannot be attributed to direct prototype retrieval. Instead, they indicate that periodic consolidation transfers part of the stored memory evidence into the parametric prediction head.

Together, these controls support the intended attribution of FOCUS. The gain is not explained by simply adding classifier-head updates, replaying prototype labels, or injecting arbitrary teacher logits. Rather, FOCUS is most effective when the replayed prototypes are paired with their corresponding memory-derived teacher distributions, which provide class-level evidence beyond a hard prototype label. This supports the view of TAILCON as a memory consolidation framework rather than a retrieval-only memory-augmented classifier.

G.5 Memory-Disabled Inference across Datasets

Each model is trained with its designated online and consolidation procedure, and the memory branch is removed only at inference by setting $\lambda_{\text{mem}} = 0$. The difference between the two memory-disabled settings measures the improvement associated with FOCUS that remains in the parametric predictor. Full TAILCON is evaluated with memory enabled and is the default system.

FOCUS improves the memory-disabled predictor by 1.9–2.6 percentage points across the four datasets. These improvements cannot arise from direct prototype retrieval at evaluation because the memory branch is disabled. At the same time, memory-enabled TAILCON remains stronger on every dataset. The result therefore supports a hybrid memory-parametric system benefiting from both classifier-head consolidation and retrieval, not a retrieval-independent default predictor. During each consolidation phase, the updated modules are A and H_{cls}; the results do not establish full-representation rewriting or recovery from severe feature collapse.

G.6 Compatibility with Logit Adjustment

The compatibility experiment uses CIFAR-100-LT with IR= 100. The same logit-adjustment rule and coefficient are applied to every applicable variant without TAILCON-specific retuning. Table 20 reports the monolithic baseline, memory-disabled TAILCON, and full TAILCON, each with and without the correction.

Logit adjustment improves the baseline by 5.3 overall percentage points. After TAILCON training, the additional gain is 1.1 points under memory-disabled inference and 0.4 points under default inference, with corresponding Few-shot gains of 2.7 and 1.0 points. The smaller but positive increments indicate complementarity with partial overlap in the errors addressed. The comparison is restricted to the

Table 20: Compatibility with logit adjustment on CIFAR-100-LT with IR$=100$. Top-1 accuracy is reported as mean $\pm$ standard deviation over five runs. The correction rule and coefficient are shared across the applicable variants.

Method	Many	Medium	Few	All
Baseline	61.2 ± 0.4	39.5 ± 0.5	14.4 ± 0.4	42.0 ± 0.3
Baseline + logit adjustment	59.4 ± 0.3	45.8 ± 0.4	22.6 ± 0.4	47.3 ± 0.3
TAILCON, memory disabled	73.9 ± 0.3	55.4 ± 0.4	28.7 ± 0.5	53.3 ± 0.3
TAILCON, memory disabled + logit adjustment	73.4 ± 0.3	56.6 ± 0.4	31.4 ± 0.4	54.4 ± 0.3
Full TAILCON	$\mathbf{76.1} \pm 0.3$	60.1 ± 0.4	35.4 ± 0.5	58.5 ± 0.4
Full TAILCON + logit adjustment	75.8 ± 0.3	$\mathbf{60.7} \pm 0.4$	$\mathbf{36.4} \pm 0.4$	$\mathbf{58.9} \pm 0.3$

stated logit-adjustment experiment; Prior2Posterior is discussed as a related post-hoc approach in Appendix A.6, without a matched numerical comparison.

H Additional Mechanism Diagnostics

This appendix provides additional diagnostics supporting the mechanism analysis in Section 5.3. The purpose is to test whether the empirical behavior of TAILCON is consistent with the local optimization view in Section 3. These diagnostics are not intended as a proof of global orthogonality, nor as a universal characterization of all long-tailed training regimes. Instead, they measure whether TAILCON reduces harmful head-tail interactions, stabilizes confidence-based routing, and preserves stronger tail-gradient signals in the evaluated setting.

H.1 Diagnostic Setup

Unless otherwise stated, diagnostics are computed on CIFAR-100-LT with IR$=100$. We compare the monolithic baseline and TAILCON using the same backbone and training protocol. Let $\mathcal{H}$ and $\mathcal{T}$ denote the many-shot and few-shot class groups used in the main experiments. Medium-shot classes are excluded from the head-tail diagnostic split so that the measured gradients reflect the contrast between frequent and rare classes.

For a diagnostic mini-batch, we compute gradients induced by head and tail subsets separately:

$$\mathbf{g}_H = \nabla_\theta \mathcal{L}_H(\theta), \qquad \mathbf{g}_T = \nabla_\theta \mathcal{L}_T(\theta), \tag{78}$$

where θ denotes the parameter subset being analyzed. For whole-model diagnostics, θ includes the parametric prediction components used by the online objective. For layer-wise diagnostics, θ is restricted to the corresponding module. Memory writes are detached and therefore are not treated as gradient routes.

The head-tail gradient alignment is measured by cosine similarity:

$$\Gamma(\theta) = \frac{\langle \mathbf{g}_H, \mathbf{g}_T \rangle}{\|\mathbf{g}_H\| \|\mathbf{g}_T\|}. \tag{79}$$

Negative values indicate direct conflict between head- and tail-induced update directions, while values closer to zero indicate weaker harmful interaction. The relative tail-gradient magnitude is measured by

$$r_T(\theta) = \frac{\|\mathbf{g}_T\|}{\|\mathbf{g}_H\| + \epsilon}, \tag{80}$$

where $\epsilon > 0$ is a small numerical constant.

For stable estimation, diagnostic values are averaged over fixed diagnostic mini-batches. When training curves are reported, each point is computed by averaging the corresponding diagnostic values at that epoch. Final-window summaries are computed over the last portion of training. Unless otherwise stated, the plotted curves report the mean over five runs.

Table 21: Whole-model gradient diagnostics on CIFAR-100-LT with IR$=100$. Cosine similarity measures head-tail gradient alignment. The magnitude ratio measures the relative strength of tail gradients. Values are averaged over the final training window.

Method	$\cos(\mathbf{g}_H, \mathbf{g}_T)$	$\|\mathbf{g}_T\|/\|\mathbf{g}_H\|$
Baseline	-0.18 ± 0.04	0.08
TAILCON	0.02 ± 0.03	0.45

Table 22: Layer-wise head-tail gradient diagnostics on CIFAR-100-LT with IR$=100$. The cosine values measure head-tail gradient alignment. The ratio measures $\|\mathbf{g}_T\|/\|\mathbf{g}_H\|$.

Module	Baseline cosine	TAILCON cosine	Baseline ratio	TAILCON ratio
Shallow extractor F_s	-0.06	-0.02	0.18	0.24
Mid semantic blocks	-0.15	-0.01	0.10	0.39
Deep semantic blocks	-0.24	0.02	0.06	0.48
Classifier head H_{cls}	-0.32	0.01	0.04	0.52

H.2 Whole-Model Gradient Diagnostics

Table 21 reports final-window head-tail gradient alignment and tail-to-head gradient magnitude ratio. The monolithic baseline shows negative head-tail alignment and a weak relative tail-gradient magnitude. TAILCON moves the alignment closer to zero and increases the relative strength of tail gradients.

These results are consistent with the mechanism targeted by TAILCON. The baseline update is dominated by head-class gradients and shows negative alignment with tail gradients. In contrast, TAILCON reduces the negative interaction and preserves a stronger relative tail signal. This supports the view that memory acquisition and consolidation mitigate tail signal erosion in the measured regime.

H.3 Gradient Alignment Dynamics

Figure 3 provides the full training dynamics of head-tail gradient alignment. The baseline remains negatively aligned for a large portion of training, indicating persistent conflict between head and tail objectives. TAILCON shifts the alignment closer to zero after the early training phase. This trend should not be interpreted as a guarantee of global orthogonality. Rather, it indicates that harmful negative interaction is reduced in the measured long-tailed regime.

H.4 Tail-to-Head Gradient Magnitude Dynamics

Figure 4 shows the training dynamics of the relative tail-gradient magnitude. The baseline maintains a small tail-to-head ratio, indicating that tail updates contribute weakly relative to head updates. In contrast, TAILCON maintains a larger ratio after warm-up, suggesting that the tail signal remains more visible during optimization.

H.5 Layer-Wise Gradient Diagnostics

The main text states that the strongest reduction in head-tail conflict appears in deeper semantic blocks and the classifier head. Table 22 reports the corresponding layer-wise diagnostics.

The layer-wise results should be interpreted cautiously. Early layers are shared by all pathways and therefore may not show strong separation. Deeper semantic blocks and the classifier head are closer to the decision boundary and are more directly affected by consolidation, so they are expected to show clearer changes. The purpose of this diagnostic is to identify where conflict reduction is most visible, not to claim that all layers become separated.

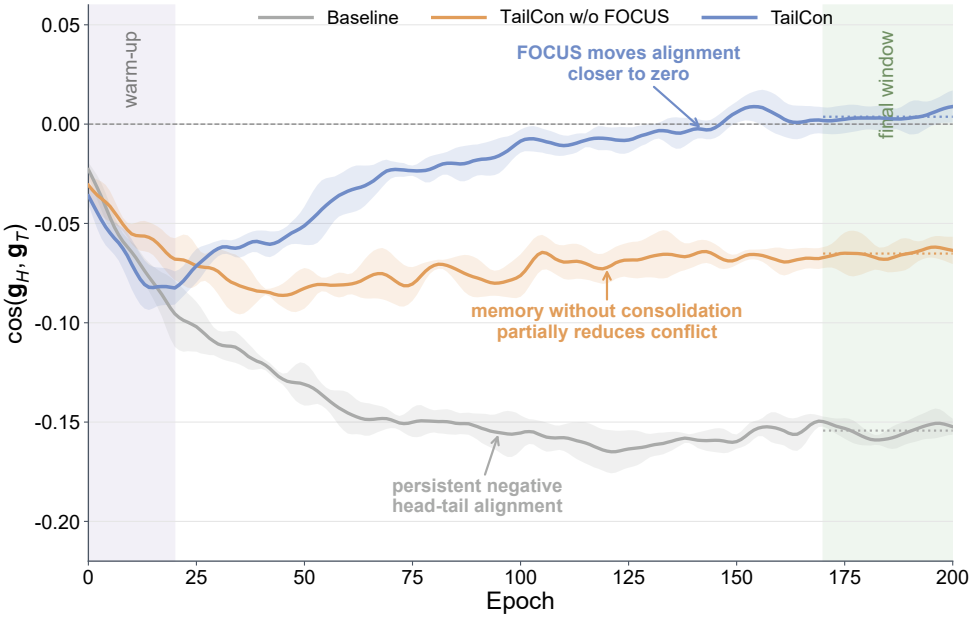

Figure 3: **Expanded head-tail gradient alignment diagnostic.** We report the training dynamics of head-tail gradient alignment on CIFAR-100-LT with IR$=100$. Curves show the mean over five runs, and shaded regions denote standard deviation. Compared with the monolithic baseline, TAILCON reduces negative head-tail alignment after warm-up. The TAILCON w/o FOCUS curve shows that memory acquisition alone partially reduces conflict, while FOCUS further moves the alignment closer to zero during the final training window. This diagnostic is intended as mechanism evidence rather than a global orthogonality claim.

H.6 Routing Stability

The confidence gate is designed to control the contribution of the semantic pathway rather than to identify tail classes directly. For a mini-batch $\mathcal{B}$, we monitor the semantic activation rate

$$\rho_{\mathcal{B}} = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{x} \in \mathcal{B}} g(\mathbf{x}). \tag{81}$$

The threshold feedback rule in Section 4 encourages $\rho_{\mathcal{B}}$ to remain close to the target budget $\rho^\star$ after warm-up. Here, activation denotes the average soft gate weight. It measures semantic contribution and does not by itself measure the fraction of forward passes skipped by conditional execution.

Figure 5 is the expanded routing diagnostic corresponding to the routing panel in the main text. A stable average gate weight near $\rho^\star$ distinguishes controlled semantic contribution from an unconstrained, uniformly strong semantic weighting.

H.7 Per-Group Accuracy Dynamics

To complement the aggregate Many/Medium/Few results, we also monitor group-wise accuracy over training. Let $\mathrm{Acc}_c(e)$ denote the evaluation accuracy of class c at epoch e. For each group $\mathcal{G} \in \{\mathrm{Many}, \mathrm{Medium}, \mathrm{Few}\}$, we compute

$$\mathrm{Acc}_{\mathcal{G}}(e) = \frac{1}{|\mathcal{G}|} \sum_{c \in \mathcal{G}} \mathrm{Acc}_c(e). \tag{82}$$

This diagnostic distinguishes two behaviors. A method that merely trades head performance for tail performance may improve Few-shot accuracy while reducing Many-shot accuracy. In contrast, TAILCON improves Few-shot performance while maintaining strong Many-shot accuracy in the main results, which is consistent with memory consolidation rather than simple tail-biased correction.

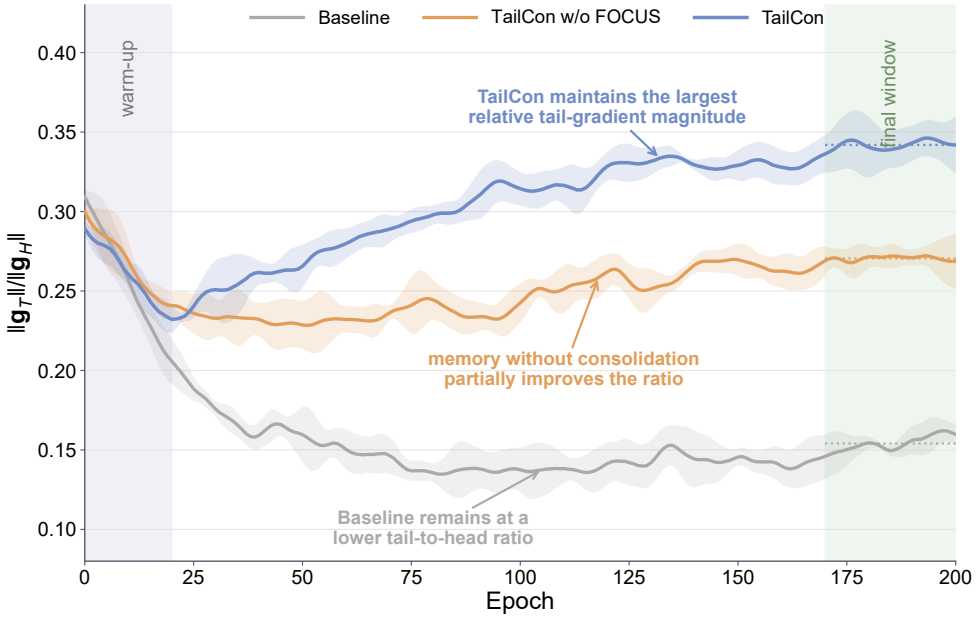

Figure 4: **Expanded tail-to-head gradient magnitude ratio diagnostic.** We report the training dynamics of the tail-to-head gradient magnitude ratio on CIFAR-100-LT with IR$=100$. Curves show the mean over five runs, and shaded regions denote standard deviation. Compared with the monolithic baseline, TAILCON maintains a substantially larger relative tail-gradient magnitude after warm-up. The TAILCON w/o FOCUS curve shows that memory acquisition alone partially improves the ratio, while FOCUS further strengthens retained tail signal during the final training window.

Figure 6 provides a complementary view of group-wise training dynamics, showing whether TAILCON improves the balance across groups rather than merely shifting accuracy from head classes to tail classes.

H.8 Connection to the Main Mechanism Analysis

The mechanism figure in the main text summarizes the primary evidence, while this appendix provides a more explicit description of how those diagnostics are computed and how they should be interpreted. The main text focuses on four observations: reduced head-tail conflict, controlled routing, improved gradient dynamics, and stronger retained tail signal. The expanded plots in Figures 3, 4, and 5, together with the layer-wise results in Table 22, provide the supporting details for those observations.

These diagnostics complement the component ablation and memory-disabled inference results in Section 5.2. The component ablation shows that episodic memory, semantic prediction, and FOCUS consolidation play distinct roles. The memory-disabled inference result shows that memory evidence is not only used as a retrieval cache but is partially transferred into the memory-disabled predictor, mainly through the classifier head.

H.9 Interpretation and Limitations

The diagnostics support the main claim in three ways. First, routing diagnostics show that the confidence gate remains controlled near the target activation budget after warm-up. Second, gradient diagnostics show that TAILCON reduces negative head-tail alignment and preserves a stronger relative tail signal. Third, layer-wise diagnostics show that the strongest conflict reduction appears in deeper semantic blocks and the classifier head, where consolidation most directly affects the decision boundary.

These findings are consistent with the local theory in Section 3. However, they should not be interpreted as proof that TAILCON enforces global gradient orthogonality or guarantees improvement under every distribution. The claim is narrower: in the evaluated long-tailed regime, TAILCON

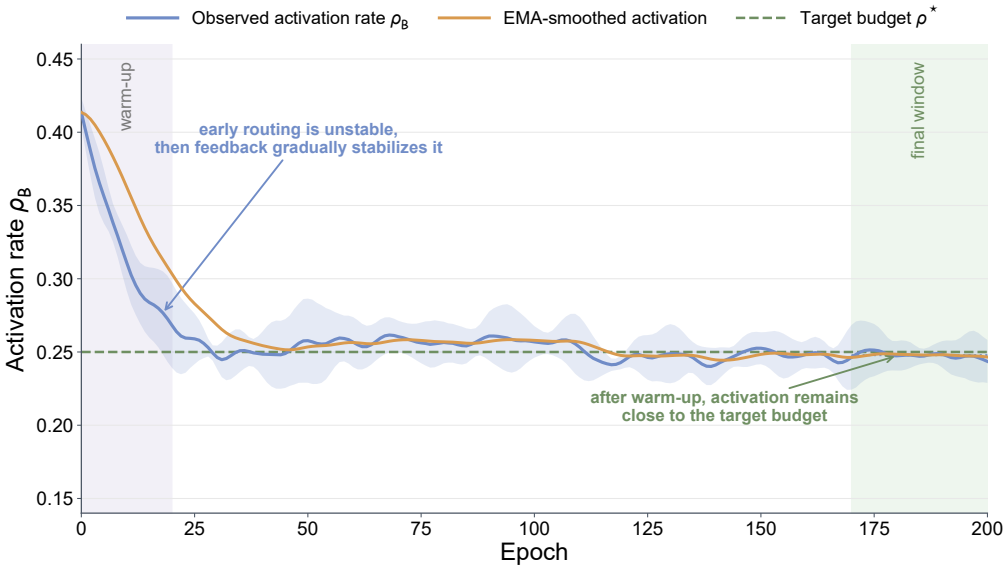

Figure 5: **Expanded routing stability diagnostic.** We report the training dynamics of the semantic-pathway activation rate on CIFAR-100-LT with IR$=100$. The observed activation rate is averaged over five runs, and shaded regions denote standard deviation. An EMA-smoothed trace is included to visualize the feedback trajectory, and the dashed line marks the target budget $\rho^\star$. After warm-up, the adaptive threshold keeps the activation rate close to the target budget, showing that the soft contribution remains controlled instead of drifting toward uniformly large weights.

provides empirical evidence that preserving and consolidating rare evidence mitigates tail signal erosion.

I Gate Warm-Up and Routing Analysis

This appendix analyzes the confidence gate used by TAILCON. The gate is not used as a tail-class detector. It improves the final model while primarily serving routing stability and activation-budget control. Its role is to stabilize routing, control the semantic-pathway activation budget, and improve the reliability of memory acquisition during early training. The analysis separates the routing mechanism from the effects of memory acquisition and consolidation.

I.1 Role of the Confidence Gate

The confidence gate controls how strongly the semantic pathway contributes to the final prediction. Given the scout prediction, we compute the margin confidence

$$C(\mathbf{x}) = P_{(1)}(\mathbf{x}) - P_{(2)}(\mathbf{x}), \tag{83}$$

where $P_{(1)}$ and $P_{(2)}$ are the largest and second-largest scout softmax probabilities. The gate is computed as

$$g(\mathbf{x}) = \text{sigmoid}\left(\kappa(\tau - C(\mathbf{x}))\right), \tag{84}$$

where κ controls gate sharpness and τ is an adaptive threshold. Low-confidence samples receive larger gate values and therefore stronger semantic-pathway contribution.

The threshold is updated by negative feedback:

$$\tau \leftarrow \tau - \zeta\left(\bar{g} - \rho^\star\right), \tag{85}$$

where $\bar{g}$ is the mini-batch average of $g(\mathbf{x})$. Since increasing τ increases the activation rate under the gate definition, this update decreases τ when the observed activation exceeds the target budget.

The gate modulates semantic contribution and maintains its average near the target budget. It uses the same scout uncertainty signal that participates in write eligibility, but the two decisions are not

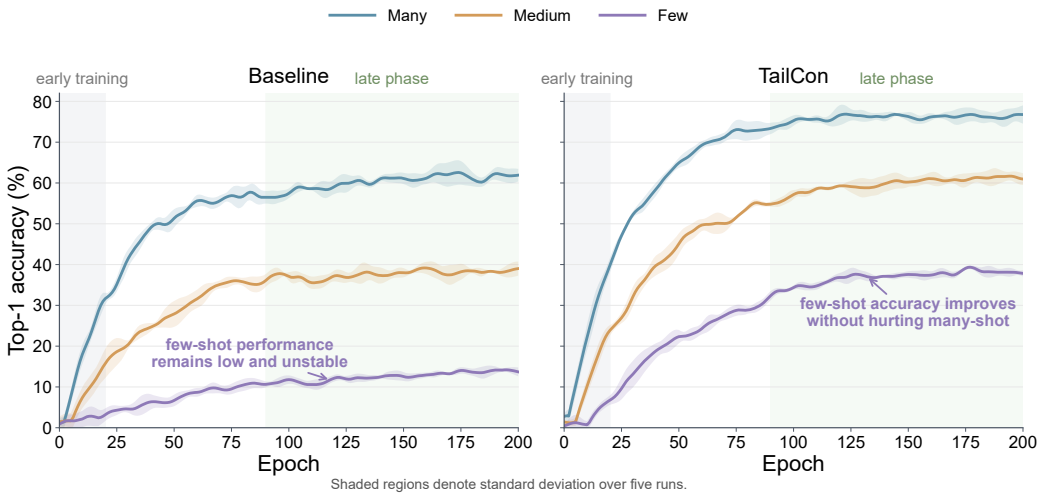

Figure 6: **Group-wise accuracy dynamics on CIFAR-100-LT with IR$=100$.** We compare the training dynamics of Many-, Medium-, and Few-shot accuracy for the baseline model and TAILCON. Curves show the mean over five runs, and shaded regions denote standard deviation. TAILCON improves Few-shot and Medium-shot accuracy substantially while also improving Many-shot accuracy, indicating that the tail gain is not achieved by sacrificing head performance but by improving the balance across groups.

Table 23: Early-routing safeguard analysis on CIFAR-100-LT with IR$=100$. Activation-rate deviation is the absolute deviation between the average semantic activation and the target budget. Memory write consistency measures whether written features are consistent with the class-specific memory structure, as defined in Eq. 89.

Variant	Overall Acc.	Few-shot Acc.	Activation Rate Dev.	Memory Write Consistency
No warm-up safeguards	48.2$\pm$0.5	23.4$\pm$0.5	0.45	45.2
LR warm-up only	51.5$\pm$0.4	26.8$\pm$0.4	0.38	58.7
Full TAILCON safeguards	**58.5**$\pm$0.4	**35.4**$\pm$0.5	**0.02**	**94.8**

identical. In particular, changing τ changes the routing weight, while uncertainty-based writing uses the separate threshold τ_{write}. Tail awareness enters writing through $N_y < t$. An average gate weight of 0.25 is not, by itself, evidence that only 25% of semantic forward passes are executed.

I.2 Early-Routing Instability

Early in training, the scout classifier is not yet reliable. If routing decisions are made too sharply at this stage, uncertain samples may receive unstable gate values and the episodic memory may accumulate noisy or unrepresentative prototypes. This issue is especially important in long-tailed recognition because rare classes have fewer opportunities to correct early memory errors.

The early-training safeguards are learning-rate warm-up, soft-to-sharp annealing of κ, and adaptive threshold feedback. Learning-rate warm-up moderates the initial parameter updates. The gate starts softly and becomes sharper as training progresses. Threshold feedback regulates the average gate weight around the target budget. Detached writes and the prohibition of evaluation-time memory updates remain common protocol rules; they are distinct from these warm-up safeguards.

Table 23 reports the effect of these safeguards on CIFAR-100-LT with IR$=100$. Without warm-up safeguards, the activation rate deviates strongly from the target and the post-hoc memory write consistency metric degrades. Learning-rate warm-up alone improves stability but remains insufficient. The full safeguard design keeps the activation rate close to the target and substantially improves memory write consistency.

Table 24: Comparison of confidence metrics on CIFAR-100-LT with IR$= 100$. The activation rate $\bar{g}$ is the average semantic-pathway activation.

Confidence metric	Overall Acc.	$\bar{g}$
Maximum softmax probability	57.9$\pm$0.3	0.26
Entropy	58.2$\pm$0.3	0.25
Margin	**58.5**$\pm$0.4	0.25

Table 25: Activation-budget analysis on CIFAR-100-LT with IR$= 100$. The default budget $\rho^\star = 0.25$ provides a stable trade-off between accuracy and semantic-pathway activation.

Target budget $\rho^\star$	Overall Acc.	Actual activation rate $\bar{g}$
0.10	57.4$\pm$0.4	0.11
0.25	**58.5**$\pm$0.4	0.25
0.50	58.4$\pm$0.3	0.49
1.00	58.3$\pm$0.4	1.00

These results show that the gate is most important as a stability mechanism. The final accuracy gain does not come from the gate alone, as shown by the component ablation in the main text. Instead, the safeguards make memory acquisition reliable enough for subsequent FOCUS consolidation.

I.3 Confidence Metrics

We compare three confidence metrics for routing: maximum softmax probability, entropy, and margin. Maximum softmax probability measures the confidence of the top predicted class. Entropy measures uncertainty over the whole predicted distribution. Margin measures the separation between the top two predictions. Table 24 shows that margin performs best among the tested metrics.

The margin score is effective because it captures decision ambiguity more directly than the top probability alone. A sample with two close competing classes can be ambiguous even if its maximum softmax probability is moderately high. This property is useful for long-tailed recognition, where rare-class samples are often confused with visually similar head classes.

I.4 Activation Budget

The target activation budget $\rho^\star$ controls how often the semantic pathway receives strong contribution. Table 25 reports the effect of different target budgets. The default value $\rho^\star = 0.25$ provides the best accuracy among the tested settings while keeping the actual activation rate close to the target.

Increasing the target budget to 1.00 gives an effectively unattenuated semantic contribution but does not improve accuracy in this comparison. The result concerns prediction weighting and routing stability; it does not by itself quantify a reduction in executed computation.

I.5 Activation Dynamics

We monitor the actual semantic activation rate during training to evaluate whether the adaptive threshold maintains the intended activation budget. Figure 7 reports the observed activation rate, the target budget, and the absolute deviation from the target across training epochs. After the warm-up phase, the activation rate remains close to $\rho^\star$, and the deviation decreases and stabilizes, indicating effective budget control rather than trivial always-on semantic-pathway activation.

I.6 Memory Write Consistency

The memory write condition determines whether a training sample should be stored in episodic memory:

$$\omega(\mathbf{x}, y) = \mathbb{I}\left[C(\mathbf{x}) < \tau_{\text{write}} \text{ or } N_y < t\right]. \tag{86}$$

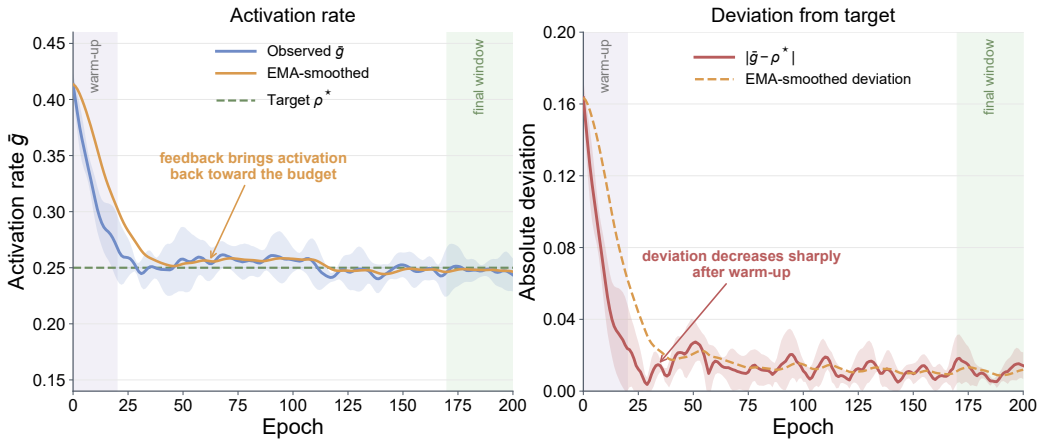

Figure 7: **Activation dynamics on CIFAR-100-LT with IR**$= 100$**.** The left panel shows the observed activation rate $\bar{g}$ and the target budget $\rho^\star$, while the right panel reports the absolute deviation $|\bar{g} - \rho^\star|$. Curves show the mean over five runs, and shaded regions denote standard deviation. After warm-up, the adaptive threshold feedback keeps the actual activation rate close to the target budget and progressively reduces the deviation, indicating stable budget control rather than trivial always-on activation.

Because the training label is known, write consistency should not be interpreted as label correctness. Instead, it measures whether the written feature is consistent with the class-specific memory structure and does not strongly conflict with prototypes from other classes.

Let $\mathcal{W}$ denote the set of memory write events. For a write event $w = (\bar{\mathbf{f}}, y)$, define

$$S_y(w) = \max_{\mathbf{p}\in\mathcal{P}_y} \bar{\mathbf{f}}^\top \mathbf{p}, \qquad S_{\neg y}(w) = \max_{c\neq y} \max_{\mathbf{p}\in\mathcal{P}_c} \bar{\mathbf{f}}^\top \mathbf{p}. \tag{87}$$

When the corresponding memory set is empty, the maximum similarity is treated as a fixed low value. We define the write-consistency indicator as

$$r(w) = \mathbb{I}\left[S_y(w) \geq S_{\neg y}(w) + \delta_{\text{pur}}\right], \tag{88}$$

where δ_{pur} is a small margin. The memory write consistency is

$$\text{Consistency} = \frac{1}{|\mathcal{W}|} \sum_{w\in\mathcal{W}} r(w). \tag{89}$$

This metric is used only for analysis and is not part of the training objective. Tables and plots express it as a percentage by multiplying the fraction in Eq. 89 by 100.

The consistency results in Table 23 show that full safeguards substantially improve memory quality. This is important because noisy or inconsistent memory writes can harm both retrieval and consolidation. FOCUS relies on replayed memory prototypes, so stabilizing memory acquisition is a prerequisite for reliable consolidation.

I.7 Interaction Between Gate and Memory

The gate and the memory write condition serve related but distinct roles. The gate controls the soft contribution of the semantic pathway. The memory write condition determines whether an example should be stored in episodic memory. Low confidence can increase semantic-pathway contribution and can also make a sample eligible for memory writing, but the two decisions are not identical because memory writing also depends on the class-frequency condition $N_y < t$.

This separation is important. If routing and writing were identical, the memory could simply become a cache of all semantically routed samples. In TAILCON, routing controls prediction contribution, while memory writing controls evidence preservation. This separation allows uncertain head samples, rare-class samples, and ambiguous boundary samples to be handled differently.

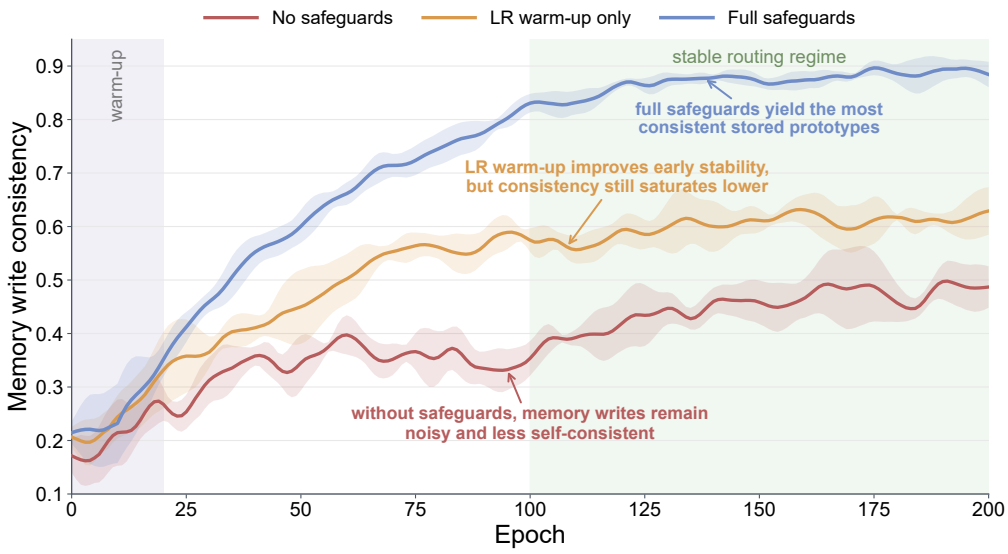

Figure 8: **Memory write consistency over training on CIFAR-100-LT with IR$=100$.** We compare the dynamics of memory write consistency under three settings: no safeguards, LR warm-up only, and full safeguards. Curves show the mean over five runs, and shaded regions denote standard deviation. Full warm-up and routing safeguards substantially improve the consistency of stored prototypes throughout training, which provides a more reliable memory signal for subsequent FOCUS consolidation.

I.8 Summary

The routing analysis supports three conclusions. First, early routing requires safeguards because the scout classifier is unreliable at the beginning of training. Second, margin-based confidence provides the best routing signal among the tested confidence metrics. Third, the default activation budget $\rho^\star = 0.25$ provides a stable accuracy and semantic-weighting trade-off. These findings justify the routing design used in TAILCON.

At the same time, the routing results should be interpreted in context. The gate stabilizes online acquisition and controls semantic-pathway activation, but the main performance gain comes from the combination of memory acquisition and FOCUS consolidation rather than from gating alone.

J Robustness and Stress Tests under Weakened Assumptions

This appendix reports robustness analyses and stress tests under weakened assumptions. These experiments are designed to evaluate whether the behavior of TAILCON remains consistent when the conditions behind the local optimization view are perturbed. They are not intended to prove a universal theory of long-tailed training. Instead, they quantify how the empirical gain changes under altered dominance, noisy supervision, memory-write corruption, calibration analysis, and protocol shifts. Unless otherwise specified, accuracy values for TAILCON and controlled variants are reported as mean $\pm$ standard deviation over repeated runs.

J.1 Purpose of the Stress Tests

The main text presents a local view of tail signal erosion in conflict-dominant long-tailed regimes. This view is intentionally scoped: rare evidence is sparse, frequent-class updates can dominate the effective update direction, and head-tail gradients can be weakly aligned or conflicting. In practice, these conditions may be weakened or perturbed.

We therefore consider three diagnostic regimes:

1. **Standard conflict-dominant long-tailed recognition.** This is the primary setting considered in the main text.

Table 26: Performance impact when the assumptions behind the local analysis are weakened. The gain is largest in the standard conflict-dominant long-tailed regime and becomes smaller when the assumptions are weakened.

Regime	Weakened assumption	Diagnostic setting	Baseline Acc.	TAILCON Acc.	Gain
Standard conflict-dominant LTR	None	CIFAR-100-LT, IR= 100	42.0±0.3	**58.5**±0.4	+16.5
Head dominance weakened	Effective head dominance	Noisy-head, clean-tail LT	20.4±0.5	**32.8**±0.4	+12.4
No-revisit protocol	Repeated exposure	Split-CIFAR-100, Class-IL	45.7±0.4	**51.6**±0.3	+5.9

Table 27: Noisy-head and clean-tail stress test on CIFAR-100-LT with IR= 100. Many-shot classes receive 80% symmetric label noise while few-shot classes remain clean.

Method	Head noise	Many	Medium	Few	All
Baseline	0%	61.2±0.4	39.5±0.5	14.4±0.4	42.0±0.3
Baseline	80%	15.3±0.5	28.1±0.4	18.2±0.5	20.4±0.5
TAILCON	80%	**22.5**±0.4	**45.6**±0.3	**31.2**±0.5	**32.8**±0.4

2. **Altered dominance.** Head-class supervision is corrupted while tail-class supervision remains clean, weakening the usual assumption that head classes provide reliable dominant updates.
3. **No-revisit class-incremental protocol.** Data are presented in a continual-learning-style protocol, weakening the repeated-exposure assumption of standard long-tailed training.

These settings test whether the gain of memory consolidation changes as the assumptions behind the analysis weaken. We do not expect TAILCON to maintain the same gain in every regime. Rather, if the proposed mechanism is correctly scoped, its gain should be largest in the standard conflict-dominant regime and may become smaller when the assumptions become less favorable.

J.2 Performance Impact under Weakened Assumptions

Table 26 summarizes the performance impact when the assumptions behind the local analysis are weakened. The largest gain appears in the standard long-tailed regime. When head dominance is weakened by noisy-head supervision, the gain becomes smaller but remains substantial. Under the no-revisit class-incremental protocol, the gain is still positive but further reduced.

The trend is consistent with the scoped interpretation of the theory. The advantage of TAILCON is largest when sparse rare evidence is most likely to be weakened by head-dominated updates. When the assumptions are weakened, the gain decreases but remains positive in the tested diagnostic settings. This supports the interpretation of TAILCON as an optimization-level mechanism rather than a universal guarantee.

J.3 Noisy-Head and Clean-Tail Stress Test

The noisy-head stress test modifies the standard long-tailed setting by corrupting the supervision of many-shot classes while keeping few-shot classes clean. This weakens the usual assumption that head classes provide reliable dominant supervision. Specifically, many-shot classes receive symmetric label noise, while few-shot classes remain clean. This setting tests whether TAILCON remains useful when the head-dominance structure becomes less favorable.

Table 27 reports the results. Compared with the standard long-tailed setting, both the baseline and TAILCON degrade substantially. However, TAILCON still retains a clear advantage. This suggests that memory consolidation remains useful when the usual head-dominance assumption is weakened, although the gain is smaller than in the standard setting.

This result should not be interpreted as evidence that TAILCON is immune to head-class noise. Rather, it shows that preserving uncertain and rare evidence remains beneficial under this altered-dominance diagnostic.

Table 28: Symmetric label noise robustness on CIFAR-100-LT with IR$=100$. Drop is measured in absolute percentage points relative to the corresponding clean-training result of each method.

Noise ratio	Method	Many	Medium	Few	All	Relative drop
0%	Baseline	61.2±0.4	39.5±0.5	14.4±0.4	42.0±0.3	–
0%	TAILCON	76.1±0.3	60.1±0.4	35.4±0.5	58.5±0.4	–
10%	Baseline	55.8±0.5	35.7±0.4	12.2±0.4	37.6±0.4	-4.4
10%	TAILCON	72.4±0.4	56.6±0.4	32.1±0.5	55.1±0.4	-3.4
20%	Baseline	48.6±0.5	31.0±0.5	9.8±0.4	32.4±0.5	-9.6
20%	TAILCON	67.8±0.4	52.3±0.4	28.6±0.5	50.7±0.4	-7.8

Table 29: Calibration analysis on CIFAR-100-LT with IR$=100$. ECE is reported in percent.

Method	Accuracy	ECE	NLL
Baseline	42.0±0.3	12.8±0.4	2.41±0.04
TAILCON	58.5±0.4	8.6±0.3	1.72±0.03
TAILCON without memory at inference	53.3±0.3	7.9±0.2	1.83±0.03

J.4 Symmetric Label Noise Robustness

We further evaluate robustness under symmetric label noise on CIFAR-100-LT with IR$=100$. This setting corrupts a fixed fraction of training labels uniformly at random across the training distribution. Unlike the noisy-head stress test, symmetric label noise affects both frequent and rare classes. This evaluates whether the memory module and FOCUS consolidation are overly sensitive to noisy memory writes.

Table 28 reports symmetric-label-noise robustness.

This experiment is important because the episodic memory stores features from the training stream. If noisy labels create incorrect memory writes, consolidation may transfer unreliable evidence into the memory-disabled predictor. In practice, TAILCON uses confidence-based routing, rare-class eligibility, prototype merging, and warm-up safeguards to reduce noisy writes. The write-consistency analysis in Appendix I provides additional evidence for this stabilization.

J.5 Calibration and Reliability

Because TAILCON combines scout, memory, and semantic evidence, we evaluate whether the fused prediction becomes poorly calibrated. Calibration is measured by expected calibration error:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{|S_b|}{n} \left|\mathrm{acc}(S_b) - \mathrm{conf}(S_b)\right|, \tag{90}$$

where S_b is the set of samples whose confidence falls into bin b, $\mathrm{acc}(S_b)$ is the empirical accuracy in the bin, and $\mathrm{conf}(S_b)$ is the average predicted confidence.

Table 29 reports calibration results on CIFAR-100-LT with IR$=100$. TAILCON improves accuracy while reducing ECE and NLL relative to the monolithic baseline. The memory-disabled variant has slightly lower accuracy than full TAILCON but retains favorable calibration, suggesting that FOCUS improves the memory-disabled predictor without relying only on overconfident memory fusion.

If further calibration is required, pathway temperatures such as τ_{mem} and τ_{sem} can be selected on the validation set without changing the training procedure.

J.6 Reliability Diagrams

Figure 9 compares reliability diagrams for the baseline, TAILCON, and memory-disabled TAILCON on CIFAR-100-LT with IR$=100$. Each panel plots empirical accuracy against predicted confidence across confidence bins. The dashed diagonal denotes ideal calibration, while the vertical gaps indicate

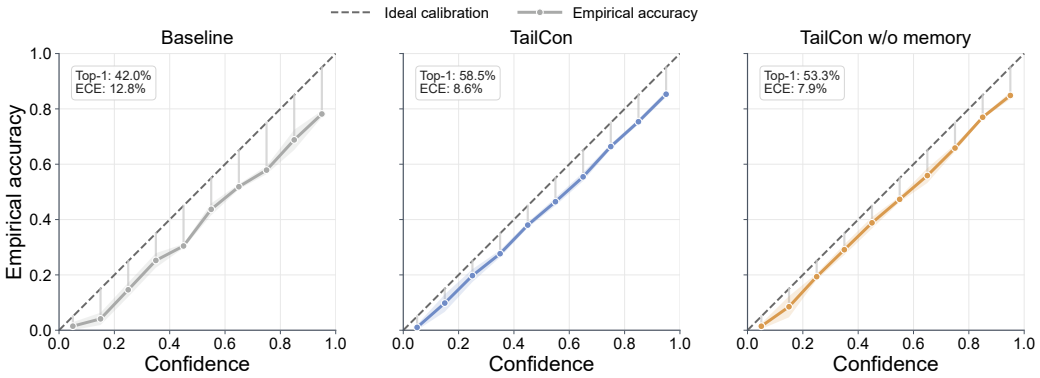

Figure 9: **Reliability diagrams on CIFAR-100-LT with IR$=100$.** We compare the calibration behavior of the monolithic baseline, TAILCON, and memory-disabled TAILCON. Each panel plots empirical accuracy against predicted confidence across confidence bins. The dashed diagonal indicates ideal calibration, vertical segments indicate calibration gaps, and shaded regions denote standard deviation over five runs. TAILCON improves accuracy while maintaining a closer correspondence between confidence and empirical correctness, and the memory-disabled variant shows that the memory-disabled predictor remains well calibrated after FOCUS.

Table 30: Robustness to corrupted memory writes on CIFAR-100-LT with IR$=100$. A fraction π of write events is assigned to an incorrect class memory.

Corruption ratio π	Many	Medium	Few	All
0%	76.1$\pm$0.3	60.1$\pm$0.4	35.4$\pm$0.5	58.5$\pm$0.4
10%	74.8$\pm$0.4	58.4$\pm$0.4	33.7$\pm$0.5	56.9$\pm$0.4
20%	72.6$\pm$0.5	55.7$\pm$0.4	30.8$\pm$0.5	54.2$\pm$0.4
30%	69.4$\pm$0.5	52.1$\pm$0.5	27.0$\pm$0.5	50.6$\pm$0.5

the discrepancy between confidence and empirical correctness. This diagnostic tests whether memory fusion improves recognition performance without introducing severe overconfidence, and whether the memory-disabled predictor remains well calibrated when the memory branch is disabled at inference.

J.7 Robustness to Memory Write Corruption

We also analyze the effect of corrupted memory writes by artificially perturbing a fraction of write events. This stress test directly targets the memory-acquisition path. For a corruption ratio π, a fraction π of memory write events is assigned to an incorrect class memory sampled uniformly from the remaining classes. This setting evaluates whether the system degrades gracefully when some stored evidence is unreliable.

Table 30 reports the results. Moderate corruption reduces accuracy but does not cause immediate collapse. As the corruption ratio increases, performance degrades more substantially, indicating that high-quality memory acquisition remains important for reliable consolidation.

This test separates two questions. The first is whether TAILCON benefits from high-quality memory writes. The second is whether the system collapses when a moderate fraction of writes is corrupted. The observed degradation pattern suggests that memory consolidation is useful but still depends on reliable memory acquisition, as expected.

J.8 Interpretation of Robustness Results

The stress tests support the following interpretation. In the standard long-tailed setting, TAILCON has the largest gain because rare evidence is sparse and frequent-class updates can dominate the stochastic training trajectory. When head dominance is weakened or the protocol changes, the gain becomes smaller. This is consistent with the local theory, which predicts that the advantage of separating acquisition from consolidation should be strongest in conflict-dominant regimes.

Table 31: Default control hyperparameters used by TAILCON in the main experiments.

Hyperparameter	Default value	Role
Target activation rate $\rho^\star$	0.25	Semantic-pathway activation budget
Threshold feedback rate ζ	0.01	Adaptive threshold update step
Gate sharpness κ	$1 \to 50$	Soft-to-sharp routing schedule
Rare-class cutoff t	20	Rare-class memory write eligibility
Memory write threshold τ_{write}	0.3	Uncertainty-based memory write eligibility
Merge threshold θ_{merge}	0.95	Prototype merge criterion
Maximum prototypes per class $B_{\max}$	10	Memory capacity control
FOCUS interval K	5	Consolidation frequency in epochs
Memory weight λ_{mem}	0.5	Weight of memory logits
Semantic weight λ_{sem}	1.0	Weight of semantic logits
Memory temperature τ_{mem}	2.0	Memory-logit calibration
Semantic temperature τ_{sem}	1.0	Semantic-logit calibration
Distillation temperature τ_T	3.0	FOCUS distillation temperature
Alignment module	Linear	Low-capacity consolidation pathway
Candidate classes k_{cand}	All for CIFAR, 50 for large-scale datasets	Retrieval acceleration

The noisy-label and memory-write corruption tests evaluate a different concern: whether the episodic memory becomes brittle when stored evidence is unreliable. The warm-up safeguards, confidence-based routing, and prototype merge rules are designed to reduce this risk. The results show graceful degradation under moderate perturbations rather than an immediate failure of the memory pathway.

J.9 Non-Claims

These robustness experiments should be interpreted with the same scope as the main theory. They do not imply that TAILCON guarantees improvement under arbitrary noise, arbitrary distribution shift, or arbitrary continual-learning protocols. They only show how the method behaves under targeted stress tests that weaken or perturb the assumptions behind the main analysis.

In particular, the class-incremental diagnostic in Table 26 is included to quantify behavior under a no-revisit protocol. It is not used as evidence of a strict equivalence between long-tailed recognition and continual learning. A more detailed description of that protocol is provided in Appendix L.

J.10 Summary

Overall, the robustness analyses show that the gain of TAILCON is largest in the standard conflict-dominant long-tailed regime, decreases when the assumptions are weakened, and remains positive in the tested diagnostic settings. The noisy-label, calibration, and memory-corruption analyses further indicate that the memory pathway degrades gracefully under moderate perturbations rather than acting as a brittle retrieval cache. These findings are consistent with the scoped local analysis in Section 3.

K Hyperparameter Sensitivity and Scalability

This appendix reports hyperparameter sensitivity and scalability analyses for TAILCON. The main experiments use a single default configuration across datasets whenever possible. Here, we examine whether the method is stable with respect to the most important control parameters, including the semantic activation budget, memory merge threshold, rare-class cutoff, FOCUS frequency, memory budget, alignment-module capacity, and candidate-class shortlist size. The purpose of this analysis is to verify that the performance of TAILCON does not depend on a narrow or fragile hyperparameter setting.

K.1 Default Configuration

Table 31 summarizes the default configuration used by TAILCON in the main experiments. These values are shared across datasets unless otherwise stated. Dataset-specific optimizer, schedule, and backbone details are provided in Appendix D.

Table 32: Sensitivity to target activation rate $\rho^\star$ on CIFAR-100-LT with IR$= 100$. The default value $\rho^\star = 0.25$ provides the best accuracy among the tested settings while keeping the actual activation rate close to the target.

Target budget $\rho^\star$	Overall Acc.	Actual activation rate $\bar{g}$
0.10	57.4$\pm$0.4	0.11
0.25	**58.5**$\pm$0.4	0.25
0.50	58.4$\pm$0.3	0.49
1.00	58.3$\pm$0.4	1.00

Table 33: Sensitivity to memory merge threshold θ_{merge} on CIFAR-100-LT with IR$= 100$.

θ_{merge}	Many	Medium	Few	All	Avg. prototypes/class	Observation
0.80	74.9$\pm$0.4	58.2$\pm$0.4	32.6$\pm$0.5	56.4$\pm$0.3	3.1	Aggressive merging
0.85	75.6$\pm$0.3	59.2$\pm$0.4	34.1$\pm$0.4	57.6$\pm$0.3	4.7	Moderate merging
0.90	75.8$\pm$0.3	59.7$\pm$0.3	34.8$\pm$0.4	58.1$\pm$0.3	6.3	Conservative merging
0.95	**76.1**$\pm$0.3	**60.1**$\pm$0.4	**35.4**$\pm$0.5	**58.5**$\pm$0.4	8.4	Default
0.98	75.7$\pm$0.4	59.4$\pm$0.4	34.9$\pm$0.5	58.0$\pm$0.4	9.7	More prototype fragmentation

K.2 Sensitivity to Activation Budget

The target activation rate $\rho^\star$ controls the average semantic-pathway activation. A very small budget may underuse the semantic pathway, while a very large budget may remove the benefit of selective routing. Table 32 reports the sensitivity on CIFAR-100-LT with IR$= 100$.

An effectively unattenuated semantic contribution does not improve accuracy in this comparison. The gate controls prediction weighting and routing stability; the soft activation statistic is not a direct measurement of computational savings.

K.3 Sensitivity to Memory Merge Threshold

The memory merge threshold θ_{merge} controls whether an eligible feature is merged into an existing prototype or inserted as a new prototype. A low threshold may over-merge distinct modes within a class, while a high threshold may create redundant prototypes. Table 33 reports the sensitivity to θ_{merge}.

The results show that high but not extreme thresholds provide a good trade-off between prototype compactness and intra-class diversity. If θ_{merge} is too low, visually distinct samples may collapse into a small number of prototypes. If it is too high, the memory becomes more fragmented and retrieval cost increases without clear accuracy gains.

K.4 Sensitivity to Rare-Class Cutoff

The rare-class cutoff t determines whether a class is automatically eligible for memory writing based on its training count. The default value is $t = 20$, matching the Few-shot threshold used in common long-tailed recognition protocols. Table 34 reports the sensitivity results.

A very small cutoff may fail to protect medium-frequency classes whose evidence is still sparse or unstable. A very large cutoff writes too many samples into memory and reduces selectivity. The default $t = 20$ provides the best balance among the tested settings.

K.5 Sensitivity to FOCUS Interval

The FOCUS interval K controls how often memory prototypes are replayed and distilled into the parametric prediction pathway. Frequent consolidation can transfer memory evidence quickly but increases training cost. Infrequent consolidation reduces overhead but delays parametric integration. Table 35 reports the sensitivity to K.

Table 34: Sensitivity to rare-class cutoff t on CIFAR-100-LT with IR$= 100$.

Rare cutoff t	Many	Medium	Few	All	Memory writes	Observation
10	75.2±0.4	58.7±0.4	33.1±0.5	57.2±0.3	0.74×	More selective rare writes
20	**76.1**±0.3	**60.1**±0.4	**35.4**±0.5	**58.5**±0.4	1.00×	Default
50	75.5±0.3	59.8±0.4	35.0±0.5	58.1±0.3	1.42×	More classes treated as rare
100	74.7±0.4	58.9±0.4	34.5±0.5	57.4±0.4	1.86×	Less selective memory writing

Table 35: Sensitivity to FOCUS interval K on CIFAR-100-LT with IR$= 100$.

FOCUS interval K	Many	Medium	Few	All	Training overhead
1	76.0±0.3	**60.3**±0.3	**35.5**±0.5	**58.6**±0.3	High
5	**76.1**±0.3	60.1±0.4	35.4±0.5	58.5±0.4	Moderate
10	75.5±0.4	59.2±0.4	34.3±0.5	57.7±0.4	Low
No FOCUS	74.9±0.4	56.5±0.3	31.3±0.5	55.9±0.4	0

The comparison with the no-FOCUS setting is particularly important. Removing FOCUS reduces the overall accuracy from 58.5% to 55.9%, indicating that memory consolidation is a core component rather than an auxiliary training detail. The default interval $K = 5$ provides nearly the same accuracy as more frequent consolidation while keeping training overhead moderate.

K.6 Sensitivity to Alignment-Module Capacity

FOCUS uses a restricted memory-disabled predictor to transfer memory evidence into the classifier. The capacity of this pathway should be sufficient for alignment but limited enough to avoid overfitting sparse prototypes. Table 36 reports the sensitivity to consolidation capacity.

The default consolidation capacity provides the best balance between alignment and stability. A smaller pathway underfits the transfer from memory prototypes to classifier logits, while a larger pathway can overfit sparse prototypes during consolidation.

K.7 Memory Size and Prototype Budget

The memory budget controls how many prototypes are stored per class. In the reported implementation, TAILCON uses a fixed per-class prototype cap:

$$B_c = B_{\max}, \tag{91}$$

where $B_{\max}$ is the maximum number of prototypes per class. This keeps memory usage bounded and avoids linear growth with the number of training samples. Table 37 reports sensitivity to the prototype budget.

Too small a memory budget reduces prototype coverage for rare or diverse classes. Increasing the budget beyond the default slightly improves Few-shot and overall accuracy but also increases memory size. The default setting provides a practical trade-off between accuracy and retrieval cost.

K.8 Candidate-Class Shortlist Size

For large-scale datasets, TAILCON accelerates memory retrieval by using scout logits to shortlist candidate classes before prototype search. Table 38 reports sensitivity to candidate-class shortlist size on ImageNet-LT and iNaturalist 2018.

A small candidate set reduces retrieval cost but may exclude useful memory evidence, while an overly large candidate set can introduce noisy or irrelevant prototype matches. The results show that the default $k_{\text{cand}} = 50$ achieves the best Top-1 accuracy on both ImageNet-LT and iNaturalist 2018 while using only a small fraction of the exact retrieval search space. CIFAR-scale datasets use exact cosine search over all stored prototypes because the number of classes is small.

K.9 Large-Scale Hyperparameter Stability

The main text reports results on ImageNet-LT and iNaturalist 2018 using the same default control parameters whenever possible. Table 39 summarizes large-scale sensitivity results for the memory

Table 36: Sensitivity to alignment-module capacity on CIFAR-100-LT with IR$=100$. The default setting updates a lightweight linear alignment module and the classifier head during FOCUS.

Consolidation capacity	Many	Medium	Few	All	Updated parameters
Low	75.0±0.4	58.1±0.4	32.8±0.5	56.8±0.4	Classifier head only
Default	**76.1**±0.3	**60.1**±0.4	**35.4**±0.5	**58.5**±0.4	Alignment module + classifier head
High	75.4±0.4	59.0±0.4	34.2±0.5	57.6±0.4	Larger prediction head

Table 37: Sensitivity to memory budget on CIFAR-100-LT with IR$=100$.

Budget setting	Target prototypes/class	Max prototypes/class	Avg. prototypes/class	Few Acc.	All Acc.	Relative memory size
Small	4	5	4.2	32.7±0.5	56.6±0.4	0.54×
Default	8	10	8.4	**35.4**±0.5	58.5±0.4	1.00×
Large	16	20	14.8	**35.7**±0.4	**58.7**±0.3	1.76×

write threshold and memory merge threshold, showing that the default values remain stable beyond CIFAR-scale datasets.

The large-scale results show the same qualitative trend as the CIFAR-scale sensitivity analysis. Moderate changes around the default values do not substantially degrade performance, while overly aggressive or overly conservative settings are less effective.

Direction of the write-threshold effect. For fixed scout confidences, increasing τ_{write} enlarges the set satisfying $C(\mathbf{x}) < \tau_{\text{write}}$. Thus, 0.2 is more selective than 0.3, while 0.4 permits more uncertainty-based writes. The separate rare-class condition $N_y < t$ remains unchanged.

K.10 Computational Scalability

Table 40 reports the computational scalability profile of TAILCON. The first block reports measured end-to-end inference cost on CIFAR-100-LT. The second block reports large-scale prototype-memory and retrieval-search scalability under the batch-size-256 profiling configuration. For the large-scale rows, GFLOPs are estimated from a ResNet-50 execution anchor plus the deterministic prototype-retrieval dot-product cost. Latency and peak VRAM are taken from the observed batch-size-256 profiling runs. This separates backbone-level execution cost from the additional cost introduced by prototype retrieval.

On CIFAR-100-LT, TAILCON increases GFLOPs from 0.14 to 0.19 and latency from 4.4 ms to 6.1 ms, while improving overall accuracy from 42.0% to 58.5%. The additional inference cost is modest compared with a naive three-expert model, which requires 0.42 GFLOPs, 13.2 ms latency, and 4.5 GB peak VRAM. This comparison shows that TAILCON improves rare-evidence preservation without replicating full parametric branches.

The memory branch is bounded by the per-class prototype cap. Let C be the number of classes, $B_{\max}$ the maximum number of prototypes per class, and d the prototype feature dimension. The number of stored prototypes is bounded by

$$|\mathcal{M}| \leq CB_{\max}, \tag{92}$$

and the prototype storage scales as $O(CB_{\max}d)$. With $B_{\max} = 10$ and $d = 2048$ for the large-scale ResNet-50 setting, the upper bound is 10,000 prototypes on ImageNet-LT and 81,420 prototypes on iNaturalist 2018. The corresponding prototype storage is 78.12 MB and 636.09 MB, respectively. These values remain small relative to the observed batch-size-256 profiling memory, indicating that the prototype bank is not the dominant GPU-memory bottleneck.

Retrieval cost is controlled by candidate-class shortlisting. For CIFAR-100-LT, exact retrieval over all classes is inexpensive because $C = 100$. For ImageNet-LT and iNaturalist 2018, TAILCON uses scout logits to shortlist $k_{\text{cand}} = 50$ candidate classes before prototype retrieval. With a fixed prototype cap, exact retrieval scales as

$$O(CB_{\max}d), \tag{93}$$

whereas shortlisted retrieval scales as

$$O(k_{\text{cand}}B_{\max}d). \tag{94}$$

Table 38: Sensitivity to candidate-class shortlist size k_{cand} on large-scale datasets. Relative search cost is normalized by exact retrieval over all classes.

Dataset	k_{cand}	Top-1 Acc.	Relative search cost	Observation
ImageNet-LT	20	62.6±0.4	2.00×10^{-2}	Aggressive shortlist
ImageNet-LT	50	**63.2**±0.4	5.00×10^{-2}	Default
ImageNet-LT	100	61.9±0.4	1.00×10^{-1}	Wider shortlist
ImageNet-LT	All	62.4±0.5	1.00×	Exact retrieval
iNaturalist 2018	20	76.1±0.3	2.46×10^{-3}	Aggressive shortlist
iNaturalist 2018	50	**76.9**±0.3	6.14×10^{-3}	Default
iNaturalist 2018	100	75.7±0.4	1.23×10^{-2}	Wider shortlist
iNaturalist 2018	All	76.6±0.3	1.00×	Exact retrieval

Table 39: Large-scale hyperparameter sensitivity on ImageNet-LT and iNaturalist 2018.

Dataset	Hyperparameter setting	Value	Top-1 Acc.	Observation
ImageNet-LT	Memory write threshold τ_{write}	0.2	62.7±0.4	Fewer uncertainty-based writes
ImageNet-LT	Memory write threshold τ_{write}	0.3	**63.2**±0.4	Default
ImageNet-LT	Memory write threshold τ_{write}	0.4	62.9±0.4	More uncertainty-based writes
ImageNet-LT	Merge threshold θ_{merge}	0.80	62.5±0.5	More merging
ImageNet-LT	Merge threshold θ_{merge}	0.95	**63.2**±0.4	Default
ImageNet-LT	Merge threshold θ_{merge}	0.98	63.0±0.4	More prototypes
iNaturalist 2018	Memory write threshold τ_{write}	0.2	76.4±0.3	Fewer uncertainty-based writes
iNaturalist 2018	Memory write threshold τ_{write}	0.3	**76.9**±0.3	Default
iNaturalist 2018	Memory write threshold τ_{write}	0.4	76.6±0.4	More uncertainty-based writes
iNaturalist 2018	Merge threshold θ_{merge}	0.80	76.2±0.4	More merging
iNaturalist 2018	Merge threshold θ_{merge}	0.95	**76.9**±0.3	Default
iNaturalist 2018	Merge threshold θ_{merge}	0.98	76.7±0.3	More prototypes

The search fraction is therefore k_{cand}/C, which is 5.00×10^{-2} on ImageNet-LT and 6.14×10^{-3} on iNaturalist 2018 under the default $k_{\text{cand}} = 50$ setting. Using a multiply-add dot-product estimate, the retrieval term is

$$G_{\text{retr}} = \frac{2k_{\text{cand}}B_{\max}d}{10^9} \quad \text{GFLOPs.} \tag{95}$$

This term is only about 0.002 GFLOPs for $k_{\text{cand}} = 50$, but grows to about 0.333 GFLOPs for exact retrieval on iNaturalist 2018. The large-scale rows show that increasing the retrieval width does not improve accuracy in the reported settings. The default scout shortlist achieves the best Top-1 accuracy on both ImageNet-LT and iNaturalist 2018 while using only 5.00×10^{-2} and 6.14×10^{-3} of the exact retrieval search space, respectively. This suggests that scout-based shortlisting reduces retrieval cost and may also reduce the influence of irrelevant prototype matches from a larger retrieval pool.

The large-scale peak-VRAM entries refer to the stated batch-size-256 profiling configuration, not to prototype storage alone. The prototype-bank estimate and whole-run peak allocation are different quantities and should not be used interchangeably. The CIFAR end-to-end inference measurements and the large-scale profiles are reported in separate blocks because their profiling configurations differ. Overall, TAILCON provides a practical scalability trade-off: modest CIFAR-scale inference overhead, bounded prototype memory, and efficient shortlisted retrieval for large-scale long-tailed recognition.

K.11 Training-Time Overhead

FOCUS adds training work through periodic prototype replay. With $S_{\text{focus}} = 200$ update steps at each scheduled consolidation phase, the scheduled number of extra head-update steps over E epochs is $S_{\text{focus}}\lfloor E/K \rfloor$ when the memory is nonempty at each phase. Only A and H_{cls} are updated during these steps, with the feature extractor and semantic pathway frozen. This update count describes the schedule and is not a measured wall-clock overhead. Inference GFLOPs, latency, and peak allocation do not determine the additional training time.

Table 40: Computational scalability profile of TAILCON. The CIFAR-100-LT block reports measured end-to-end inference cost. The large-scale block reports prototype-memory and retrieval-search scaling under the batch-size-256 profiling configuration. For ImageNet-LT and iNaturalist 2018, the default retrieval setting uses $k_{\text{cand}} = 50$ candidate classes selected by the scout classifier. Search fraction is normalized by exact prototype search over all classes. Large-scale GFLOPs are estimated from a ResNet-50 backbone anchor plus the prototype-retrieval dot-product cost. Latency is reported per image and measured under the batch-size-256 profiling configuration. Peak VRAM follows the observed batch-size-256 profiling configuration.

Setting	Method / retrieval mode	Candidate classes	Search fraction	Max prototypes	Prototype memory	GFLOPs	Latency	Peak VRAM	Top-1 Acc.
End-to-end inference cost on CIFAR-100-LT									
CIFAR-100-LT	Baseline	–	–	–	–	0.14	4.4 ms	1.5 GB	42.0±0.3
CIFAR-100-LT	Naive expert 3×	–	–	–	–	0.42	13.2 ms	4.5 GB	–
CIFAR-100-LT	TAILCON	all / 100	1.00×	1,000	0.24 MB	0.19	6.1 ms	1.8 GB	58.5±0.4
Prototype-memory and retrieval scalability on large-scale datasets									
ImageNet-LT	Scout shortlist	50 / 1000	5.00×10^{-2}	10,000	78.12 MB	5.61	16.4 ms	33.7 GB	63.2±0.4
ImageNet-LT	Wider shortlist	100 / 1000	1.00×10^{-1}	10,000	78.12 MB	5.63	16.7 ms	33.8 GB	61.9±0.4
ImageNet-LT	Exact retrieval	all / 1000	1.00×	10,000	78.12 MB	5.65	17.5 ms	34.2 GB	62.4±0.5
iNaturalist 2018	Scout shortlist	50 / 8142	6.14×10^{-3}	81,420	636.09 MB	5.62	16.5 ms	33.2 GB	76.9±0.3
iNaturalist 2018	Wider shortlist	100 / 8142	1.23×10^{-2}	81,420	636.09 MB	5.64	16.9 ms	33.4 GB	75.7±0.4
iNaturalist 2018	Exact retrieval	all / 8142	1.00×	81,420	636.09 MB	5.95	22.1 ms	35.1 GB	76.6±0.3

K.12 Practical Sensitivity and Untested Replay Choices

The default control configuration is shared across datasets: target gate weight 0.25, write threshold 0.3, merge threshold 0.95, rare cutoff 20, at most 10 prototypes per class, and a FOCUS interval of five epochs. The uncertainty and rarity exponents are fixed at $\beta = 1.5$ and $\gamma = 0.5$, with $\epsilon_u = 10^{-6}$. Backbone, augmentation, and optimization schedules follow the dataset-specific recipes in Appendix D.2.

The tested factors have different practical importance. Memory corruption is the clearest failure mode: increasing the corrupted-write fraction from 0% to 30% reduces All from 58.5 to 50.6. Removing FOCUS reduces All to 55.9, whereas consolidating every epoch or every five epochs gives 58.6 and 58.5, respectively. Merge threshold and prototype capacity have moderate effects, with diminished returns beyond the default memory budget. Average gate targets from 0.25 to 1.00 yield All between 58.3 and 58.5. Large-scale write-threshold changes from 0.2 to 0.4 produce only small changes around the default results in Table 39.

These observations support reuse of the shared defaults within the tested settings. They do not establish robustness to arbitrary corruption or to all interactions among hyperparameters. Alternative replay-distribution families and joint sweeps of the replay exponents are not exhaustively evaluated. The controlled label-only comparison in Appendix G changes the supervision target under the same replay setup; it should not be read as a comparison against every possible class-balanced or tail-only sampler.

K.13 Summary

The sensitivity analysis indicates that TAILCON is reasonably stable under moderate changes to the main control hyperparameters. The default activation budget $\rho^\star = 0.25$ provides a stable accuracy and semantic-weighting trade-off, and removing FOCUS clearly reduces performance. Sensitivity to memory merge threshold, rare-class cutoff, consolidation interval, memory budget, alignment-module capacity, and candidate shortlist size further shows that the method remains stable under moderate hyperparameter changes. The scalability analysis shows that TAILCON adds modest inference overhead compared with the monolithic baseline while remaining much lighter than a naive multi-expert model. On large-scale datasets, the default $k_{\text{cand}} = 50$ shortlist provides the best accuracy-efficiency trade-off, reducing retrieval search by one to two orders of magnitude while preserving the strongest Top-1 accuracy among the tested retrieval widths.

L Continual-Learning Diagnostic

This appendix reports a limited continual-learning diagnostic. The experiment is included only as an auxiliary analysis and is not used as evidence of strict equivalence between long-tailed recognition

Table 41: Class-incremental diagnostic on Split-CIFAR-100. This experiment is used only as an auxiliary diagnostic and is not evidence of strict equivalence between long-tailed recognition and continual learning.

Method	Protocol	Final Avg. Acc.	Gain
LUCIR	Class-IL	45.7±0.4	–
TAILCON	Class-IL	**51.6**±0.3	+5.9

and continual learning. The main paper studies long-tailed recognition, where the training set is static and imbalanced. In contrast, continual learning presents data in a task sequence and often restricts access to earlier data. We therefore use this diagnostic only to test whether the preservation and consolidation mechanism of TAILCON remains useful under a no-revisit protocol.

L.1 Protocol

We evaluate a class-incremental protocol on Split-CIFAR-100. The 100 classes are divided into sequential tasks. At each task, the model receives data from the current task classes and does not revisit the full training data from previous tasks. This differs from standard long-tailed recognition because the data distribution changes across tasks and previous classes are not repeatedly sampled during later training.

Let $\mathcal{T}_1, \ldots, \mathcal{T}_M$ denote the task sequence, and let $\mathcal{C}_m$ be the set of classes introduced at task m. After training on task m, the model is evaluated on all classes observed so far:

$$\mathcal{C}_{1:m} = \bigcup_{j=1}^{m} \mathcal{C}_j. \tag{96}$$

The average accuracy after task m is

$$A_m = \frac{1}{|\mathcal{D}_{1:m}^{\text{test}}|} \sum_{(\mathbf{x},y)\in\mathcal{D}_{1:m}^{\text{test}}} \mathbb{I}\left[\arg\max_c z_c(\mathbf{x}) = y\right], \tag{97}$$

where $\mathcal{D}_{1:m}^{\text{test}}$ contains test examples from the classes in $\mathcal{C}_{1:m}$. We report the final average accuracy after completing the last task.

L.2 Compared Methods

We compare TAILCON with LUCIR, a representative class-incremental learning baseline. The baseline is included only as a reference point for the no-revisit setting. This appendix is not intended to establish a new continual-learning benchmark result for TAILCON.

TAILCON is adapted to this protocol without changing its core design. The episodic memory preserves uncertain or rare evidence from the observed stream, and FOCUS periodically consolidates stored evidence into the parametric prediction pathway. No future-task samples are used during training. At inference, the model predicts over all classes observed so far.

L.3 Results

Table 41 reports the Split-CIFAR-100 class-incremental diagnostic. TAILCON improves the final average accuracy from 45.7% to 51.6%, corresponding to a gain of 5.9 points. This gain is smaller than the gain observed in the standard CIFAR-100-LT setting, which is consistent with the scoped interpretation of the theory. The no-revisit protocol weakens the repeated-exposure assumption of long-tailed training, so the expected advantage of TAILCON should decrease rather than remain identical.

L.4 Complementary Forgetting Metrics

Forgetting compares the best historical accuracy of each task with its final accuracy and requires the full task-wise accuracy matrix. Let $a_{m,j}$ denote the accuracy on task j after training task m. The

average forgetting after the final task M is

$$F = \frac{1}{M-1} \sum_{j=1}^{M-1} \left(\max_{m \in \{j,\ldots,M-1\}} a_{m,j} - a_{M,j} \right). \tag{98}$$

Backward transfer can be measured as

$$\mathrm{BWT} = \frac{1}{M-1} \sum_{j=1}^{M-1} \left(a_{M,j} - a_{j,j} \right). \tag{99}$$

The diagnostic reports final average accuracy, not numerical forgetting or backward-transfer scores. Eqs. 98 and 99 define distinct complementary metrics; they cannot be recovered from the final aggregate accuracy alone.

L.5 Interpretation

The result in Table 41 supports a limited conclusion. Memory preservation and consolidation remain useful under a no-revisit protocol, but the gain is smaller than in the standard conflict-dominant long-tailed regime. This is consistent with the local analysis in Section 3: when the assumptions behind the local view are weakened, the advantage of separating evidence acquisition from parametric consolidation should decrease.

This diagnostic should not be overinterpreted. It does not show that long-tailed recognition and continual learning are equivalent. It only shows that the same preservation-and-consolidation mechanism can provide a positive optimization-level benefit in a related setting where sparse evidence may also be weakened by subsequent updates.

L.6 Non-Claims

We emphasize the following non-claims:

1. We do not claim that long-tailed recognition is a special case of continual learning.
2. We do not claim that TAILCON is a general-purpose continual-learning method.
3. We do not use this diagnostic as a main benchmark for TAILCON.
4. We use the result only to test whether preserving and consolidating sparse evidence can remain useful beyond the standard long-tailed setting.

L.7 Summary

This appendix provides a narrow auxiliary diagnostic under a no-revisit class-incremental protocol. TAILCON obtains a positive gain over the class-incremental baseline, but the gain is smaller than in the standard long-tailed recognition setting. This behavior is consistent with the scoped theory: the benefit of memory consolidation is expected to be strongest when sparse rare evidence is repeatedly exposed to head-dominated updates, and weaker when the training protocol changes substantially. Therefore, this diagnostic supports the broader mechanism of evidence preservation and consolidation without expanding the main claim beyond long-tailed recognition.

M Limitations and Broader Impact

M.1 Limitations

Memory dependence and cost. TAILCON is a hybrid memory-parametric recognition system. Its default predictor uses the training-derived prototype bank, and memory-enabled accuracy remains higher than memory-disabled accuracy across the evaluated datasets. Memory-disabled inference is an attribution protocol, not the proposed default deployment mode. Bounded per-class storage limits growth with the number of training examples, but total storage still grows with class count and feature dimension. Retrieval and periodic consolidation introduce additional implementation and execution cost compared with a single parametric classifier.

Memory quality and representation drift. Prototypes summarize eligible training observations and can inherit annotation errors, inadequate intra-class coverage, or stale feature coordinates. Detached writes do not provide invariance to later representation changes. Continual updates, strength-weighted merging, and age-aware replacement improve the tested memory design, but the frozen-prototype and corrupted-write results show that memory quality remains important. Prototype memory is effective in the evaluated equal-storage comparison; it is not established to dominate all instance-queue or exemplar-selection designs.

Scope of consolidation. During FOCUS, the shallow extractor and semantic pathway are frozen, while the linear alignment module and classifier head are updated. This supports classifier-head consolidation, not full-representation reconstruction. When tail-class failure is primarily caused by severe feature collapse or absent discriminative information, updating the head alone may be insufficient. Subsequent ordinary training continues to update the parametric pathway, but the consolidation objective itself does not directly retrain the frozen representation modules.

Diagnostics and evaluation scope. The theoretical analysis is local and conditional on the head-tail diagnostic split. Class frequency creates a structural imbalance in exposure, while the tail-loss effect of a particular update also depends on gradient magnitude and alignment. The analysis does not guarantee improved accuracy under arbitrary distributions or equate fixed-dataset long-tailed training with continual learning. Detailed gradient, memory-design, and teacher-attribution studies concentrate on CIFAR-100-LT with IR$= 100$; the cross-dataset accuracy and memory-disabled results broaden the evidence without replacing equally detailed mechanistic studies on every large-scale benchmark.

Hyperparameters and related corrections. Shared defaults perform consistently within the tested neighborhoods, but the memory threshold, capacity, routing weight, and consolidation schedule interact. The replay exponents are fixed, and alternative replay families are not exhaustively compared. The logit-adjustment experiment supports compatibility under its stated protocol, not superiority over every prior-correction approach. Prior2Posterior is discussed conceptually without a matched experiment. Scout-based shortlisting also makes retrieval depend on the candidate classes made available by the scout.

M.2 Broader Impact

Improved rare-category recognition may benefit biodiversity monitoring, industrial inspection, and other applications where infrequent categories are practically important. Nevertheless, class imbalance can reflect collection patterns, annotation practices, geographic coverage, or unequal sampling effort. Higher aggregate accuracy does not establish fairness across populations or reliability in every deployment setting. Domain-specific validation, calibration assessment, dataset auditing, and appropriate human oversight remain necessary when applying the method beyond the evaluated recognition benchmarks.

The memory bank retains training-derived features. Stop-gradient storage is an optimization choice and does not provide a privacy guarantee; avoiding storage of raw images is not sufficient to establish that retained prototypes contain no sensitive information. Privacy-sensitive deployments should assess retention, access control, and applicable data-use restrictions. Mislabeled or corrupted prototypes can also influence both direct retrieval and the distilled classifier, so memory provenance and quality deserve attention. Finally, prototype storage and periodic replay add computational cost, which should be considered alongside recognition gains and the requirements of the target application.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the main contributions of TAILCON, including the tail-signal-erosion perspective, detached episodic memory acquisition, FOCUS consolidation, and empirical validation. The claims are scoped to long-tailed visual recognition and conflict-dominant regimes, and the paper explicitly states that the local optimization analysis is diagnostic rather than a universal guarantee.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Appendix M discusses scope, assumptions, scalability, practical deployment, broader impact, and environmental considerations. The main text and Appendix B also clarify that the theory is local and diagnostic, and does not imply universal gradient dominance, equivalence to continual learning, or global orthogonality.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: Section 3 states the diagnostic quantities and local assumptions used in the analysis. Appendix B provides the full derivations, including the local tail-drift condition, smoothness remainder, dominance/alignment interpretation, sparse-recurrence analysis, and the stated non-claims.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper provides architecture details, training recipes, hyperparameters, memory read/write rules, FOCUS implementation details, inference protocols, dataset statistics, and evaluation protocols in Appendices C–E. We also include an anonymized implementation in the supplementary material, with scripts and README instructions for reproducing the main experiments.

   Guidelines:

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

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: We provide an anonymized code package in the supplementary material, including configuration files, training and evaluation scripts, and README instructions for reproducing the main results and ablations. The experiments use public benchmark datasets; we do not redistribute raw datasets and instead provide instructions for obtaining them from their official sources.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 5 summarizes the datasets, metrics, implementation choices, and comparison protocol. Appendix D reports backbone choices, optimizer settings, learning-rate schedules, data augmentation, memory budgets, merge thresholds, confidence-gate parameters, FOCUS settings, and reproducibility notes. Appendix E describes dataset statistics, shot splits, evaluation metrics, and inference protocols.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Main results, core ablations, and attribution controls for TAILCON are reported as mean $\pm$ standard deviation over repeated runs with different random seeds. The tables explicitly indicate uncertainty ranges for the reported accuracy values.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper reports inference GFLOPs, latency, and peak VRAM in Table 5. Appendix K reports computational scalability, prototype-memory cost, candidate-shortlist behavior, and training-time overhead. The supplementary code package also includes configuration files specifying the experimental settings used to reproduce the reported results.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

Justification: We have reviewed the NeurIPS Code of Ethics and have made a good-faith effort to comply with it. The work uses public supervised image-classification benchmarks and does not involve human subjects, private personal data, scraped sensitive data, or high-risk generative models.

Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: Appendix M discusses broader impact, including potential benefits for rare-category recognition and more reliable imbalanced classification, as well as risks such as uneven performance across classes, deployment in high-stakes settings, dataset bias, and computational cost. The paper emphasizes that the method should be evaluated carefully before use in sensitive applications.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper does not release high-risk generative models, scraped datasets, or assets with substantial misuse risk. We release standard anonymized training and evaluation code for public supervised image-classification benchmarks, and we do not redistribute raw datasets.

Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We use public benchmark datasets and cite their original sources in the paper and references. We do not redistribute raw datasets; users are instructed to obtain them from the official sources and follow the corresponding licenses and terms of use. The released code package will include its own license and list required open-source dependencies.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: We introduce and release an anonymized implementation of TAILCON as supplementary material. The package includes README documentation, configuration files, training and evaluation scripts, and instructions for dataset preparation and reproduction. We do not introduce a new dataset, and we do not redistribute raw benchmark data.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing, user studies, or research with human subjects, so IRB approval or equivalent human-subjects review is not applicable.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method, experiments, analysis, and evaluation do not use LLMs as an important, original, or non-standard component. Any writing, editing, or formatting assistance does not affect the scientific claims, methodology, or experimental results.

    Guidelines:

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