TailAdapt: Heavy-Tailed Sparse Variational Adaptation for Long-Tailed Class Incremental Learning

Abhishek Kumar Sinha
Space Applications Centre
Indian Institute of Science
abhishek9@iisc.ac.in

Nitant Dube
Space Applications Centre
nitant@sac.isro.gov.in

Soma Biswas
Indian Institute of Science
somabiswas@iisc.ac.in

Abstract

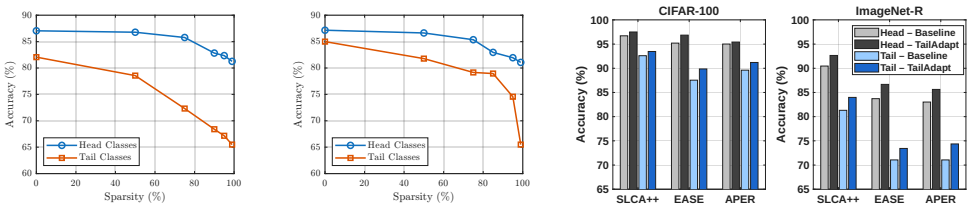
Long-Tailed Class-Incremental Learning (LT-CIL) is challenging due to severe class imbalance, where frequent head classes dominate gradient updates and suppress learning for rare tail classes. Beyond data imbalance, we identify asymmetric likelihood signals within each session, where underrepresented classes often provide weaker and more localized evidence for adapting previously learned representations. This asymmetry disrupts the stability-plasticity tradeoff, leading to pronounced forgetting and poor tail-class performance. We propose a hierarchical variational framework, TailAdapt, for prompt and adapter-based continual learning that enables adaptive group-wise regularization of model parameters. TailAdapt employs a hierarchical inverse Gaussian scale-mixture prior, which induces a heavy-tailed marginal Normal Inverse Gaussian (NIG) distribution over prompt and adapter parameters. This formulation encourages selective plasticity so that most parameter groups are strongly regularized to preserve previously learned knowledge, while a small, data-supported subset is allowed to adapt substantially. This selective plasticity allocates adaptation capacity according to the relevance of parameter groups to the current data, mitigating interference from head classes and improving learning for tail classes. Extensive experiments on LT-CIL benchmarks demonstrate improvements over strong baselines, with better tail-class performance, reduced forgetting, and more efficient utilization of prompt and adapter capacity. The code is available at https://github.com/darkbird1/TailAdapt.

1 Introduction

Large-scale foundation models for vision and vision-language understanding suffer from catastrophic forgetting when adapted to sequential data streams (Yan et al. [2023], McClelland et al. [1995]). In real-world continual learning (CL) (Li and Hoiem [2018], Shin et al. [2017], Lopez-Paz and Ranzato [2017], Zenke et al. [2017], Sinha et al. [2026]), this stability-plasticity tradeoff is further exacerbated by long-tailed data distributions. Under these imbalanced conditions, head classes can dominate the aggregate gradient updates, biasing shared parameters and causing disproportionate forgetting of minority tail classes. Existing methods, such as memory replay (Liu et al. [2022], Rebuffi et al. [2017]) and loss reweighting (Cui et al. [2019], Cao et al. [2019], Wang et al. [2021], He [2024]), largely treat parameter adaptation uniformly. These approaches scale classifier gradients without addressing the underlying representation dynamics, offering limited protection against tail-class forgetting while remaining vulnerable to noisy updates.

To diagnose this degradation, we analyze parameter sensitivity (via L_2 norm) within parameter-efficient tuning architectures (e.g., DualPrompt Wang et al. [2022]) under long-tailed class-incremental learning (LT-CIL). By masking prompt tokens based on increasing gradient sensitivity

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



(a) DualP (b) DualP + TailAdapt (c) Head & Tail Class Accuracies

Figure 1: Variation of average accuracy on CIFAR-100 in head and tail classes with fraction of masked prompts (sparsity) in **(a)** DualPrompt and **(b)** TailAdapt enabled DualPrompt. **(c)**: Head & Tail class accuracies for baselines with and without TailAdapt for CIFAR-100 and ImageNet-R.

(Fig. 1a), we uncover a striking asymmetry in parameter utilization: head classes distribute information redundantly and remain robust to partial parameter ablation, whereas tail classes rely on a fragile, highly sensitive subset of parameters. Because standard uniform updates blindly overwrite these critical weights, tail-class representations degrade rapidly. This highlights a fundamental flaw in applying fixed regularization across tasks and underscores the need for *selective plasticity*, a mechanism to isolate and protect task-relevant capacity.

Motivated by this insight, we propose **TailAdapt**, a hierarchical probabilistic framework for adaptive, group-wise parameter control. We partition learnable parameters into disjoint groups governed by latent scale variables with an Inverse Gaussian prior. This formulation allows the model to infer which parameter groups require flexibility for new data while strongly regularizing the remainder. By inducing sparse, data-driven updates, TailAdapt naturally concentrates adaptation on parameter groups most responsive to the current data, helping shield previously learned representations from interference. As shown in Fig. 1b, integrating TailAdapt preserves tail-class accuracy much more robustly under parameter sparsity while maintaining head-class stability.

Our main contributions are as follows:

- We identify uniform adaptation as a primary driver of tail-class forgetting in LT-CIL, demonstrating empirically that minority classes rely on localized, highly sensitive parameter groups that are easily overwritten by dominant head classes.
- We introduce TailAdapt, a variational framework that establishes a principled connection between selective plasticity and long-tailed continual learning through a group-wise semi-heavy-tailed Inverse Gaussian prior, enabling adaptive localized updates for task-relevant parameter groups while preserving previously learned representations.
- We demonstrate that TailAdapt is a lightweight, architecture-agnostic module seamlessly compatible with prompt-based tuning and low-rank adapters, requiring no modifications to the backbone or replay strategies. Extensive evaluations on LT-CIL benchmarks validate its effectiveness in mitigating catastrophic forgetting and optimizing adaptation capacity.

2 Related Work

Class-Incremental Learning: Class-incremental learning (CIL) seeks to learn new classes sequentially without forgetting prior knowledge (Wang et al. [2024]). Existing methods mainly rely on regularization, replay, or parameter-efficient fine-tuning (PEFT). Regularization approaches such as EWC (Kirkpatrick et al. [2017]) constrain updates to important parameters, but scale poorly with large pre-trained models and many tasks. Replay-based methods (Shin et al. [2017], Li et al. [2024]) alleviate forgetting through stored or generated samples, yet under long-tailed distributions replay buffers become dominated by head classes, further amplifying imbalance. *TailAdapt instead performs adaptive regularization by inferring per-group importance online through latent scale variables conditioned on current task data.* PEFT methods adapt frozen backbones using lightweight adapters (Zhou et al. [2024a], Sun et al. [2025], Zhou et al. [2024b], Zhang et al. [2023]) or prompts (Wang et al. [2022], Smith et al. [2023], Hong et al. [2024]). Although effective in balanced settings, they inherit biases from pre-trained representations, limiting adaptation for tail classes. Existing methods largely assume uniform parameter updates, which either destabilize head-class knowledge or under-adapt tail classes. *In contrast, TailAdapt partitions parameters into groups with independently*

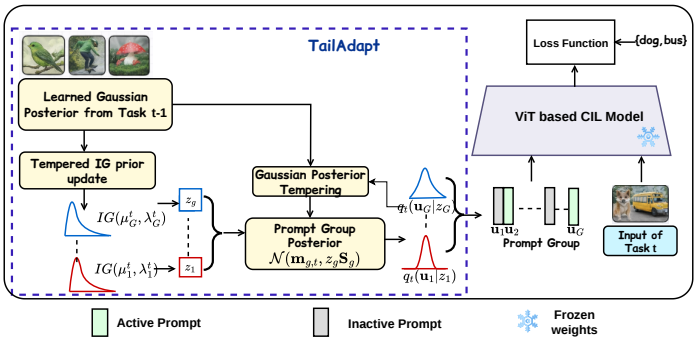


Figure 2: **Overview of TailAdapt for prompt-based continual learning**. The posterior from task $t-1$ is converted into a tempered Inverse Gaussian prior via moment matching. For task t, prompt groups use Gaussian posteriors with variances governed by group-wise Inverse Gaussian scales, enabling most groups to remain strongly regularized while a small subset adapts to new classes.

inferred latent scales, enabling selective adaptation while preserving prior knowledge.

Long-Tailed Class-Incremental Learning: Long-tailed recognition studies learning under severe class imbalance, where head classes dominate training while tail classes contain few samples. Standard empirical risk minimization biases models toward head classes, motivating resampling and reweighting methods (Chawla et al. [2002]), though these often overfit rare classes or under-utilize informative head samples.

Recent continual learning approaches address imbalance through classifier or data-level corrections. Liu et al. [2022] use learnable weight scaling to mitigate classifier bias, while PANDA (Raghavan et al. [2026]) transfers semantically rich patches from head to tail classes using a frozen CLIP encoder and distribution smoothing. However, these methods rely on classifier adjustment, augmentation, or strong CLIP-task alignment, leaving representation-level adaptation underexplored. In contrast, *TailAdapt directly adapts representations by selectively relaxing regularization for parameter groups most responsive to tail-class data, without requiring augmentation or classifier-level correction.*

Variational Continual Learning: Variational continual learning (VCL) applies online variational inference by treating the posterior after each task as the prior for the next (Nguyen et al. [2018]). Standard VCL assumes a fully factorized posterior, which neglects correlations between parameters and limits expressiveness. Subsequent work (Loo et al. [2021]) connects VCL with probabilistic regularization and shows how likelihood tempering bridges variational and Laplace-based approaches such as EWC. These studies highlight the potential of probabilistic modeling for continual learning, particularly in low-data regimes. Our work builds on this perspective by introducing hierarchical heavy-tailed priors to enable group-wise selective plasticity in long-tailed class-incremental learning. *TailAdapt addresses all these limitations simultaneously, by introducing hierarchical Inverse Gaussian scale-mixture prior to replace the Gaussian prior, and using tempered moment matching to prevent variance collapse.*

3 Problem Definition and Method Overview

Problem setting: In this work, we address long-tailed class incremental learning under a prompt tuning framework. Given a frozen pre-trained Vision Transformer encoder f_θ (e.g., CLIP ViT-B/16), the model is exposed to a sequence of tasks $\mathcal{T} = \{t_1, t_2, \ldots, t_T\}$, where each task t is associated with a labeled dataset $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^{N_t}$ consisting of input images x_i and corresponding labels y_i. The class distribution across tasks follows a long-tailed distribution, where head classes contain many training instances, whereas tail classes contain substantially fewer instances.

Overview: Fig. 2 illustrates how the proposed framework, TailAdapt augments a prompt-based continual learning pipeline. We assume a frozen Vision Transformer backbone and learn a small set of task-conditioned prompts. We partition the learnable parameters $\mathbf{U} = \{\mathbf{u}_g\}_{g=1}^{G}$ into G *prompt groups*, where each prompt token $\mathbf{u}_g$ forms one group. This grouping provides a structured

unit for controlling how much each subset of parameters is allowed to change across tasks (*see appendix for different grouping strategies*). After completing task $t-1$, the model maintains a variational posterior over all prompt groups that summarizes the knowledge accumulated so far. TailAdapt converts this posterior into a tempered hierarchical prior that serves as the starting point for the next task. For the current task t, each prompt group is associated with a latent scale variable that determines its adaptation capacity. During optimization, only those parameter groups that are already effective for the new task remain close to their prior values due to strong regularization, while groups that require task-specific adaptation receive larger inferred scales and undergo more substantial updates. The updated prompts are then combined with the frozen backbone to produce task predictions. This iterative posterior-to-prior propagation enables selective plasticity across tasks, allowing the model to preserve previously learned knowledge while allocating adaptation capacity to underrepresented or semantically novel classes.

4 Proposed TailAdapt Framework

We now describe the proposed TailAdapt framework and its key components. Our approach builds on hierarchical scale modeling to enable selective plasticity across parameter groups while preserving previously learned knowledge. We begin by introducing the hierarchical formulation that associates each parameter group with a latent scale variable modulating its adaptation capacity (Sec. 4.1). We then present the scale coupled variational posterior, which binds latent scales with group parameters to enable uncertainty-aware updates during optimization (Sec. 4.2). Next, we analyze how the interaction between the likelihood and KL regularization naturally induces sparsity and selective activation within the learning objective (Sec. 4.3). Finally, we describe the tempered prior update mechanism that propagates knowledge across tasks and supports continual learning without variance collapse (Sec. 4.4). Together, these components provide a unified probabilistic framework for allocating adaptation capacity across tasks in long-tailed continual learning.

4.1 Selective Plasticity via Hierarchical Scale Modeling

In long-tailed class-incremental learning, each training session provides access only to the current task dataset $\mathcal{D}_t$, which typically contains a highly imbalanced and limited number of samples for many classes. Moreover, tail classes are characterized by low sample counts, high intra-class variance, and poor alignment with representations learned from previous sessions. Consequently, gradient updates derived from such data are often insufficient to modify uniformly regularized models, leading to persistent underfitting and significant forgetting.

To enable adaptive learning under these constraints, we introduce a hierarchical prior that modulates parameter updates based on how well the accumulated prior knowledge from past sessions explains the current session data. The choice of scale prior is critical: it must simultaneously encourage strong shrinkage for most parameter groups to preserve previously learned knowledge, while allowing occasional large deviations to accommodate task-specific adaptation, particularly for underrepresented tail classes. Heavy-tailed Gaussian scale mixtures naturally satisfy these requirements by coupling parameter deviation with uncertainty. Among possible candidates, we adopt the Inverse Gaussian (IG) distribution because it exhibits a semi-heavy-tailed behavior that balances sparsity and numerical stability, while providing independent control over mean and precision through closed-form moments. For each parameter group g, we define,

$$z_g \sim \mathrm{IG}(\mu_g, \lambda_g), \quad \text{and } \boldsymbol{u}_g | z_g \sim \mathcal{N}(\boldsymbol{\mu}_{0,g}, z_g \mathbf{I}_d), \tag{1}$$

where z_g controls the effective learning capacity of group g and is sampled from long-tailed Inverse Gaussian distribution $\mathrm{IG}(\mu_g, \lambda_g)$. Unlike Fisher-based methods (Fisher [1925], Kirkpatrick et al. [2017]) that impose fixed importance derived from past tasks, TailAdapt identifies relevant parameters conditioned on current data via hierarchical latent scales. This adaptive, uncertainty-aware mechanism is more robust in long-tailed regimes, where Fisher estimate ignores the presence of tail-class samples in the current session.

During training on session t, the model observes only samples from $\mathcal{D}_t$, and the inferred latent scale z_g determines how strongly parameter group g should deviate from its prior to explain the current data. When the prior mean $\boldsymbol{\mu}_{0,g}$ already explains the current data well, the likelihood provides limited incentive for substantial deviation, keeping the inferred scale small and the corresponding regularization strong. Conversely, when the current data require substantial adaptation of a particular

parameter group, repeated optimization signals for that group favor larger inferred scales z_g, thereby relaxing regularization locally. Under long-tailed distributions, this mechanism is particularly useful for underrepresented classes, whose distinctive evidence may be concentrated in a small subset of parameter groups. For the parameter regimes used in TailAdapt, the IG scale prior assigns substantial probability to small positive scales while retaining a comparatively long right tail, enabling most groups to remain tightly regularized while permitting a small subset to expand when supported by the likelihood. Moreover, since the prior mean encodes parameters accumulated from previous tasks through tempered updates (Sec. 4.4), adaptation occurs relative to established representations, enabling localized updates for underrepresented classes while minimizing interference with prior knowledge.

4.2 Variational Coupling of Latent Scales and Parameter Groups

In LT-CIL, effective adaptation requires identifying a small subset of parameters that should remain plastic for the current session, while constraining the remaining parameters to preserve previously learned representations. This requires a posterior formulation that can simultaneously represent stable and adaptive groups and modulate their interaction during optimization. To achieve this, we approximate the posterior $p(\mathbf{U}, \mathbf{z}|\mathcal{D}_t)$ with a structured variational distribution that explicitly binds each parameter group to a learnable scale variable:

$$q(\mathbf{U}, \mathbf{z}) = \prod_{g=1}^{G} q(z_g) q(\boldsymbol{u}_g|z_g), \tag{2}$$

The latent scale z_g serves as a continuous indicator of group-level relevance to the current task. Parameter groups that consistently contribute to reducing the session loss are assigned larger posterior scales, whereas groups with limited contribution remain strongly regularized. This mechanism enables automatic allocation of adaptation capacity based on observed optimization dynamics, rather than relying on predefined class information.

Reparameterization Trick: Unlike Gaussian distribution in VAE (Kingma and Welling [2014]), Inverse Gaussian does not have a closed form expression for reparameterization trick. For computational efficiency and stable training, we parameterize $q(z_g)$ using a log-normal surrogate distribution (Aitchison and Brown [1957]) with learnable parameters $\{a_g, b_g\}$ as,

$$\log z_g = a_g + b_g \epsilon_g, \;\; \epsilon_g \sim \mathcal{N}(0, 1), \tag{3}$$

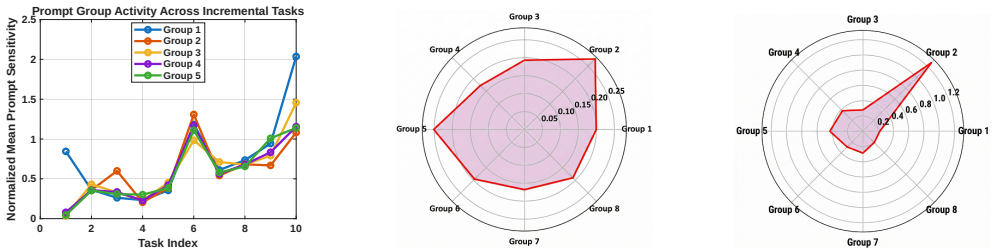
which enables low-variance gradient estimation through reparameterization. This choice retains the essential heavy-tailed behavior required for selective activation while avoiding the numerical instability associated with directly optimizing Inverse Gaussian posteriors. A theoretical analysis (*see appendix*) shows that distribution-mismatch bias primarily modulates the strength of selective plasticity without compromising learning or knowledge stability.

Conditioned on z_g, the posterior over each parameter group is modeled as the Gaussian distribution with learnable mean and scaled covariance matrix,

$$q(\mathbf{u}_g|z_g) = \mathcal{N}(\mathbf{m}_g, z_g \mathbf{S}_g), \text{ where } \mathbf{S}_g = \text{diag}(s_1, \dots s_d) \in \mathbb{R}^{d \times d} \tag{4}$$

The posterior mean $\mathbf{m}_g$ represents the adapted parameter value for group g, while the latent scale z_g controls the degree to which this group can deviate from its prior. Groups with small z_g yield concentrated posteriors that remain close to their prior means, whereas larger scales allow controlled deviations to incorporate new information. This coupling between group means and latent scales enables adaptive allocation of plasticity, restricting substantial updates to a small subset of parameters while anchoring the remainder to previously learned representations, promoting long-term stability.

Extension to Adapter based methods: Although we present TailAdapt in the context of prompt-based tuning, the proposed framework naturally extends to adapter based architectures. In this setting, parameter groups correspond to adapter modules with the learnable parameter of a given adapter belonging to a single group, and the hierarchical latent scales regulate their adaptation in the same manner.



(a) Sessionwise group sensitivity (b) Group sensitivity for head class (c) Group sensitivity for tail class

Figure 3: (**a**) Average group sensitivity across incremental sessions in ordered LT-CIL using CIFAR-100, and sensitivity patterns for **(b)** head classes and **(c)** tail classes shown separately.

4.3 Stability-Plasticity Tradeoff in Variational Objective

TailAdapt's adaptive behavior is characterized by the variational objective in Eq. 5. For session t, we optimize the evidence lower bound (ELBO),

$$\mathcal{L}_t = \mathbb{E}_{q(\mathbf{U},\mathbf{z})}\big[\log p(\mathcal{D}_t|\mathbf{U})\big] - \beta_{\mathrm{KL}} \cdot \mathrm{KL}\big(q_t(\mathbf{U},\mathbf{z}) \,\|\, p_t(\mathbf{U},\mathbf{z})\big), \tag{5}$$

where the prior p_t encodes knowledge accumulated from previous sessions through tempered updates, as described in section 4.4. Due to the hierarchical factorization of the posterior and prior, the KL term decomposes across groups:

$$\mathrm{KL}\big(q_t(\mathbf{U},\mathbf{z}) \,\|\, p_t(\mathbf{U},\mathbf{z})\big) = \sum_{g=1}^{G} \Big[\mathbb{E}_{q_t(\mathbf{z}_g)}\big[\mathrm{KL}\big(q_t(\mathbf{u}_g \mid z_g) \,\|\, p_t(\mathbf{u}_g \mid z_g)\big)\big] + \mathrm{KL}\big(q_t(z_g) \,\|\, p_t(z_g)\big)\Big] \tag{6}$$

The KL divergence between the conditional Gaussians simplifies due to the shared scale z_g, and after taking expectations on both sides, we get,

$$\begin{aligned}\mathbb{E}_{q_t(\mathbf{z}_g)}[\mathrm{KL}\big(q_t(\mathbf{u}_g \mid z_g) \,\|\, p_t(\mathbf{u}_g \mid z_g)\big)] = \frac{1}{2}\Big[&\mathrm{tr}(\mathbf{S}_g) - d - \log|\mathbf{S}_g| \\ &+ \mathbb{E}_{q_t}[z_g^{-1}]\Big[(\mathbf{m}_g^{(t)} - \boldsymbol{\mu}_{0,g}^{(t)})^\top (\mathbf{m}_g^{(t)} - \boldsymbol{\mu}_{0,g}^{(t)})\Big].\end{aligned} \tag{7}$$

Since z_g is sampled from the log-normal distribution, $\mathbb{E}_q[z_g^{-1}]$ (Aitchison and Brown [1957]) is given by, $\mathbb{E}_q\big[z_g^{-1}\big] = \exp\big(-a_g + \frac{1}{2}b_g^2\big)$. The last equation shows that the deviation of the posterior mean from the prior mean is weighted by the inverse latent scale. Thus, the latent scale acts as an adaptive precision term that modulates the effective regularization strength for each parameter group. Small inferred scales yield large precision, tightly anchoring parameters to previously learned representations, whereas larger scales relax this constraint and permit greater data-driven adaptation.

This mechanism naturally aligns with learning under long-tailed distributions. When current head-class data are well explained by the accumulated prior, their optimization signals need not induce substantial deviations from prior parameters, resulting in relatively small inferred scales. Conversely, when underrepresented classes exhibit distinctive evidence that is not adequately captured by the prior, the corresponding parameter groups may require larger deviations, increasing their inferred scales and weakening effective regularization. Consequently, TailAdapt allocates plasticity selectively to parameter groups most relevant for underrepresented classes while preserving stable head-class knowledge.

We empirically validate this behavior by measuring the average prompt-group sensitivity across incremental sessions in ordered LT-CIL, where head classes precede tail classes (Fig. 3a). Early sessions activate diverse prompt groups to capture varied semantic features, followed by a coordinated increase in mid-sessions as semantically similar classes accumulate. In contrast, later sessions dominated by tail classes exhibit selective reliance on a subset of groups rather than uniformly utilizing all groups. Furthermore, Figs. 3b and 3c show that head-class sessions leverage multiple groups, whereas tail-dominated sessions exhibit more concentrated reliance on a subset of groups, enabling representation learning while preserving stability.

Model	CIFAR-100				ImageNet-R				CUB-200			
	10 Tasks		5 Tasks		10 Tasks		5 Tasks		10 Tasks		5 Tasks	
	$\overline{Acc}\uparrow$	$\mathcal{F}\downarrow$	$\overline{Acc}\uparrow$	$\mathcal{F}\downarrow$	$\overline{Acc}\uparrow$	$\mathcal{F}\downarrow$	$\overline{Acc}\uparrow$	$\mathcal{F}\downarrow$	$\overline{Acc}\uparrow$	$\mathcal{F}\downarrow$	$\overline{Acc}\uparrow$	$\mathcal{F}\downarrow$
SLCA++	91.73	6.73	92.06	5.22	68.07	6.96	74.26	6.38	83.65	12.26	85.12	9.26
SLCA++ +PANDA	92.05	6.23	93.10	4.87	81.48	5.48	83.90	5.67	88.34	11.52	91.30	8.57
SLCA++ +TailAdapt	**93.58**	**4.56**	**94.28**	**4.33**	**82.49**	**4.69**	**84.95**	**5.10**	**90.99**	**10.82**	**92.47**	**7.43**
EASE [CVPR 24]	89.94	4.69	92.28	5.75	67.25	10.26	72.84	7.79	86.81	12.14	85.62	11.86
EASE+PANDA	91.95	6.65	93.15	5.47	71.68	7.82	73.75	7.46	87.56	11.47	88.17	12.92
EASE+TailAdapt	**92.52**	6.35	**95.27**	**5.25**	**74.84**	**6.89**	**75.92**	**6.36**	**88.57**	**10.49**	**88.36**	**9.36**
MOS [AAAI 25]	91.60	4.69	92.18	5.21	76.21	9.16	76.86	7.69	89.27	9.26	90.50	10.20
MOS+Panda	92.04	**4.48**	93.09	4.76	77.94	**8.26**	78.16	7.56	90.79	7.86	90.92	10.06
MOS+TailAdapt	**92.74**	4.86	**93.76**	**4.36**	**79.61**	8.32	**79.85**	**7.09**	**92.79**	**5.24**	**91.92**	**8.76**
APER-Adapter [IJCV 24]	87.66	5.66	88.11	5.17	78.04	4.86	78.75	4.78	87.21	8.92	87.35	7.65
APER-Adapter+PANDA	88.94	5.26	89.26	4.98	78.76	4.87	79.49	4.69	89.63	**5.11**	89.02	5.10
APER-Adapter+TailAdapt	**90.56**	**4.96**	**91.11**	**4.21**	**80.97**	**4.78**	**80.96**	**4.33**	**91.12**	5.42	**90.46**	**5.02**
DualP [ECCV 22]	74.24	8.14	78.12	5.32	70.24	8.29	68.49	7.58	75.69	14.21	78.36	12.63
DualP+PANDA	81.05	7.38	85.82	4.92	70.83	5.39	71.42	4.59	77.75	12.36	81.35	11.96
DualP+TailAdapt	**85.91**	**6.59**	**86.48**	**4.43**	**72.26**	**5.06**	**73.18**	**3.97**	**80.23**	**11.49**	**83.24**	**8.73**
CodaP [CVPR 23]	76.55	7.55	78.02	6.89	77.45	6.02	77.85	6.06	74.23	13.21	75.64	8.75
CodaP+PANDA	87.42	**4.61**	88.64	6.47	77.95	5.35	78.29	4.62	80.53	11.23	82.96	**6.08**
CodaP+TailAdapt	**88.54**	5.33	**89.27**	6.41	**78.85**	**5.41**	**79.12**	**4.55**	**87.50**	**10.56**	**89.69**	6.91
DAP [IJCAI 24]	60.12	15.21	56.67	13.50	47.32	13.26	44.65	11.94	47.95	12.79	46.83	10.69
DAP+PANDA	62.55	12.26	58.02	12.16	**49.20**	12.65	45.36	10.26	48.12	12.69	47.60	9.76
DAP+TailAdapt	**63.24**	**9.28**	**58.95**	**9.22**	47.06	**9.36**	**46.25**	**9.12**	**51.92**	**8.50**	**50.12**	**8.97**

Table 1: Average top-1 accuracy and average forgetting for several baselines and comparison with PANDA [AAAI 2026] for shuffled long-tailed scenarios. **Bold**: best results.

4.4 Tempered Prior Propagation for Stability

A central challenge in continual learning is maintaining stability across tasks while retaining sufficient plasticity to incorporate new knowledge. Directly propagating the posterior from one session as the prior for the next, as used in Loo et al. [2021] and Nguyen et al. [2018], can lead to overconfident parameter estimates and progressive variance collapse, which restricts future adaptation and increases susceptibility to catastrophic forgetting. Following Mandt et al. [2016] to address this issue, TailAdapt employs a tempered prior propagation mechanism that explicitly controls how much information from previous sessions influences subsequent learning. After completing task $t-1$, we summarize the learned posterior over latent scales and parameter groups through moment matching. Specifically, for the log-normal posterior $q^{(t-1)}(z_g)$, we compute its mean and variance as,

$$\hat{\mu}_g = \mathbb{E}_{q^{(t-1)}}[z_g] = \exp\Big(a_g^{(t-1)} + \tfrac{1}{2}\big(b_g^{(t-1)}\big)^2\Big), \tag{8}$$

$$\hat{\sigma}_g^2 = \mathrm{Var}_{q^{(t-1)}}[z_g] = \Big(\exp\big((b_g^{(t-1)})^2\big) - 1\Big)\exp\Big(2a_g^{(t-1)} + (b_g^{(t-1)})^2\Big). \tag{9}$$

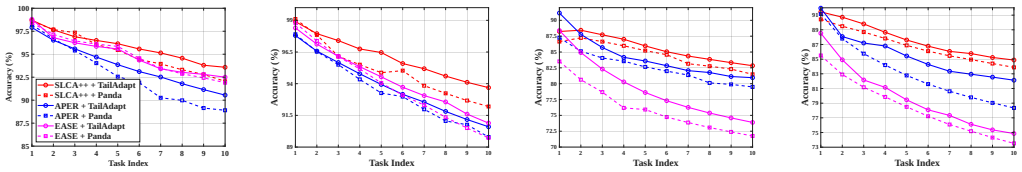
We then fit an Inverse-Gaussian IG$(\tilde{\mu}, \tilde{\lambda})$ with matching moments as, $\tilde{\mu} = \hat{\mu}_g$ and $\tilde{\lambda} = \frac{\hat{\mu}_g^3}{\hat{\sigma}_g^2}$. To control how much newly learned information is incorporated into the prior for subsequent tasks, we construct a tempered hierarchical prior via linear interpolation with rate γ, given by,

$$\mu_g^{(t)} = (1-\gamma)\,\mu_g^{(t-1)} + \gamma\tilde{\mu}^{(t-1)}, \qquad \lambda_g^{(t)} = (1-\gamma)\,\lambda_g^{(t-1)} + \gamma\,\tilde{\lambda}^{(t-1)},$$
$$\boldsymbol{\mu}_{0,g}^{(t)} = (1-\gamma)\,\boldsymbol{\mu}_{0,g}^{(t-1)} + \gamma\,\mathbf{m}_g^{(t-1)}$$

This tempering mechanism provides explicit control over the stability–plasticity tradeoff across sessions. Smaller values of γ emphasize previously accumulated knowledge, yielding stronger regularization and improved stability, whereas larger values increase responsiveness to newly observed data, enabling faster adaptation. Importantly, the interpolation prevents abrupt shifts in the prior distribution, which stabilizes optimization and reduces the risk of catastrophic forgetting caused by sudden parameter drift.

5 Experiments and Results

Datasets and Evaluation Protocol. We evaluate on CIFAR-100 (Krizhevsky [2009]), ImageNet-R (Hendrycks et al. [2021]), and CUB-200 (Wah et al. [2011]) under 5-task and 10-task shuffled and ordered long-tailed class-incremental learning (LT-CIL) settings. In the shuffled setting, head and



(a) CIFAR-100 $\rho = 0.01$ (b) CIFAR-100 $\rho = 0.1$ (c) INet-R $\rho = 0.01$ (d) INet-R $\rho = 0.1$

Figure 4: Sessionwise accuracy of CIFAR-100-LT and ImageNet-R-LT for decay rate (ρ) set to 0.01 and 0.1.

tail classes are randomly distributed across sessions, whereas in the ordered setting, head classes precede tail classes across sessions. For fair comparison, we use the same class order and long-tailed distribution with exponential decay rate $\rho = 0.01$.

TailAdapt is compatible with both adapter- and prompt-based tuning. For adapters, we evaluate SLCA++ (Zhang et al. [2023, 2024]), EASE (Zhou et al. [2024a]), MOS (Sun et al. [2025]), and APER (Zhou et al. [2024b]). For prompt tuning, we include DAP (Hong et al. [2024]), CodaPrompt (Smith et al. [2023]), and DualPrompt (Wang et al. [2022]). We also compare with PANDA (Raghavan et al. [2026]), a state-of-the-art augmentation framework, applied to the above methods.

We report (i) *Average Accuracy:* $\overline{Acc} = \frac{1}{T}\sum_{t=1}^{T} Acc_t$ where Acc_t denotes top-1 accuracy of task t. (ii) *Forgetting Rate:* $\mathcal{F} = \frac{1}{T-1}\sum_{t=1}^{T-1}(\max_{\tau<T} \text{Acc}_t^{\tau} - \text{Acc}_t^{T})$ as evaluation metrics, where Acc_t^{τ} is the accuracy on task t after learning task τ. We additionally measure effective sparsity through the average inverse latent scale: $\mathcal{S} = \frac{1}{G}\sum_{g=1}^{G} \mathbb{E}[z_g^{-1}]$.

Implementation Details. Each prompt token or adapter is treated as a parameter group. We train using Adam for 15 epochs with learning rate 0.015. The KL coefficient β_{KL} is linearly increased from 0 to 1 during the first 10 epochs for stable optimization. All other hyperparameters, including the number of adapters and prompt tokens, are kept identical to the original baselines for fair comparison.

5.1 Experimental Results

Tables 1 and 2 show that TailAdapt generally improves performance over the evaluated baselines across datasets and task settings in both shuffled and ordered LT-CIL. On CIFAR-100, it provides clear gains for representative adapter-based methods such as SLCA++, EASE, and APER, demonstrating robustness under long-tailed class-incremental settings. Prompt-based methods benefit even more, with DualPrompt and CodaPrompt achieving substantial improvements, supporting the effectiveness of group-wise adaptive regularization in mitigating degradation associated with uniform parameter updates. Similar trends are observed on ImageNet-R and CUB-200, where TailAdapt mostly enhances accuracy, confirming its generalizability.

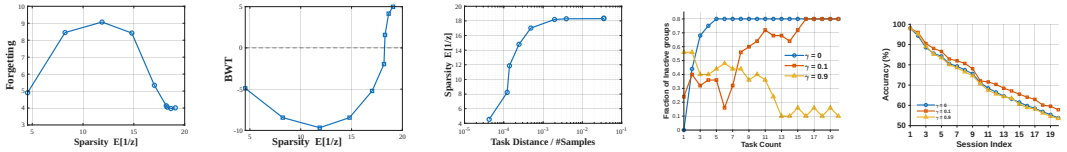
Fig. 4 further presents session-wise average accuracy under varying imbalance ratios. TailAdapt maintains stable performance across intermediate sessions and reduces the degradation observed in later sessions as the class distribution becomes increasingly tail-dominated. Overall, TailAdapt achieves higher accuracy and lower forgetting across strong baselines and datasets, validating its effectiveness. *For theoretical results and related empirical validation, and detailed studies on grouping strategies, see the Appendix.*

5.2 Additional Analyses

(i) Mechanistic Interpretability via Sparsity-Forgetting Correlation: To understand the role of sparsity, we use an Ordered CIFAR-100-LT-CIL setup where head classes appear in early sessions followed by tail classes. We analyze the relationship between prompt activation sparsity $\mathbb{E}[z_g^{-1}]$, forgetting, and backward transfer (BWT) across tasks. Fig. 5a shows an inverted U-shaped relationship between sparsity and forgetting: both dense and overly sparse configurations yield high forgetting, while intermediate sparsity minimizes it. This reflects a fundamental trade-off: insufficient sparsity causes parameter interference across tasks, whereas excessive sparsity limits capacity for new concepts, forcing reuse of existing groups and overwriting prior knowledge.

Model	CIFAR-100				ImageNet-R				CUB-200			
	10 Tasks		5 Tasks		10 Tasks		5 Tasks		10 Tasks		5 Tasks	
	$\overline{Acc}\uparrow$	$\mathcal{F}\downarrow$	$\overline{Acc}\uparrow$	$\mathcal{F}\downarrow$	$\overline{Acc}\uparrow$	$\mathcal{F}\downarrow$	$\overline{Acc}\uparrow$	$\mathcal{F}\downarrow$	$\overline{Acc}\uparrow$	$\mathcal{F}\downarrow$	$\overline{Acc}\uparrow$	$\mathcal{F}\downarrow$
SLCA++	90.06	6.06	90.06	6.26	70.32	7.30	74.98	5.96	84.96	11.33	86.07	8.72
SLCA++ +PANDA	90.69	5.69	90.81	**4.17**	80.15	4.89	81.49	5.29	89.69	10.98	92.36	9.01
SLCA++ +TailAdapt	**92.60**	**5.03**	**92.42**	4.31	**83.81**	**4.66**	**84.69**	**4.97**	**90.30**	**9.14**	**93.19**	**7.98**
EASE [CVPR 24]	91.02	**4.03**	93.62	5.18	69.02	9.36	74.36	7.39	88.60	11.98	86.99	10.63
EASE+PANDA	92.12	6.90	**94.13**	5.45	71.46	6.96	72.23	7.02	88.23	10.26	88.39	11.36
EASE+TailAdapt	**93.03**	6.45	92.77	**4.06**	**74.56**	**5.90**	**74.02**	**6.36**	**90.17**	**9.14**	**89.03**	**10.05**
MOS [AAAI 25]	91.36	8.02	93.66	6.36	78.02	10.19	77.98	7.64	90.23	9.36	91.69	11.30
MOS+PANDA	92.65	5.15	94.39	**5.22**	79.02	10.03	80.00	9.64	92.36	8.01	91.89	10.42
MOS+TailAdapt	**93.41**	**3.96**	**95.49**	5.74	**81.36**	**8.99**	**82.12**	**8.74**	**93.80**	**5.55**	**92.65**	**9.23**
APER [IJCV 24]	88.11	**4.84**	85.32	5.23	76.00	**3.59**	76.60	3.90	85.96	7.95	86.02	7.30
APER+PANDA	88.61	4.86	86.45	**5.21**	77.69	3.62	77.36	4.06	86.98	7.09	87.04	6.62
APER+TailAdapt	**90.23**	4.29	**90.50**	5.69	**80.48**	3.60	**80.91**	**3.76**	**88.15**	**6.65**	**88.99**	**5.91**
DualP [ECCV 22]	76.36	9.36	77.01	8.24	70.16	7.96	69.01	7.41	76.39	13.02	78.09	11.77
DualP+PANDA	86.09	8.94	87.14	7.87	71.11	8.23	71.26	5.96	78.54	11.96	82.01	10.99
DualP+TailAdapt	**90.67**	**8.69**	**90.09**	**6.55**	**73.11**	**5.31**	**74.55**	**4.65**	**81.02**	**10.45**	**83.69**	**9.37**
CodaP [CVPR 23]	78.01	7.65	79.44	6.96	79.14	5.96	80.96	5.96	75.02	12.54	76.09	8.36
CodaP+PANDA	85.36	5.92	84.96	6.02	80.98	5.65	82.06	5.88	79.36	11.02	80.33	8.25
CodaP+TailAdapt	**88.23**	**5.88**	**87.09**	**5.77**	**82.60**	**5.39**	**84.36**	**4.90**	**81.88**	**10.65**	**83.01**	**8.11**
DAP [IJCAI 24]	62.02	12.69	59.01	10.08	49.36	13.01	45.65	10.59	48.99	11.09	47.98	10.26
DAP+PANDA	63.69	**11.75**	60.67	9.60	51.06	12.65	46.36	9.65	50.02	10.56	49.26	9.89
DAP+TailAdapt	**64.52**	11.87	**61.98**	**9.55**	**52.23**	**11.96**	**47.96**	**8.69**	**51.77**	**10.51**	**52.10**	**8.99**

Table 2: Average top-1 accuracy and average forgetting for several baselines and comparison with PANDA [AAAI 2026] for ordered long-tailed scenarios. **Bold**: best results.



(a) Forgetting vs sparsity (b) BWT vs sparsity (c) Task similarity vs.sparsity (d) Inactive groups vs. session (e) Accuracy vs. session

Figure 5: **Mechanistic interpretability of sparsity mechanism in TailAdapt.**(a) Variation of $\mathcal{F}$ with sparsity (b) Variation of BWT with sparsity (c) Variation of prompt activity (sparsity) with task similarity normalized by number of samples. (d) Variation of inactive groups with sessions in ordered LT-CIL (e) Accuracy vs session for different values of γ.

Fig. 5b presents a complementary view through BWT (Lin et al. [2022]), which also exhibits a U-shaped trend. While forgetting measures the magnitude of performance drop from the peak accuracy, BWT captures the direction of knowledge transfer from new tasks to previously learned tasks and therefore offers additional insight into inter-task interference. In the dense regime ($\mathcal{S} < 8$), forgetting is high but BWT is less negative because updates affect all tasks uniformly, producing an averaged representation. In the optimal regime ($8 < \mathcal{S} < 12$), overall forgetting is lowest, but BWT reaches its minimum due to task-specific specialization that induces asymmetric forgetting. In the over-sparse regime ($\mathcal{S} > 12$), most parameters remain frozen, preserving past knowledge and resulting in higher BWT. Fig. 5c further shows that TailAdapt automatically calibrates sparsity based on task semantic similarity. Similar tasks (low CLIP distance) lead to lower sparsity to encourage feature sharing, whereas dissimilar tasks trigger higher sparsity to enforce prompt orthogonality and reduce interference. This adaptive allocation emerges naturally from the IG prior without requiring manual hyperparameter tuning.

(ii) Mitigating Variance Collapse via Tempered Matching: A key challenge in Bayesian continual learning is variance collapse: as posteriors are propagated as priors across tasks, epistemic uncertainty shrinks, limiting adaptation to new data. We address this using tempered moment matching with parameter $\gamma \in (0, 1]$, which acts as an implicit forgetting rate. Since the prior is updated recursively through interpolation, the contribution of task s to the model at task $t > s$ scales approximately as $(1-\gamma)^{t-s}$. This controlled decay regulates the influence of past knowledge while preserving useful information, maintaining calibrated uncertainty and supporting stable adaptation to future tasks.

Fig. 5d shows how γ affects the fraction of inactive prompt groups over 20 tasks on CIFAR-100. The trajectory reveals a trade-off: aggressive consolidation ($\gamma = 0$) overconfidently preserves the prior, causing premature capacity exhaustion, whereas large γ overemphasizes current data, increasing parameter activity and inducing forgetting. Empirically, balanced behavior occurs for $\gamma \in [0.05, 0.2]$. Fig. 5e provides a complementary view through session-wise accuracy. Performance drops for $\gamma = 0$ due to insufficient adaptation, while large γ (e.g., 0.9) causes excessive updates and catastrophic forgetting, again reducing accuracy.

6 Conclusion and Limitations

We studied long-tailed class-incremental learning and identified uniform parameter adaptation as a potential cause of tail-class forgetting in parameter-efficient fine-tuning. We proposed TailAdapt, a hierarchical variational framework with Inverse Gaussian priors that enables group-wise selective plasticity, allocating capacity adaptively to preserve head representations while allowing targeted tail updates. However, TailAdapt assumes reasonably informative prior representations and may be less effective under severe domain shifts. Moreover, current version of TailAdapt utilizes handcrafted groups of learnable parameters,which is another possible direction to extend the learning capacity via adaptive grouping, and is not compatible with training free methods.

References

Jiexi Yan, Zhihui Yin, Erkun Yang, Yanhua Yang, and Heng Huang. Learning with diversity: Self-expanded equalization for better generalized deep metric learning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19308–19317, 2023.

James L. McClelland, Bruce L. McNaughton, and Randall C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419–457, 1995.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, 2017.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, page 3987–3995. PMLR, 2017.

Abhishek Kumar Sinha, Nitant Dube, and Soma Biswas. Quantized residuals to continuous prompts for few-shot class incremental learning in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Xialei Liu, Yu-Song Hu, Xu-Sheng Cao, Andrew D Bagdanov, Ke Li, and Ming-Ming Cheng. Long-tailed class incremental learning. In *European Conference on Computer Vision (ECCV)*, pages 495–512. Springer, 2022.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

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 (CVPR)*, 2019.

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

Jiaqi Wang, Wenwei Zhang, Yuhang Zang, Yuhang Cao, Jiangmiao Pang, Tao Gong, Kai Chen, Ziwei Liu, Chen Change Loy, and Dahua Lin. Seesaw loss for long-tailed instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

Jiangpeng He. Gradient reweighting: Towards imbalanced class-incremental learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16668–16677, June 2024.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Dualprompt: Complementary prompting for rehearsal-free continual learning. *European Conference on Computer Vision*, 2022.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, March 2017. ISSN 1091-6490.

Yichen Li, Qunwei Li, Haozhao Wang, Ruixuan Li, Wenliang Zhong, and Guannan Zhang. Towards efficient replay in federated incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12820–12829, June 2024.

Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *CVPR*, pages 23554–23564. IEEE, 2024a.

Hai-Long Sun, Da-Wei Zhou, Hanbin Zhao, Le Gan, De-Chuan Zhan, and Han-Jia Ye. Mos: Model surgery for pre-trained model-based class-incremental learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(19):20699–20707, Apr. 2025.

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *International Journal of Computer Vision*, 2024b.

Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11909–11919, June 2023.

Chenxing Hong, Yan Jin, Zhiqi Kang, Yizhou Chen, Mengke Li, Yang Lu, and Hanzi Wang. Dynamically anchored prompting for task-imbalanced continual learning. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI*, 2024.

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.

Siddeshwar Raghavan, Jiangpeng He, and Fengqing Zhu. Panda - patch and distribution-aware augmentation for long-tailed exemplar-free continual learning. In *Proceedings of the Annual AAAI Conference on Artificial Intelligence*, 2026.

Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner. Variational continual learning. In *International Conference on Learning Representations*, 2018.

Noel Loo, Siddharth Swaroop, and Richard E Turner. Generalized variational continual learning. In *International Conference on Learning Representations*, 2021.

R. A. Fisher. Theory of statistical estimation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 22(5):700–725, 1925.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR*, 2014.

John Aitchison and James Alan Calvert Brown. *The lognormal Distribution*, volume 5. Cambridge University Press, Cambridge, 1957.

Stephan Mandt, James McInerney, Farhan Abrol, Rajesh Ranganath, and David Blei. Variational tempering. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 704–712. PMLR, 09–11 May 2016.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Jul 2011.

Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. Slca++: Unleash the power of sequential fine-tuning for continual learning with pre-training, 2024. URL `https://arxiv.org/abs/2408.08295`.

Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. Beyond not-forgetting: continual learning with backward knowledge transfer. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022.

Raj S Chhikara and J Leroy Folks. *The Inverse Gaussian Distribution: Theory, Methodology, and Applications*. CRC Press, New York, 1988.

A Effective Curvature and Local Analysis

For a fixed task t, the TailAdapt ELBO from Eq. 5 can be written (ignoring constants) as,

$$\mathcal{L}(\{m_g\},\{z_g\}) = \ell(\mathbf{m}) - \beta_{\mathrm{KL}} \sum_{g=1}^{G} \alpha_g \|\mathbf{m}_g - \boldsymbol{\mu}_{0,g}\|^2 - \beta_{\mathrm{KL}} \sum_{g=1}^{G} \mathrm{KL}\big(q(z_g) \,\|\, p(z_g)\big), \tag{10}$$

where $\alpha_g := \mathbb{E}_q[z_g^{-1}]$. We analyze optimization near a stationary point $\mathbf{m}^\star$ under a standard local quadratic approximation. In a neighborhood of $\mathbf{m}^\star$, $\ell(\mathbf{m}) \approx \ell(\mathbf{m}^\star) - \frac{1}{2}\sum_{g=1}^{G}(\mathbf{m}_g - \mathbf{m}_g^\star)^\top \mathbf{H}_g (\mathbf{m}_g - \mathbf{m}_g^\star)$, where $L_g \mathbf{I} \succeq \mathbf{H}_g \succeq 0$. Substituting the approximation in the ELBO, we get effective curvature matrix as,

$$\tilde{\mathbf{H}}_g = \mathbf{H}_g + \beta_{\mathrm{KL}} \alpha_g \mathbf{I}. \tag{11}$$

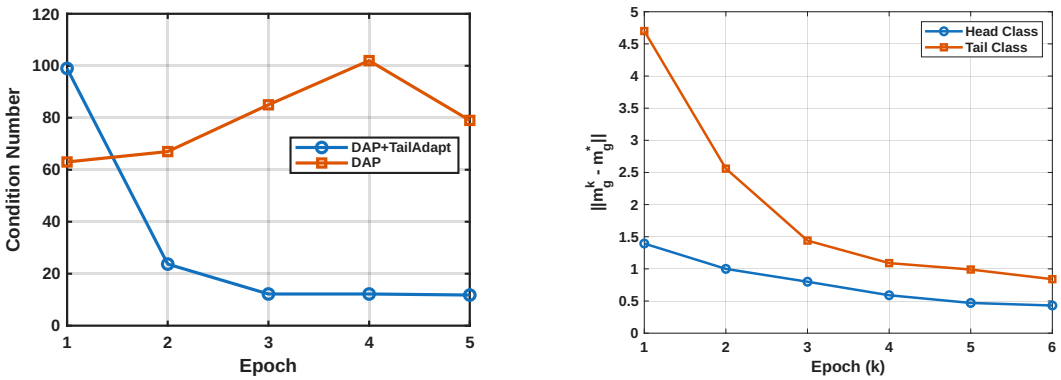
Thus, learning latent scales z_g is equivalent to learning additive curvature corrections $\beta_{KL}\alpha_g \mathbf{I}$ for each group.

(ii) Scale-Mean Coupling Under IG Prior: Under the IG prior used by TailAdapt, stationary points exhibit a characteristic coupling between learned scales and parameter deviations.

Proposition 1 (Local stationary coupling under quadratic approximation)**.** *Under a local quadratic approximation of the task loss and for an active group g with $\mathbf{m}_g \neq \boldsymbol{\mu}_{0,g}$ at stationarity, and in the large-adaptation regime such that $\left\|\mathbf{m}_g^\star - \boldsymbol{\mu}_{0,g}\right\| \gg \sqrt{\lambda_g}$,*

$$\alpha_g^\star = \mathbb{E}\big[z_g^{-1}\big] = \frac{c_1}{\left\|\mathbf{m}_g^\star - \boldsymbol{\mu}_{0,g}\right\|}, \;\; c_1 \geq 0 \tag{12}$$

This implies that groups undergoing large adaptations are automatically assigned smaller curvature penalties, while inactive or minimally adapted groups receive strong regularization.



(a) Condition number vs epoch

(b) Error vs. epoch

Figure 6: Experimental observations for Theorems 1–2 using CIFAR-100 in DAP.

B Theoretical Analysis

We provide theoretical insights into why TailAdapt improves optimization stability and mitigates forgetting in long-tailed continual learning. We show that the proposed hierarchical scale modeling induces an adaptive curvature regularization that (i) alleviates ill-conditioning in low-data regimes typical of tail classes and (ii) promotes efficient convergence for parameter groups requiring adaptation. Together, these properties explain how TailAdapt stabilizes training while enabling selective plasticity across tasks. We begin by analyzing the conditioning of the optimization problem under the proposed objective.

Theorem 1 (Condition-number bound)**.** *Consider the TailAdapt objective with respect to the posterior mean m_g. The second-order curvature of the objective can be written as*

$$\tilde{\mathbf{H}}_g = \mathbf{H}_g + \beta_{\mathrm{KL}}\alpha_g \mathbf{I}, \tag{13}$$

where $\mathbf{H}_g$ denotes the Hessian of the data loss and $\alpha_g = \mathbb{E}[z_g^{-1}]$ depends on the inferred latent scale. For each active group g with $\mathbf{m}_g \neq \boldsymbol{\mu}_{0,g}$, the condition number satisfies

$$\kappa(\tilde{\mathbf{H}}_g) \leq 1 + \frac{L_g}{\beta_{\mathrm{KL}}\alpha_g} = 1 + \frac{L_g\|\boldsymbol{m}_g - \boldsymbol{\mu}_{0,g}\|}{\beta_{\mathrm{KL}}c_1}, \tag{14}$$

where L_g is the Hessian bound, and $c_1 = e^{\frac{1}{2}b_g^2}\sqrt{\frac{\lambda_g}{(\mu_g)^2}}$.

Theorem 1 shows that TailAdapt introduces an adaptive curvature term $\beta_{\mathrm{KL}}\alpha_g I$ that increases the smallest eigenvalue of the objective, preventing near-singular directions even when the data Hessian $\mathbf{H}_g$ is poorly conditioned. This stabilizes optimization particularly in low-data regimes such as tail classes. We next analyze the convergence behavior of the optimization dynamics under the adaptive regularization.

Theorem 2 (Linear convergence of active groups)**.** *Consider gradient ascent on the TailAdapt objective with step size η satisfying $0 < \eta \leq \frac{1}{L_g+\beta_{KL}\alpha_g^\star}$. The stationary point of the resulting regularized local objective is $\widetilde{\mathbf{m}}_g^\star = \tilde{\mathbf{H}}_g^{-1}\left(\mathbf{H}_g\mathbf{m}_g^\star + \beta_{\mathrm{KL}}\alpha_g^\star\boldsymbol{\mu}_{0,g}\right)$. For each active group g at k^{th} iteration,*

$$\|\mathbf{m}_g^{(k)} - \widetilde{\mathbf{m}}_g^*\| \leq (1 - \eta\beta_{\mathrm{KL}}\alpha_g)^k \|\mathbf{m}_g^{(0)} - \widetilde{\mathbf{m}}_g^*\|, \tag{15}$$

Theorem 2 establishes linear convergence for active parameter groups, with a rate controlled by the learned scale. Since the scales adapt based on how well the current data is explained by prior knowledge, TailAdapt automatically allocates faster adaptation to groups that require larger updates while maintaining stability elsewhere.

B.1 Theoretical Verification:

Fig. 6a shows that the baseline maintains high, unstable condition numbers for tail classes, indicating persistent ill-conditioning, whereas DAP+TailAdapt rapidly reduces them within two epochs

and keeps them low, consistent with adaptive curvature regularization that prevents near-singular directions in low-data regimes. Fig. 6b shows that although tail classes start with larger optimization errors than head classes, they decay quickly and nearly linearly, approaching the head-class regime, demonstrating improved stability and faster convergence under TailAdapt.

C Marginal Effect: Understanding the IG Prior Mathematically

When we marginalize out the latent scale variable z_g, we obtain the marginal prior over the prompt (or adapter) parameter u_g:

$$p(u_g) = \int_0^\infty \mathcal{N}(u_g \mid \mu_{0,g}, z_g I_{d_g}) \, \mathrm{IG}(z_g \mid \mu_g, \lambda_g) \, dz_g \tag{16}$$

For clarity we analyze a single coordinate. For $d_g > 1$, the marginal of $\mathbf{u}_g$ is a multivariate symmetric NIG: coordinates are uncorrelated but dependent through the shared z_g, so shrinkage acts at the group level rather than coordinate-wise. .We consider a single scalar parameter $u_g \in \mathbb{R}$ with prior center $\mu_{0,g}$. The scale mixture integral is:

$$p(u_g) = \int_0^\infty \frac{1}{\sqrt{2\pi z_g}} \exp\left(-\frac{(u_g - \mu_{0,g})^2}{2z_g}\right) \cdot \sqrt{\frac{\lambda_g}{2\pi z_g^3}} \exp\left(-\frac{\lambda_g (z_g - \mu_g)^2}{2\mu_g^2 z_g}\right) dz_g \tag{17}$$

Let $\delta_g = u_g - \mu_{0,g}$. Expanding the quadratic term in the Inverse Gaussian exponent yields:

$$-\frac{\lambda_g (z_g - \mu_g)^2}{2\mu_g^2 z_g} = -\frac{\lambda_g z_g^2 - 2\lambda_g \mu_g z_g + \lambda_g \mu_g^2}{2\mu_g^2 z_g} = -\frac{\lambda_g z_g}{2\mu_g^2} + \frac{\lambda_g}{\mu_g} - \frac{\lambda_g}{2z_g} \tag{18}$$

Combining exponents across the Gaussian likelihood and the prior gives:

$$p(u_g) = \frac{\sqrt{\lambda_g}}{2\pi} \exp\left(\frac{\lambda_g}{\mu_g}\right) \int_0^\infty z_g^{-2} \exp\left(-\frac{\delta_g^2 + \lambda_g}{2z_g} - \frac{\lambda_g}{2\mu_g^2} z_g\right) dz_g \tag{19}$$

C.1 Exact Closed-Form Density: Normal-Inverse Gaussian (NIG)

The integral in Eq. 19 matches the integral representation of the modified Bessel function of the second kind, $K_\nu(\cdot)$:

$$\int_0^\infty x^{\nu-1} \exp\left(-\frac{A}{2x} - \frac{B}{2}x\right) dx = 2\left(\frac{A}{B}\right)^{\nu/2} K_\nu\left(\sqrt{AB}\right), \quad \text{for } A > 0, B > 0$$

Setting $\nu = -1$, $A = \delta_g^2 + \lambda_g$, and $B = \frac{\lambda_g}{\mu_g^2}$, and using the identity $K_{-1}(\cdot) = K_1(\cdot)$, the marginal prior admits an exact closed-form solution:

$$p(u_g) = \frac{\sqrt{\lambda_g}}{\pi} \exp\left(\frac{\lambda_g}{\mu_g}\right) \left(\frac{\delta_g^2 + \lambda_g}{\lambda_g/\mu_g^2}\right)^{-1/2} K_1\left(\sqrt{\frac{\lambda_g}{\mu_g^2}(\delta_g^2 + \lambda_g)}\right)$$

Simplifying with $\gamma_g = \frac{\sqrt{\lambda_g}}{\mu_g}$:

$$p(u_g) = \frac{\lambda_g}{\pi \mu_g} \exp\left(\frac{\lambda_g}{\mu_g}\right) \frac{K_1\left(\gamma_g \sqrt{\delta_g^2 + \lambda_g}\right)}{\sqrt{\delta_g^2 + \lambda_g}} \tag{20}$$

This confirms that the hierarchical model induces a symmetric Normal-Inverse Gaussian (NIG) marginal prior on u_g, parameterized by steepness $\alpha = \sqrt{\gamma_g^2} = \gamma_g$ and scale $\beta = 0$.

C.2 Asymptotic Regimes and Sparsity Properties

The selective plasticity behavior can be evaluated directly from the asymptotic properties of $K_1(\xi)$:
(i) Near Center ($\delta_g \to 0$)**: Strong Local Shrinkage**
For arguments near the mode $\xi_0 = \gamma_g\sqrt{\lambda_g} = \frac{\lambda_g}{\mu_g}$, when λ_g is relatively small, the Bessel function expands as $K_1(\xi) \sim \frac{1}{\xi}$. Substituting this back into Eq. 20:

$$p(u_g) \propto \frac{1}{\delta_g^2 + \lambda_g} = \frac{1}{(u_g - \mu_{0,g})^2 + \lambda_g}$$

The effective regularization penalty on deviations from $\mu_{0,g}$ corresponds to the negative log-marginal density:

$$-\log p(u_g) = \log\left((u_g - \mu_{0,g})^2 + \lambda_g\right) + \text{const}$$

Regularization gradient is given by,

$$\nabla_{u_g}[-\log p(u_g)] = \frac{2(u_g - \mu_{0,g})}{(u_g - \mu_{0,g})^2 + \lambda_g}$$

Sparsity mechanism: As $\lambda_g \to 0$ (the heavy-tailed limit of the IG prior), $\nabla_{u_g}[-\log p(u_g)] \to \frac{2}{u_g - \mu_{0,g}}$, approaching a non-convex, sub-exponential sparsity prior that penalizes tiny perturbations with high force while stabilizing around zero for small, non-zero λ_g. For head-class dimensions where the gradient of the log-likelihood is small, parameters are strongly anchored to the prior mode $\mu_{0,g}$.

(ii) In the Tails ($|\delta_g| \to \infty$)**: Semi-Heavy Tails and Bounded Regularization**
For large parameter adaptations ($|\delta_g| \gg \sqrt{\lambda_g}$), the argument $\xi = \gamma_g\sqrt{\delta_g^2 + \lambda_g} \approx \frac{\sqrt{\lambda_g}}{\mu_g}|\delta_g|$ becomes large. Using the asymptotic expansion of the Bessel function for large arguments:

$$K_1(\xi) = \sqrt{\frac{\pi}{2\xi}}\exp(-\xi)\left(1 + \mathcal{O}\left(\frac{1}{\xi}\right)\right)$$

Substituting this expansion into Eq. 20 yields:

$$p(u_g) \propto \frac{1}{\sqrt{|\delta_g|}} \cdot \frac{1}{|\delta_g|}\exp\left(-\frac{\sqrt{\lambda_g}}{\mu_g}|\delta_g|\right) = \frac{1}{|\delta_g|^{3/2}}\exp\left(-\frac{\sqrt{\lambda_g}}{\mu_g}|\delta_g|\right)$$

Unlike a standard Gaussian prior, where the penalty grows quadratically ($\propto |\delta_g|^2$) and exerts an unbounded restoring force that prevents tail-class adaptation, the marginal IG prior imposes an ℓ_1-like (Laplace-type) linear tail with logarithmic dampening.This bounded penalty ensures that when a parameter group receives persistent gradient signals to adapt to novel tail-class semantics, the prior does not over-penalize large deviations, thereby preserving the plasticity of critical representations without destructive interference.

D Proof of Proposition 1

$$\mathcal{L}(\{m_g\}, \{z_g\}) = \ell(\mathbf{m}) - \beta_{\mathrm{KL}}\sum_{g=1}^{G}\alpha_g\|\mathbf{m}_g - \mu_{0,g}\|^2 - \beta_{\mathrm{KL}}\sum_{g=1}^{G}\mathrm{KL}\big(q(z_g)\,\|\,p(z_g)\big), \tag{21}$$

The KL term is

$$\mathrm{KL}(q\|p) = \mathbb{E}_q\big[\log q(z_g)\big] - \mathbb{E}_q\big[\log p(z_g)\big].$$

For an Inverse-Gaussian prior,

$$p(z) = \sqrt{\frac{\lambda_g}{2\pi z^3}}\exp\left(-\frac{\lambda_g\big(z - \mu_g\big)^2}{2\big(\mu_g\big)^2 z}\right),$$

the dominant z-dependent term in $-\log p(z)$ is

$$-\log p(z) = \frac{\lambda_g}{2\big(\mu_g\big)^2}\, z + \frac{3}{2}\log z + \frac{\lambda_g}{2z} + O(1).$$

Taking expectation under $q(z)$,

$$\mathbb{E}_q[-\log p(z)] = \frac{\lambda_g}{2\big(\mu_g\big)^2}\, \mathbb{E}_q[z] + \frac{3}{2}\, \mathbb{E}_q[\log z] + \frac{\lambda_g}{2}\mathbb{E}_q[z^{-1}] + O(1).$$

For a log-normal,

$$\mathbb{E}_q[z] = \exp\big(a_g + \tfrac{1}{2}b_g^2\big)\,, \qquad \mathbb{E}_q[\log z] = a_g, \qquad \mathbb{E}_q[z^{-1}] = \exp\left(-a_g + \frac{1}{2}b_g^2\right)$$

Collecting terms that are functions of a_g:

$$\mathcal{L}_g = -\frac{\beta_{\mathrm{KL}}}{2}\, e^{-a_g + \frac{1}{2}b_g^2}\, (\Delta_g + \lambda_g) - \beta_{\mathrm{KL}} \left(\frac{\lambda_g}{2\big(\mu_g\big)^2}\, e^{a_g + \frac{1}{2}b_g^2} + \frac{3}{2}\, a_g \right) + \text{other term indep. of } a_g.$$

Taking derivative and setting to zero:

$$\frac{\partial \mathcal{L}_g}{\partial a_g} = \frac{\beta_{\mathrm{KL}}}{2}\, e^{-a_g + \frac{1}{2}b_g^2}\, (\Delta_g + \lambda_g) - \beta_{\mathrm{KL}}\, \frac{\lambda_g}{2\big(\mu_g\big)^2}\, e^{a_g + \frac{1}{2}b_g^2} - \frac{3}{2}\, \beta_{\mathrm{KL}} = 0.$$

For nonzero $\Delta_g = \left\|\mathbf{m}_g^\star - \boldsymbol{\mu}_{0,g}\right\|^2$, the exponential terms dominate, and the linear term in a_g is lower order. Ignoring the constant term (this is the only approximation), we get:

$$e^{-a_g}\, (\Delta_g + \lambda_g) = \frac{\lambda_g}{\big(\mu_g\big)^2}\, e^{a_g}.$$

Rearranging:

$$e^{2a_g} = \frac{\big(\mu_g\big)^2}{\lambda_g}\, (\Delta_g + \lambda_g).$$

Thus

$$e^{-a_g} = \sqrt{\frac{\lambda_g}{\big(\mu_g\big)^2}}\, \frac{1}{\sqrt{(\Delta_g + \lambda_g)}}.$$

Substitute back into $\alpha_g = e^{-a_g + \frac{1}{2}b_g^2}$ and applying $\|m_g^* - \mu_{0,g}\| \gg \sqrt{\lambda_g}$:

$$\alpha_g^\star = \underbrace{e^{\frac{1}{2}b_g^2}\, \sqrt{\frac{\lambda_g}{\big(\mu_g\big)^2}}}_{c_1}\, \frac{1}{\left\|\mathbf{m}_g^\star - \boldsymbol{\mu}_{0,g}\right\|},$$

E Proof of Theorem 1

By definition,

$$\tilde{\mathbf{H}}_g = \mathbf{H}_g + \beta_{\mathrm{KL}}\, \alpha_g^\star \mathbf{I}.$$

From Assumption, the block Hessian $\mathbf{H}_g$ satisfies

$$\mathbf{0} \preceq \mathbf{H}_g \preceq L_g \mathbf{I}.$$

Therefore, for every eigenvalue $\lambda_i(\mathbf{H}_g)$,

$$0 \le \lambda_i(\mathbf{H}_g) \le L_g.$$

Adding $\beta_{\mathrm{KL}}\alpha_g^\star\mathbf{I}$ shifts all eigenvalues uniformly, yielding

$$\lambda_i\Big(\tilde{\mathbf{H}}_g\Big) = \lambda_i(\mathbf{H}_g) + \beta_{\mathrm{KL}}\,\alpha_g^\star.$$

The smallest eigenvalue is

$$\lambda_{\min}\Big(\tilde{\mathbf{H}}_g\Big) = \beta_{\mathrm{KL}}\,\alpha_g^\star.$$

The largest eigenvalue satisfies

$$\lambda_{\max}\Big(\tilde{\mathbf{H}}_g\Big) = \lambda_{\max}(\mathbf{H}_g) + \beta_{\mathrm{KL}}\,\alpha_g^\star \le L_g + \beta_{\mathrm{KL}}\,\alpha_g^\star.$$

Substituting this in the expression of condition number $\kappa(\tilde{\mathbf{H}}) = \frac{\lambda_{max}(\tilde{\mathbf{H}})}{\lambda_{min}(\tilde{\mathbf{H}})}$, we get the desired inequality.

F Proof of Theorem 2

The likelihood admits the following local quadratic approximation,

$$\ell(\boldsymbol{m}) = \ell(\boldsymbol{m}^\star) - \frac{1}{2}\sum_{g=1}^{G}(\boldsymbol{m}_g - \boldsymbol{m}_g^\star)^\top \mathbf{H}_g(\boldsymbol{m}_g - \boldsymbol{m}_g^\star), \tag{22}$$

with $0 \le \mathbf{H}_g \le L_g\mathbf{I}$ Substituting it in ELBO, we get,

$$\mathcal{L}_g(\mathbf{m}_g) = C - \frac{1}{2}(\mathbf{m}_g - \mathbf{m}_g^\star)^\top \mathbf{H}_g(\mathbf{m}_g - \mathbf{m}_g^\star) - \frac{\beta_{\mathrm{KL}}\alpha_g^\star}{2}\|\mathbf{m}_g - \boldsymbol{\mu}_{0,g}\|^2 \tag{23}$$

Define effective curvature,

$$\tilde{\mathbf{H}}_g := \mathbf{H}_g + \beta_{KL}\alpha_g^\star\mathbf{I} \tag{24}$$

From Theorem 1, we know that,

$$\beta_{KL}\alpha_g^\star\mathbf{I} \le \tilde{\mathbf{H}}_g \le (L_g + \beta_{KL}\alpha_g^\star)\mathbf{I} \tag{25}$$

The gradient ascent update for group g is

$$\mathbf{m}_g^{(k+1)} = \mathbf{m}_g^{(k)} + \eta\nabla_{\mathbf{m}_g}\mathcal{L}_g\Big(\mathbf{m}_g^{(k)}\Big) \tag{26}$$

Using Eq. 23,

$$\nabla_{\mathbf{m}_g}\mathcal{L}_g(\mathbf{m}_g) = -\tilde{\mathbf{H}}_g\big(\mathbf{m}_g - \widetilde{\mathbf{m}}_g^\star\big) \tag{27}$$

Define the error vector

$$\mathbf{e}_g^{(k)} := \mathbf{m}_g^{(k)} - \widetilde{\mathbf{m}}_g^\star \tag{28}$$

Then

$$\mathbf{e}_g^{(k+1)} = \big(\mathbf{I} - \eta\tilde{\mathbf{H}}_g\big)\mathbf{e}_g^{(k)}. \tag{29}$$

The eigenvalues of $\mathbf{I} - \eta\tilde{\mathbf{H}}_g$ are

$$1 - \eta\lambda_i\Big(\tilde{\mathbf{H}}_g\Big).$$

Moreover,

$$1 - \eta\big(L_g + \beta_{KL}\alpha_g^\star\big) \le 1 - \eta\lambda_i\Big(\tilde{\mathbf{H}}_g\Big) \le 1 - \eta\beta_{KL}\alpha_g^\star.$$

The step-size condition

$$0 < \eta \le \frac{1}{L_g + \beta_{KL}\alpha_g^\star}$$

ensures that all eigenvalues lie in $(-1, 1)$, so the spectral radius satisfies

$$\rho\Big(\mathbf{I} - \eta\tilde{\mathbf{H}}_g\Big) = 1 - \eta\beta_{KL}\alpha_g^\star \tag{30}$$

Using norm on Eq. 29 and using Eq. 30,

$$\|\mathbf{e}_g^{(k+1)}\| \le \rho\Big(\mathbf{I} - \eta\tilde{\mathbf{H}}_g\Big)\,\|\mathbf{e}_g^{(k)}\| = \big(1 - \eta\beta_{KL}\alpha_g^\star\big)\,\|\mathbf{e}_g^{(k)}\|.$$

Iterating yields

$$\|\mathbf{m}_g^{(k)} - \mathbf{m}_g^\star\| \le \big(1 - \eta\beta_{KL}\alpha_g^\star\big)^k\,\|\mathbf{m}_g^{(0)} - \widetilde{\mathbf{m}}_g^\star\|.$$

G Bias Analysis for Log-Normal Surrogate

We analyze the bias induced by approximating the Inverse Gaussian (IG) latent scale variables with a log-normal variational surrogate. We characterize the bias induced by this approximation and show that its sign and magnitude depend on the learned posterior log-scale variance, leading to different regularization regimes.

(i) Problem Setup: In our hierarchical model, each parameter group is associated with a latent scale variable z_g following an Inverse Gaussian prior,

$$p(z) = \mathrm{IG}(z \mid \mu, \lambda),$$

with density $p(z) = \sqrt{\frac{\lambda}{2\pi z^3}} \exp\left(-\frac{\lambda(z-\mu)^2}{2\mu^2 z}\right), \quad z > 0$. Due to the difficulty of reparameterizing the IG distribution, we approximate the variational posterior using a log-normal distribution,

$$q(z) = \mathrm{LogNormal}(z|a, b^2),$$

with density $q(z) = \frac{1}{z\sqrt{2\pi b^2}} \exp\left(-\frac{(\log z - a)^2}{2b^2}\right), \quad z > 0$. The variational parameters are learned by minimizing KL divergence and the conditional Gaussian KL contribution associated with the deviation of the parameter-group mean from its prior center $\Delta = \|\mathbf{m}_g - \boldsymbol{\mu}_{0,g}\|^2$. The objective for the stationary conditions is given by,

$$J(a, b^2) = KL(q||p) + \frac{\Delta}{2}\mathbb{E}_q[z^{-1}] = \mathbb{E}_q[\log q(z)] - \mathbb{E}_q[\log p(z)] + \frac{\Delta}{2}\mathbb{E}_q[z^{-1}] \tag{31}$$

(ii) Closed Form Expression of KL objective: For the log-normal distribution, the following moments hold:

$$\mathbb{E}_q[\log z] = a, \quad \mathbb{E}_q[z] = e^{a+\frac{1}{2}b^2}, \quad \mathbb{E}_q[z^{-1}] = e^{-a+\frac{1}{2}b^2} \tag{32}$$

Using these identities, the expectation of variational entropy term is given by,

$$\mathbb{E}_q[\log q(z)] = -a - \frac{1}{2}\log(2\pi b^2) - \frac{1}{2} \tag{33}$$

Now, the log-density of the IG distribution can be written as,

$$\log p(z) = C - \frac{3}{2}\log z - \frac{\lambda}{2\mu^2}\left(z - 2\mu + \frac{\mu^2}{z}\right), \tag{34}$$

where C is the constant independent of z. Taking expectation under q, we get,

$$\mathbb{E}_q[\log p(z)] = C - \frac{3}{2}a - \frac{\lambda}{2\mu^2}\left(e^{a+\frac{1}{2}b^2} - 2\mu + \mu^2 e^{-a+\frac{1}{2}b^2}\right) \tag{35}$$

Substituting Eq. 34 and 35 in the Eq. 31, KL divergence term becomes,

$$J(a, b^2) = \frac{1}{2}a - \frac{1}{2}\log b^2 + \frac{\lambda}{2\mu^2}\left(e^{a+\frac{1}{2}b^2} + \mu^2 e^{-a+\frac{1}{2}b^2} - 2\mu\right) + \frac{\Delta}{2}\beta + K, \tag{36}$$

(iii) Stationary Conditions: Minimizing Eq. 36 w.r.t. a and b^2 to yield first order optimality condition,

$$\frac{\partial}{\partial a} J(a, b^2) = \frac{1}{2} + \frac{\lambda}{2\mu^2}\left(e^{a+\frac{1}{2}b^2} - \mu^2 e^{-a+\frac{1}{2}b^2}\right) - \frac{\Delta}{2}\beta = 0, \tag{37}$$

and

$$\frac{\partial}{\partial b^2} J(a, b^2) = -\frac{1}{2b^2} + \frac{\lambda}{4\mu^2}\left(e^{a+\frac{1}{2}b^2} + \mu^2 e^{-a+\frac{1}{2}b^2}\right) + \frac{\Delta}{2}\beta = 0 \tag{38}$$

Let $\alpha = e^{a+b^2/2}$ and $\beta = e^{-a+b^2/2}$. Then Eq. 37 and 38 can be written as,

$$1 + \frac{\lambda}{\mu^2}(\alpha - \mu^2\beta) - \Delta\beta = 0 \tag{39}$$

$$\frac{\lambda}{\mu^2}\alpha + (\lambda + \Delta)\beta = \frac{2}{b^2}. \tag{40}$$

Metric	IG	Log-Normal	Gamma	Gaussian
$\overline{Acc}$	63.24	59.08	59.48	58.98
$\mathcal{F}$	9.28	11.31	10.84	11.25

Table 3: Performance comparison for different heavy tail priors for CIFAR-100 in DAP baseline.

	CIFAR-100	ImageNet-R	CUB-200
CodaP	0.533	0.968	0.352
DualP	0.632	0.421	0.336
EASE	0.702	0.464	0.422
APER	-0.436	0.545	-0.601

Table 4: Surrogate bias analysis across different datasets under the 5-task setting.

Solving the linear equations gives $\alpha = \frac{\mu^2}{\lambda}(\frac{1}{b^2} - \frac{1}{2})$ and $\beta = \frac{1}{\lambda+\Delta}(\frac{1}{b^2} + \frac{1}{2})$.
From Eqs. 32, the inverse scale moment under the surrogate posterior is

$$\mathbb{E}_q[z^{-1}] = \beta = \frac{1}{\lambda + \Delta}(\frac{1}{b^2} + \frac{1}{2})$$

Following Chhikara and Folks [1988] for Inverse Gaussian, the corresponding moment is

$$\mathbb{E}_p[z^{-1}] = \frac{1}{\mu} + \frac{1}{\lambda}$$

The corresponding bias is computed as,

$$\delta = \mathbb{E}_q[z^{-1}] - \mathbb{E}_p[z^{-1}] = \frac{1}{\lambda + \Delta}\left(\frac{1}{b^2} + \frac{1}{2}\right) - \left(\frac{1}{\mu} + \frac{1}{\lambda}\right) \tag{41}$$

Since $\mu, \lambda > 0$, the magnitude of b controls the sign of the bias.

When $\delta > 0$. This condition arises early in the training where we explicitly choose smaller b so that the regularization strength remains stronger in the initial phase of training to ensure stability. Similarly, $\delta < 0$ emerges when the optimizer has identified the group as requiring task-specific plasticity.

G.1 Choice of Heavy-Tailed Prior:

To validate that the IG prior's heavy-tailed structure is critical for selective sparsity, we compare four hierarchical prior families while keeping all other components identical. As shown in Table 3, the IG prior achieves the best average accuracy and lowest forgetting outperforming the other heavy tail alternatives and the Gaussian. We attribute this observation to primary reasons, firstly, IG has semi-heavy tail, intermediate between Gamma's exponential tail (too light, insufficient jumps) and Log-Normal's super polynomial tail (too heavy, training instability). This permits large prompt updates when needed while maintaining numerical stability. Second, the IG parameters directly control mean and precision separately, simplifying moment matching for tempered updates. Log-Normal and Gamma require nonlinear transformations that can destabilize the training.

G.2 Surrogate Bias Analysis

Based on the closed form expression of the bias, we empirically compute the average bias across the sessions using the corresponding parameters obtained after each incremental session. Table 4 provides the empirical results for three datasets using EASE (Zhou et al. [2024a]), APER (Zhou et al. [2024b]), DualPrompt (Wang et al. [2022]), and CodaPrompt (Smith et al. [2023]) as the baselines. It can be observed that the induced bias has magnitude less than 1, which proves that the log-normal approximation does not have any significant impact on the overall performance.

H Parametric Complexity

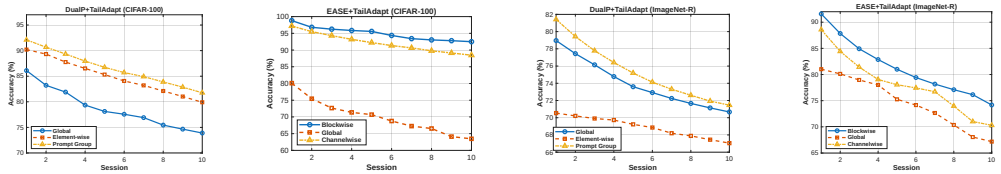
We quantify the additional overhead introduced by TailAdapt in terms of trainable parameters. Table 5 reports both the absolute parameter counts and the percentage increase in total parameters across

Backbone	Trainable parameters in TailAdapt (M)	% Additional parameters
SLCA++	1.54×10^{-2}	0.018
EASE	4.71	5.50
MOS	0.82	0.96
DualP	0.43	0.51
CodaP	4.21	4.89

Table 5: Comparison of parametric complexity due to TailAdapt across different backbones.

Model	CIFAR-100				ImageNet-R				CUB-200			
	10 Tasks		5 Tasks		10 Tasks		5 Tasks		10 Tasks		5 Tasks	
	$\overline{Acc}\uparrow$	$\mathcal{F}\downarrow$	$\overline{Acc}\uparrow$	$\mathcal{F}\downarrow$	$\overline{Acc}\uparrow$	$\mathcal{F}\downarrow$	$\overline{Acc}\uparrow$	$\mathcal{F}\downarrow$	$\overline{Acc}\uparrow$	$\mathcal{F}\downarrow$	$\overline{Acc}\uparrow$	$\mathcal{F}\downarrow$
SLCA++ +TailAdapt	$93.58_{\pm0.93}$	4.56	$94.28_{\pm1.10}$	4.33	$82.49_{\pm0.91}$	4.69	$84.95_{\pm0.77}$	5.10	$90.99_{\pm0.86}$	10.82	$92.47_{\pm0.78}$	7.43
EASE +TailAdapt	$92.52_{\pm0.97}$	6.35	$95.27_{\pm0.90}$	5.25	$74.84_{\pm0.48}$	6.89	$75.92_{\pm0.56}$	6.36	$88.57_{\pm0.69}$	10.49	$88.36_{\pm0.81}$	9.36
MOS +TailAdapt	$92.74_{\pm1.12}$	4.86	$93.76_{\pm0.98}$	4.36	$79.61_{\pm1.05}$	8.32	$79.85_{\pm0.67}$	7.09	$92.79_{\pm0.87}$	5.24	$91.92_{\pm0.79}$	8.76
APER +TailAdapt	$90.56_{\pm0.97}$	4.96	$91.11_{\pm0.81}$	4.21	$80.97_{\pm0.82}$	4.78	$80.96_{\pm0.78}$	4.33	$91.12_{\pm1.02}$	5.42	$90.46_{\pm0.92}$	5.02
DualP +TailAdapt	$85.91_{\pm0.76}$	6.59	$86.48_{\pm0.49}$	4.43	$72.26_{\pm0.37}$	5.06	$73.18_{\pm0.51}$	3.97	$80.23_{\pm0.60}$	11.49	$83.24_{\pm0.61}$	8.73
CodaP +TailAdapt	$88.54_{\pm0.68}$	5.33	$89.27_{\pm0.71}$	6.41	$78.85_{\pm0.62}$	5.41	$79.12_{\pm0.80}$	4.55	$87.50_{\pm0.91}$	10.56	$89.69_{\pm0.89}$	6.91
DAP +TailAdapt	$63.24_{\pm0.43}$	9.28	$58.95_{\pm0.35}$	9.22	$47.06_{\pm0.65}$	9.36	$46.25_{\pm0.59}$	9.12	$51.92_{\pm0.66}$	8.50	$50.12_{\pm0.49}$	8.97

Table 6: Performance comparison (with error bars) of TailAdapt across different continual learning baselines in shuffled LT-CIL setup.



(a) (b) (c) (d)

Figure 7: Sessionwise accuracies for different grouping strategies on CIFAR-100 using (a) DualPrompt and (b) EASE, and Imagenet-R using (c) DualPrompt and (d) EASE.

different continual learning frameworks. Overall, the increase in training overhead is marginal for most methods, with the exception of EASE and CodaPrompt. This discrepancy arises from their default configurations, specifically, the large prompt length and prompt pool size in CodaPrompt, and the relatively high number of adapters used in EASE.

I Grouping Structure for Selective Plasticity

A key design choice in our framework is the granularity at which plasticity is applied. In our formulation, we adopt prompt-level grouping for prompt tuning and adapter block-level grouping for low-rank adaptation as the default configurations, as these align naturally with the modular structure of the model.

To better understand the role of grouping, we perform an ablation study over different grouping strategies. For adapter-based fine-tuning, we consider the following variants:

- **Global grouping:** All adapter parameters share a single latent scale, resulting in uniform plasticity across layers.
- **Channel-wise grouping:** A separate latent scale is assigned to each output channel of the adapter. Specifically, for an adapter with C output channels, the model learns C latent variables, each shared across all parameters within a channel.
- **Block-wise grouping (default):** All parameters within an adapter block share a single latent scale, enabling structured adaptation at the layer level.

For prompt based fine-tuning methods, we analyze the following grouping strategies:

- **Global grouping:** All prompt parameters share a single latent scale, enforcing uniform plasticity across the entire prompt.

- **Element-wise grouping:** Each prompt parameter is assigned an independent latent scale, enabling fully fine-grained control over plasticity.
- **Token-wise grouping (default):** A shared latent scale is assigned to each prompt token, such that all embedding dimensions within a token are modulated jointly.

To this end, we experiment with CIFAR-100 and ImageNet-R datasets with 10 incremental sessions in each and decay factor of $\rho = 0.01$. For each grouping strategies, we fine-tune the model and observe the sessionwise accuracies. We use DualPrompt and EASE as baselines for prompt and adapter grouping, respectively. Figures 7a and 7b compares the performance of the respective strategies on CIFAR-100 dataset. It can be observed that the global grouping leads to the worst performance since it enforces a single shared latent scale across all parameters, resulting in uniform plasticity. This severely limits the model's ability to selectively adapt different components, which is critical in continual learning where parameters contribute unequally to new and previously learned tasks. Furthermore, we observe that overly fine-grained grouping, such as channel-wise grouping in adapters and element-wise grouping in prompts, tends to degrade performance. While these configurations provide maximal flexibility by assigning independent latent scales to small parameter subsets, they introduce a large number of degrees of freedom, leading to high variance updates and unstable optimization. In contrast, coarser structured grouping (e.g., tokenwise or blockwise) acts as an implicit regularizer by enforcing shared behavior within semantically meaningful units. This reduces variance, stabilizes training, and yields better generalization across tasks. Similar behavior can also be observed in the case of Imagenet-R in Figures 7c and 7d.

J Error Bars for TailAdapt Performance

Table 6 provides comparison along with the corresponding standard deviations by using 10 different seeding for random initialization.

K Computer and GPU Details

The details of the machine used for training and evaluation are as follows:

- **CPU:** Intel(R) Xeon(R) Silver 4314 CPU @ 2.40 GHz
- **RAM:** 128GB
- **GPU Details:** Nvidia RTX A6000 with 48 GB VRAM with CUDA 12.2

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 support for the claims can be found in the motivation part of Section 1, and Section 5.

   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: The limitation is discussed in the end of Section 6.

   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: All the necessary asumption along with the proof and empirical observations are provided in the appendix section.

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: All the necessary details including hyper-parameters and training setup are discussed in the paper.

   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: The code is released through GitHub repository and is available at `https://github.com/darkbird1/TailAdapt`.

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: The train-test split along with other experimental setting details are provided in the paper.

   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: Most of the experiments do not need error bars in our case.

   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 requirement of computer resources are highlighted in the paper.

   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: The research adheres to the code of ethics listed by NeurIPS.

   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: No societal impact.

    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: No such risk is possible through our research code and data.

    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: All the original oweners are duly cited wherever possible in the work.

    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: [N/A]

    Justification:

    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:

    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:

    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:

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.