Dynamical Adapter Fusion: Constructing A Global Adapter for Pre-Trained Model-based Class-Incremental Learning

Ruiqi Liu[1,2] **Boyu Diao**[1,2*] **Zijia An**[1,2] **Runjie Shao**[1,2]
Feixiang Ge[1,2] **Zhulin An**[1,2] **Fei Wang**[1,2] **Yongjun Xu**[1,2]

[1]Institute of Computing Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
liuruiqi23@mails.ucas.ac.cn diaoboyu2012@ict.ac.cn

Abstract

Class-Incremental Learning (CIL) requires models to continuously acquire new classes without forgetting previously learned ones. A dominant paradigm involves freezing a pre-trained model and training lightweight, task-specific adapters. However, maintaining task-specific parameters hinders knowledge transfer and incurs high retrieval costs, while naive parameter fusion often leads to destructive interference and catastrophic forgetting. To address these challenges, we propose Dynamical Adapter Fusion (DAF) to construct a single robust global adapter. Grounded in the PAC-Bayes theorem, we derive a fusion mechanism that explicitly integrates three components: the optimized task-specific adapter parameters, the previous global adapter parameters, and the initialization parameters. We utilize the Taylor expansion of the loss function to derive the optimal fusion coefficients, dynamically achieving the best balance between stability and plasticity. Furthermore, we propose a Robust Initialization strategy to effectively capture global knowledge patterns. Experiments on multiple CIL benchmarks demonstrate that DAF achieves state-of-the-art (SOTA) performance.

1 Introduction

The objective of Class-Incremental Learning is to sequentially assimilate novel categories while preventing the erasure of historical knowledge, a phenomenon known as catastrophic forgetting [16, 23]. Beyond the primary struggle to mitigate forgetting, the widespread application of CIL in resource-limited environments, including embodied AI and edge computing, has made computational efficiency an equally urgent research imperative [32, 20]. Generally, prevailing methods can be bisected into two streams: training from scratch and exploiting Pre-trained Models (PTMs). Traditional from-scratch methods, which rely on regularization, dynamic architectures, or data replay [19, 29, 27], frequently face intrinsic bottlenecks regarding data privacy, scalability, and performance saturation [40]. Accordingly, the field has witnessed a paradigm shift toward PTM-based frameworks [49]. By capitalizing on the robust, transferable features acquired from extensive pre-training, these methods consistently yield SOTA results. Within this domain, the dominant strategy involves locking the PTM backbone and exclusively employing Parameter-Efficient Fine-Tuning (PEFT) techniques, such as Prompt Tuning [42, 31] or Adapter Tuning [33]. This paradigm has become the standard for modern CIL, offering an optimal compromise between minimal resource consumption and superior classification accuracy.

*Corresponding author.

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

A common strategy trains separate adapters for each task [48], requiring retrieval or ensembling during inference [31, 48]. While this reduces task interference, it incurs high retrieval costs and risks performance degradation from incorrect selection. Moreover, memory usage grows with the number of tasks. Consequently, we aim to construct a single global adapter without storing historical modules. However, existing global methods often retain this storage overhead [38, 14], while simple fusion techniques like Exponential Moving Average (EMA) fail to capture sufficient global knowledge [26].

To this end, we propose Dynamical Adapter Fusion, a novel framework designed to construct a single robust global adapter for PTM-based CIL. We adopt the PAC-Bayes theorem to provide insight into the dilemma between effective learning and retaining previous knowledge. Our theoretical analysis reveals that the upper bound of the expected risk can be divided into three distinct parts: empirical risk, generalization regularization, and forgetting regularization. Guided by this analysis, upon the completion of each task, we fuse three components: the optimized task-specific adapter parameters, the previous global adapter parameters, and the initialization parameters of the current task-specific adapter. We utilize the Taylor expansion of the loss function to establish an ideal optimization objective for global adapter fusion that explicitly accounts for these three terms. By employing the method of Lagrange Multipliers to solve this objective, we derive the optimal fusion coefficients to adaptively achieve the best balance between stability and plasticity. Furthermore, we propose a Robust Initialization strategy that recursively averages historical adapters as an informative prior to capture global knowledge patterns and enhance robustness. Consequently, our entire framework operates without the need for storing historical adapters, resulting in the exceptional inference throughput and accuracy balance illustrated in Figure 1. In summary, our main contributions are as follows:

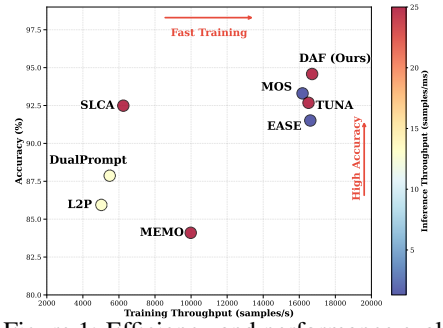


Figure 1: Efficiency and performance evaluation on CIFAR-100. We plot Average Accuracy ($\bar{A}$) against Training Throughput (samples/s). The color of each point represents the Inference Throughput (samples/ms). The ideal method occupies the top-right corner with a dark red color. TUNA [38] is evaluated here using its single global adapter. Our DAF achieves the best balance between speed and precision.

- We propose DAF, a framework developed from a PAC-Bayesian perspective to address the plasticity-stability dilemma. By formulating the fusion process as a constrained optimization problem and solving it via Lagrange Multipliers, we construct a single global adapter that dynamically integrates new knowledge, thereby eliminating the retrieval costs and storage overhead associated with maintaining historical adapters.
- We propose a memory-efficient robust initialization strategy that recursively averages historical adapters to serve as an informative prior. This strategy effectively anchors the learning process to global knowledge patterns, significantly enhancing model stability without accessing past data.
- We conduct extensive experiments on major CIL benchmarks, where DAF consistently outperforms SOTA methods with accuracy gains of up to 2.75%.

2 Related Work

CIL methods can be broadly categorized into two primary paradigms: training from scratch and leveraging PTMs.

2.1 CIL from Scratch

The conventional CIL paradigm involves training models sequentially starting from random initialization. These methods are generally classified into three major categories. **Regularization-based** methods [44, 29, 19, 43, 8, 2] add penalty terms to the loss function to constrain the drift of parameters essential for previous tasks. **Architecture-based** methods [11, 15, 21, 41, 46] prevent interference by allocating distinct model parameters or sub-networks to each incoming task. **Rehearsal-based** methods [1, 5, 6, 9, 7] maintain a small buffer of historical exemplars to approximate joint training

with past data. However, despite their fundamental importance, these from-scratch methods frequently encounter challenges related to scalability, data privacy, and performance saturation, prompting a transition toward the PTM-based paradigm [30, 40, 18].

2.2 Pre-Trained Model-based CIL

To reduce the high cost of training from scratch, recent CIL work heavily adopts PTMs with PEFT, freezing the large backbone and only optimizing a small set of incremental parameters. This makes CIL feasible under resource constraints. Diverse strategies exist: prompt-learning may share a prompt pool [42], assign disjoint prompts [40], or dynamically compose prompts [31]; other approaches fine-tune specific backbone blocks [45], combine PEFT with linear classifiers [49, 24], or append task-specific modules [48, 33].

A key challenge is effectively integrating task-specific parameters. Some recent methods employ retrieval and fusion: MOS [33] iteratively retrieves adapters at test time, TUNA [38] uses entropy-based retrieval with max-fusion, and CoMA/CoFiMA [22] introduce model merging via uniform averaging or Fisher-weighted ensembling. However, CoMA and CoFiMA are designed for full fine-tuning, ignore initialization weights, and rely on heuristic fusion coefficients. In contrast, our DAF derives optimal fusion weights based on a PAC-Bayes bound, constructing a single robust global adapter that balances stability and plasticity without inference-time retrieval.

3 Methodology

3.1 Preliminaries

Class-Incremental Learning. Formally, CIL involves learning a sequence of T tasks $\{\mathcal{D}_t\}_{t=1}^{T}$. Each dataset $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^{N_t}$ contains samples x_i with labels $y_i \in \mathcal{C}_t$. The label sets are disjoint, satisfying $\mathcal{C}_t \cap \mathcal{C}_{t'} = \emptyset$ for $t \neq t'$. Following the exemplar-free setting [42, 40, 49], we train the model using only the current data $\mathcal{D}_t$ without accessing past samples. After each task, the model is evaluated on the cumulative label set $\mathcal{Y}_t = \bigcup_{i=1}^{t} \mathcal{C}_i$.

PTM-based CIL via Task-Specific Adapters. We adopt a PTM-based framework (e.g., ViT [10]) where the backbone is frozen and only lightweight adapters are trained. Following [49, 38], bottleneck adapters are inserted parallel to the MLP layers. For input $\mathbf{x}_i \in \mathbb{R}^d$, the output $\mathbf{x}_o$ is computed as:

$$\mathbf{x}_o = \text{MLP}(\mathbf{x}_i) + \text{ReLU}(\mathbf{x}_i W_{\text{down}}) W_{\text{up}}, \tag{1}$$

where $W_{\text{down}} \in \mathbb{R}^{d \times r}$ and $W_{\text{up}} \in \mathbb{R}^{r \times d}$ are projection matrices with rank $r \ll d$. For each task t, a specific adapter set $\mathcal{A}_t$ is trained. To avoid memory expansion, we employ a fusion framework where $\mathcal{A}_t$ is integrated into a single global adapter $\mathcal{A}_g$ after training. Thus, we only maintain the global $\mathcal{A}_g$ and the current temporary $\mathcal{A}_t$. During inference, the model exclusively uses $\mathcal{A}_g$, eliminating the need for task-ID retrieval mechanisms [33, 38]. A unified classifier is constructed via prototype-based alignment (see Appendix C).

3.2 Theoretical Analysis

We analyze the generalization error using PAC-Bayes theorem [3, 37]. Our method uses a task-specific adapter and a global adapter. We define the model space as $\mathcal{H} = \mathcal{H}_s \times \mathcal{H}_g$, where $\mathcal{H}_s$ and $\mathcal{H}_g$ are parameter spaces for the task-specific adapter and global adapter. A model is $h = (h_s, h_g)$. Let $\mathcal{X}$ be the data space. Let loss $\ell : \mathcal{H} \times \mathcal{X} \to \mathbb{R}^+$ be bounded by K. For task t, the prior $P_t = P_{\text{init}} \otimes Q_{t-1}^g$ combines fixed initialization P_{init} and previous global posterior Q_{t-1}^g. The posterior $Q_t = Q_t^s \otimes Q_t^g$ combines the learned task-specific distribution Q_t^s and the updated global distribution Q_t^g. We define expected risk $\mathcal{R} = \mathbb{E}_{h \sim Q_t}\left[\mathbb{E}_{x \sim \mu_t}[\ell(h, x)]\right]$ and empirical risk $\hat{\mathcal{R}} = \frac{1}{N_t} \sum_{j=1}^{N_t} \mathbb{E}_{h \sim Q_t}\left[\ell(h, x_j)\right]$, where N_t is the sample size.

Theorem 3.1. *Let $\mathcal{D}_t = (x_1, ..., x_{N_t})$ be an iid set sampled from distribution μ_t for task t. Let prior $P_t = P_{init} \otimes Q_{t-1}^g$ and posterior $Q_t = Q_t^s \otimes Q_t^g$ be defined on $\mathcal{H}$. For any loss ℓ bounded by K,*

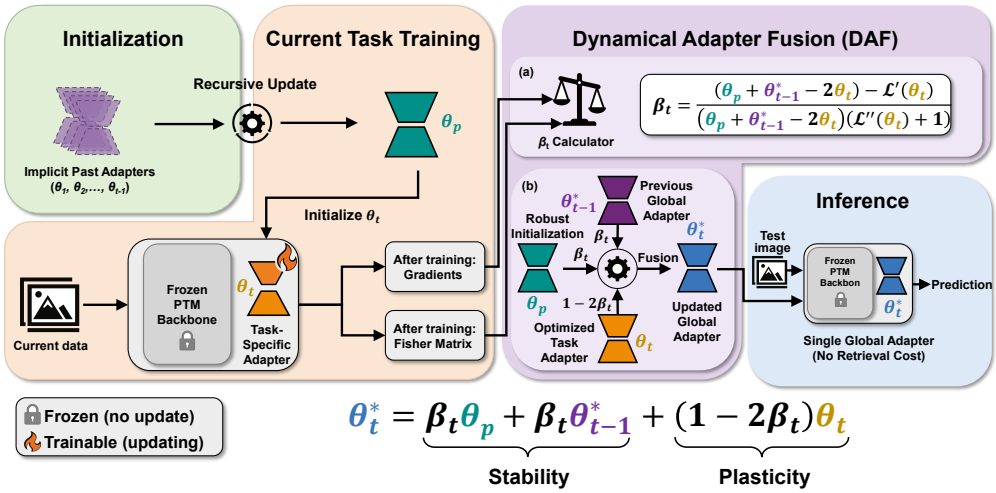


Figure 2: Overview of the proposed DAF framework. The pipeline consists of four distinct phases: **(1) Initialization:** We construct a **Robust Initialization** θ_p via a recursive update of implicit past adapters; **(2) Current Task Training:** The **Optimized Task Adapter** θ_t is learned on new data, and statistics (Gradients, Fisher Matrix) are computed; **(3) Dynamical Adapter Fusion :** This core module operates in two steps: (a) the β_t **Calculator** derives the optimal coefficient using the Lagrangian solution, and (b) the **Fusion** process integrates the Robust Initialization θ_p, the Previous Global Adapter θ^*_{t-1}, and the Optimized Task Adapter θ_t to form the updated model; **(4) Inference:** The resulting Single Global Adapter θ^*_t performs efficient, retrieval-free inference on all learned classes.

$\lambda > 0$, and $\delta \in [0, 1]$, the following inequality holds with probability $1 - \delta$:

$$\mathcal{R} \leq \underbrace{\hat{\mathcal{R}}}_{\textit{Empirical Risk}} + \underbrace{\frac{\mathrm{KL}(Q^s_t \| P_{init})}{\lambda}}_{\textit{Generalization Regularization}} + \underbrace{\frac{\mathrm{KL}(Q^g_t \| Q^g_{t-1})}{\lambda}}_{\textit{Forgetting Regularization}} + \frac{\log(1/\delta)}{\lambda} + \frac{\lambda K^2}{2N_t}. \tag{2}$$

The complete proof is in Appendix D. The bound consists of three key components: (1) **Empirical Risk** ($\hat{\mathcal{R}}$): Ensures plasticity by fitting current data. (2) **Generalization Regularization** ($\mathrm{KL}(Q^s_t \| P_{\text{init}})$): Constrains task-specific parameters to the robust initialization P_{init} to prevent overfitting. (3) **Forgetting Regularization** ($\mathrm{KL}(Q^g_t \| Q^g_{t-1})$): Constrains the global adapter to stay close to previous knowledge Q^g_{t-1}, ensuring stability. Thus, an optimal fusion mechanism must balance minimizing current loss while restricting deviations from both the initialization and the previous global state.

3.3 Dynamical Adapter Fusion Method

Guided by the PAC-Bayes analysis in Section 3.2, we propose Dynamical Adapter Fusion, as shown in Figure 2. Unlike expansion-based methods that rely on dynamic architectures, DAF maintains a fixed model capacity. Instead, it dynamically optimizes the fusion coefficients after each task to adapt to the changing data distribution. For each task t, we initialize a task adapter with parameters θ_p and optimize it on the current data to obtain θ_t. To ensure memory efficiency, we retain only a single global adapter θ^*_t by fusing the learned parameters. Specifically, Theorem 3.1 decomposes the bound into three terms, and each term corresponds to one parameter state: the empirical risk corresponds to the current task adapter θ_t, the generalization regularization keeps the parameters close to the initialization θ_p, and the forgetting regularization keeps the parameters close to the previous global adapter θ^*_{t-1}. Unlike previous fusion methods that only combine θ_t and θ^*_{t-1}, our bound identifies the initialization θ_p as a third necessary parameter state, which also motivates how we construct θ_p in Section 3.5. By combining all three parameter states to reduce the three terms together, we formulate

this fusion as:

$$\theta_t^* = \beta_t \theta_p + \beta_t \theta_{t-1}^* + (1 - 2\beta_t)\, \theta_t, \tag{3}$$

where θ denotes the individual parameters within the adapters, and the element-wise fusion coefficient β_t balances the three parameter states. Note that the PAC-Bayes bound serves as the motivation to identify which three parameter states to combine in Equation (3), rather than algebraically deriving the linear form. Based on this form, instead of setting β_t by hand as a fixed hyperparameter, we rigorously derive the closed-form β_t by minimizing the bound defined in Theorem 3.1.

For the *Empirical Risk*, we treat the optimized task adapter θ_t as a proxy target to avoid expensive retraining. We thus aim to minimize the loss gap $\mathcal{L}(\theta_t^*) - \mathcal{L}(\theta_t)$. Regarding regularization, Theorem 3.1 suggests constraining the deviations of θ_t from θ_p (Generalization Regularization) and θ_t^* from θ_{t-1}^* (Forgetting Regularization). To preserve plasticity, we enforce the generalization constraint during fusion rather than training. Specifically, we incorporate the deviation $\theta_t - \theta_p$ into the fusion objective to subtract excessive parameter shift, keeping θ_t^* close to the robust initialization. We introduce $\Delta\theta$ to capture the total parameter shift required by these stability constraints:

$$\Delta\theta = \theta_t^* - \theta_{t-1}^* + \theta_t - \theta_p. \tag{4}$$

By combining the risk approximation and the total parameter shift, we formulate the unified optimization problem as:

$$\begin{aligned} \min \quad & \mathcal{L}(\theta_t^*) - \mathcal{L}(\theta_t) + \frac{1}{2}\left(\theta_t^* - \theta_{t-1}^* + \theta_t - \theta_p\right)^2, \\ \textit{s.t.} \quad & \Delta\theta + \theta_{t-1}^* + \theta_p - \theta_t - \theta_t^* = 0. \end{aligned} \tag{5}$$

To solve the constrained minimization problem efficiently, we employ the method of Lagrange multipliers. This allows us to incorporate the stability constraint directly into the objective function by introducing a Lagrange multiplier λ. The Lagrangian function $\mathcal{J}$ is defined as:

$$\begin{aligned} \mathcal{J} = & \mathcal{L}(\theta_t^*) - \mathcal{L}(\theta_t) + \frac{1}{2}\left(\theta_t^* - \theta_{t-1}^* + \theta_t - \theta_p\right)^2 \\ & + \lambda\left(\Delta\theta + \theta_{t-1}^* + \theta_p - \theta_t - \theta_t^*\right). \end{aligned} \tag{6}$$

To facilitate the derivation, we approximate the loss difference using a second-order Taylor expansion around the optimized parameter θ_t. Omitting the high-order infinitesimal terms, we have:

$$\mathcal{L}(\theta_t^*) - \mathcal{L}(\theta_t) \approx \mathcal{L}'(\theta_t)(\theta_t^* - \theta_t) + \frac{\mathcal{L}''(\theta_t)}{2}(\theta_t^* - \theta_t)^2. \tag{7}$$

Furthermore, combining the fusion mechanism defined in Eq. (3) with the definition of $\Delta\theta$, we can express the deviation $\theta_t^* - \theta_t$ in terms of the relaxation factor $\Delta\theta$:

$$\theta_t^* - \theta_t = \frac{\beta_t}{\beta_t - 1}\Delta\theta. \tag{8}$$

Please refer to Appendix E for detailed demonstrations of this relationship.

Next, we substitute Eq. (3) into the constraint term of the Lagrangian. This transforms the stability constraint equation into a form dependent on β_t:

$$\theta_{t-1}^* + \theta_p - \theta_t - \theta_t^* = (1 - \beta_t)(\theta_p + \theta_{t-1}^* - 2\theta_t). \tag{9}$$

Substituting Eq. (7), Eq. (8), and Eq. (9) back into Eq. (6), we rewrite the Lagrangian $\mathcal{J}$ entirely in terms of β_t and $\Delta\theta$:

$$\begin{aligned} \mathcal{J} = & \mathcal{L}'(\theta_t)\frac{\beta_t}{\beta_t - 1}\Delta\theta + \frac{\mathcal{L}''(\theta_t)}{2}\left(\frac{\beta_t}{\beta_t - 1}\Delta\theta\right)^2 + \frac{1}{2}\Delta\theta^2 \\ & + \lambda\left[\Delta\theta + (1 - \beta_t)(\theta_p + \theta_{t-1}^* - 2\theta_t)\right]. \end{aligned} \tag{10}$$

To find the optimal solution, we first take the derivative of the Lagrangian with respect to β_t and set it to zero. This condition is essential for locating the stationary point of the function:

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \beta_t} = & -\frac{1}{(\beta_t - 1)^2}\mathcal{L}'(\theta_t)\Delta\theta - \frac{\beta_t}{(\beta_t - 1)^3}\mathcal{L}''(\theta_t)\Delta\theta^2 \\ & - \lambda(\theta_p + \theta_{t-1}^* - 2\theta_t) = 0. \end{aligned} \tag{11}$$

By solving these equations, we obtain a feasible solution for β_t that minimizes the objective function while satisfying the constraints:

$$\beta_t = \frac{(\theta_p + \theta^*_{t-1} - 2\theta_t) - \mathcal{L}'(\theta_t)}{(\theta_p + \theta^*_{t-1} - 2\theta_t)(\mathcal{L}''(\theta_t) + 1)}. \tag{12}$$

Please refer to Appendix F for the detailed deduction of this solution.

3.4 Optimization Approximation

To implement the optimal solution in Eq. (12), we require $\beta_t \in (0, 0.5)$ to balance stability and plasticity. The approximation $\beta_t \approx \frac{1}{\mathcal{L}''(\theta_t)+1}$ implies that $\mathcal{L}''(\theta_t) > 1$ is necessary for $\beta_t < 0.5$.

We approximate the Hessian diagonal using the Fisher Information Matrix (FIM) diagonal [16, 50], denoted as $\mathcal{F}(\theta_t)$, computed on the current task data $\mathcal{D}_t$. For a d-dimensional parameter vector θ_t, the j-th element of $\mathcal{F}(\theta_t)$ is computed on $\mathcal{D}_t$ as:

$$\mathcal{F}^{(j)}(\theta_t) = \mathbb{E}_{x\sim\mathcal{D}_t}\left[\mathbb{E}_{y\sim p(y|x,\theta_t)}\left[\left(\nabla_{\theta_t^{(j)}} \log p(y|x,\theta_t)\right)^2\right]\right], \tag{13}$$

where $p(y|x, \theta_t)$ is the model's output distribution. In practice, we approximate the inner expectation by sampling y from the model's output distribution $p(y|x, \theta_t)$ for each input x in a single additional forward-backward pass on $\mathcal{D}_t$ after task training, which has a low cost (see Appendix K). Let $\mathcal{F}_{\min} = \min_{1\le j\le d} \mathcal{F}^{(j)}(\theta_t)$ and $\bar{\mathcal{F}} = \frac{1}{d}\sum_{j=1}^{d} \mathcal{F}^{(j)}(\theta_t)$ be the minimum and mean values of $\mathcal{F}(\theta_t)$ across all d parameter dimensions. To satisfy $\mathcal{L}''(\theta_t) > 1$ while preserving relative parameter importance, we introduce a scaling hyperparameter $\alpha > 0$ to control β_t:

$$\mathcal{L}''(\theta_t) \approx \alpha \frac{\mathcal{F}(\theta_t) - \mathcal{F}_{\min}}{\bar{\mathcal{F}} - \mathcal{F}_{\min}} + 1. \tag{14}$$

This ensures the approximated Hessian lower bound is 1. Substituting Eq. (14) into Eq. (12) yields the computable coefficient:

$$\beta_t = \frac{(\bar{\mathcal{F}} - \mathcal{F}_{\min})\left[(\theta_p + \theta^*_{t-1} - 2\theta_t) - \mathcal{L}'(\theta_t)\right]}{(\theta_p + \theta^*_{t-1} - 2\theta_t)\left[\alpha\mathcal{F}(\theta_t) - \alpha\mathcal{F}_{\min} + 2\bar{\mathcal{F}} - 2\mathcal{F}_{\min}\right]}. \tag{15}$$

In practice, we clip β_t to $[0.001, 0.499]$ for numerical stability (see Appendix N for sensitivity analysis).

3.5 Robust Initialization via Historical Knowledge

As established in Section 3.2, θ_p serves as the center of the prior distribution P_{init}. Instead of standard random initialization, we utilize the average of previously learned adapters to improve stability. For a new task $t > 1$, we define θ_p as the arithmetic mean of the optimized parameters from the preceding $t - 1$ tasks:

$$\theta_p = \theta^{t-1}_{avg} = \frac{1}{t-1}\sum_{i=1}^{t-1} \theta_i, \tag{16}$$

where θ_i denotes the optimized parameters for task i. For the first task ($t = 1$), standard random initialization is applied.

Directly computing Eq. (16) requires storing all historical parameters, which causes high memory overhead. To avoid this, we use a recursive update. After training task t, we update the running average as:

$$\theta^t_{avg} = \frac{t-1}{t}\theta^{t-1}_{avg} + \frac{1}{t}\theta_t. \tag{17}$$

By maintaining only the running average θ_{avg}, this method constructs a robust initialization θ_p containing common historical patterns without requiring additional storage.

3.6 Overview of Our Method

The DAF training procedure for each task t consists of four key steps. First, we initialize the task-specific adapter θ_t using the prior θ_p (Eq. (16)) and train it on the current dataset $\mathcal{D}_t$. Second, we compute the fusion coefficient β_t via Eq. (15) using the gradient and Fisher diagonal derived from θ_t. Third, we fuse θ_t, θ_p, and the previous global adapter θ^*_{t-1} to update the global adapter θ^*_t (Eq. (3)). Finally, we update the running average θ_{avg} (Eq. (17)) and discard temporary parameters to ensure memory efficiency. The complete training procedure is summarized in Algorithm 1.

Algorithm 1 Dynamical Adapter Fusion (DAF) for CIL

In: Stream of tasks $\{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$, Pre-trained backbone ϕ, Adapter initialization θ_{init}, Learning rate η, Scaling factor α, Epochs E.
Out: Final global adapter θ^*_T.
Initialize: Global adapter $\theta^*_0 \leftarrow \theta_{init}$, Running average $\theta_{avg} \leftarrow \theta_{init}$.
For $t = 1, \ldots, T$ **do**
  1. Set prior $\theta_p \leftarrow \theta_{avg}$ and initialize task-specific adapter: $\theta_t \leftarrow \theta_p$.
  For $epoch = 1, \ldots, E$ **do**
    For each batch $(\mathbf{x}, y)$ in $\mathcal{D}_t$ **do**
      2. Compute Cross-Entropy loss $\mathcal{L}(\theta_t; \mathbf{x}, y)$.
      3. Update task-specific adapter: $\theta_t \leftarrow \theta_t - \eta \nabla_{\theta_t} \mathcal{L}$.
    End For
  End For
  4. Compute first-order gradient $\mathcal{L}'(\theta_t)$ and Fisher diagonal $\mathcal{F}(\theta_t)$ on $\mathcal{D}_t$.
  5. Compute statistics: $\mathcal{F}_{\min} \leftarrow \min(\mathcal{F}(\theta_t))$, $\bar{\mathcal{F}} \leftarrow \text{mean}(\mathcal{F}(\theta_t))$.
  6. Compute element-wise fusion coefficient β_t via Eq. (15):
    $\beta_t = \frac{(\bar{\mathcal{F}} - \mathcal{F}_{\min})[(\theta_p + \theta^*_{t-1} - 2\theta_t) - \mathcal{L}'(\theta_t)]}{(\theta_p + \theta^*_{t-1} - 2\theta_t)[\alpha\mathcal{F}(\theta_t) - \alpha\mathcal{F}_{\min} + 2\bar{\mathcal{F}} - 2\mathcal{F}_{\min}]}$.
  7. Clip β_t to valid range $[0.001, 0.499]$.
  8. Fuse parameters to update global adapter via Eq. (3):
    $\theta^*_t \leftarrow \beta_t\theta_p + \beta_t\theta^*_{t-1} + (1 - 2\beta_t)\theta_t$.
  9. Update running average via Eq. (17): $\theta_{avg} \leftarrow \frac{t-1}{t}\theta_{avg} + \frac{1}{t}\theta_t$.
  10. **Memory Cleanup:** Discard θ_t, clear statistics, and save only θ^*_t and θ_{avg} for next task.
End For
Return θ^*_T

4 Experiments

4.1 Implementation Details

Datasets and Task Protocols. We evaluate on CIFAR-100 [17], ImageNet-R [13], ImageNet-A [12], and ObjectNet [4], following the protocol of [49]. Full dataset and task construction details are in Appendix G.

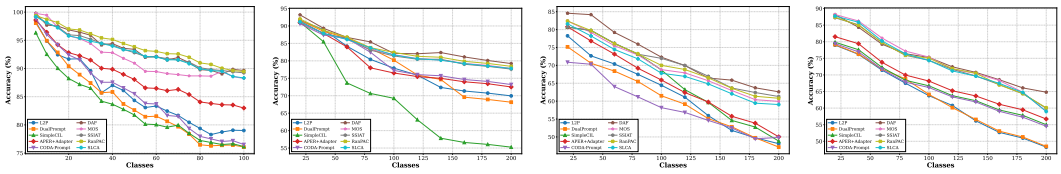
Training Details. All experiments use ViT-B/16-IN21K and ViT-B/16-IN1K as the primary backbone, implemented via the PILOT codebase [34]. The adapter bottleneck dimension is $r = 16$ and the scaling factor $\alpha = 1.25$. Full training configuration is in Appendix G.

Comparison Methods. We benchmark DAF against prominent PTM-based CIL methods: L2P [42], DualPrompt [40], CODA-Prompt [31], RanPAC [24], SimpleCIL [49], APER [49], SLCA [45], SSIAT [35], EASE [48], MOS [33], and TUNA [38]. We also include rehearsal-based methods FOSTER [39], MEMO [47], and iCaRL [28] (20 exemplars per class), and a sequential Finetune baseline as a lower bound.

Evaluation Metrics. We report Average Accuracy ($\bar{\mathcal{A}}$) and Final Accuracy ($\mathcal{A}_T$) following the standard protocol [28]. We additionally report Stability (S) and Plasticity (P) to analyze the plasticity-stability trade-off; formal definitions are provided in Appendix G.

Table 1: Average ($\bar{\mathcal{A}}$) and final task ($\mathcal{A}_T$) Top-1 accuracy comparison on four challenging class-incremental learning benchmarks. All methods use ViT-B/16-IN21K as the backbone and are evaluated in the exemplar-free setting. The specific task configurations (e.g., number of base classes and incremental steps) are detailed in the experimental setup section. The best performance in each column is highlighted in **bold**. † Results of the single global adapter.

Method	CIFAR-100		ImageNet-R		ImageNet-A		ObjectNet	
	$\bar{\mathcal{A}}$ (%)	$\mathcal{A}_T$ (%)	$\bar{\mathcal{A}}$ (%)	$\mathcal{A}_T$ (%)	$\bar{\mathcal{A}}$ (%)	$\mathcal{A}_T$ (%)	$\bar{\mathcal{A}}$ (%)	$\mathcal{A}_T$ (%)
Finetune	38.90	20.17	32.32	22.78	24.28	14.51	19.14	8.73
L2P [42]	85.94	79.93	75.46	69.77	49.39	41.71	63.78	52.19
DualPrompt [40]	87.87	81.15	73.10	67.18	53.71	41.67	59.27	49.33
CODA-Prompt [31]	89.11	81.96	77.97	72.27	53.54	42.73	66.07	53.29
SLCA [45]	92.49	88.55	81.17	77.00	68.66	58.74	72.55	61.30
RanPAC [24]	94.00	90.62	82.98	77.94	69.32	61.82	72.76	62.02
SSIAT [35]	93.52	90.07	83.20	78.85	70.83	62.23	73.65	62.45
SimpleCIL [49]	87.57	81.26	61.26	54.55	59.77	48.91	65.45	53.59
APER + Adapter [49]	90.65	85.15	75.82	67.95	60.47	49.37	67.18	55.24
EASE [48]	91.51	85.80	81.74	76.17	65.34	55.04	70.84	57.86
TUNA† [38]	92.68	86.19	81.24	72.92	56.14	28.64	72.61	58.52
MOS [33]	93.30	89.25	82.96	77.93	67.08	56.22	74.69	63.62
DAF (Ours)	**94.58**	**91.15**	**84.01**	**79.63**	**72.06**	**62.54**	**76.11**	**66.37**



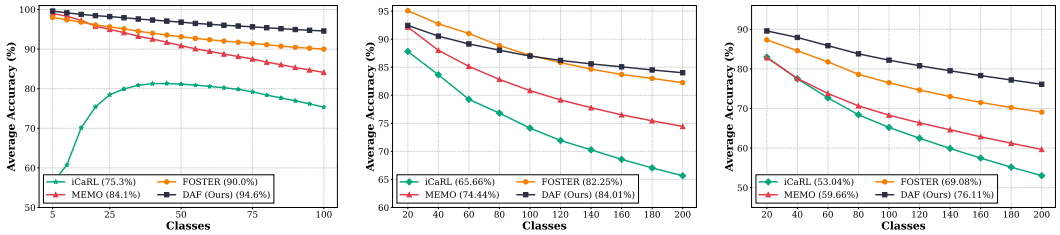
(a) CIFAR-100 (b) ImageNet-R (c) ImageNet-A (d) ObjectNet

Figure 3: Incremental accuracy trends on four benchmarks using the ViT-B/16-IN1K backbone. DAF (red curve) is compared against state-of-the-art exemplar-free methods. Across all datasets, DAF consistently demonstrates superior performance.

4.2 Main Results

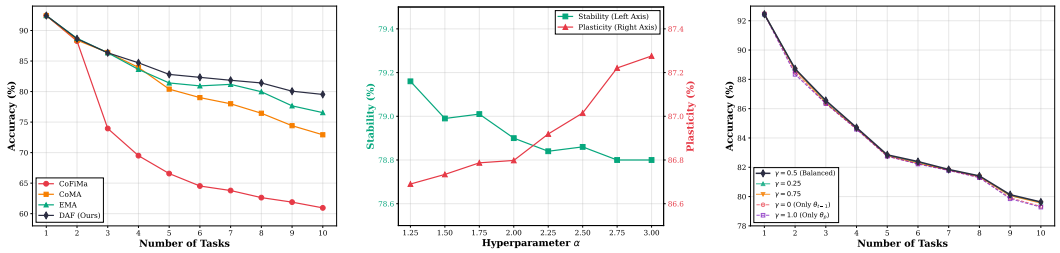
Comparison with SOTA Exemplar-Free Methods. We evaluate DAF against SOTA exemplar-free methods, with detailed results presented in Table 1. Across all four benchmarks, DAF achieves the highest Average Accuracy ($\bar{\mathcal{A}}$) and Final Accuracy ($\mathcal{A}_T$). For instance, on the challenging ObjectNet benchmark, DAF achieves a final accuracy of 66.37%, outperforming the leading retrieval-based method, MOS, by 2.75%. This result supports our central claim that constructing a theoretically grounded global adapter is a more effective strategy for mitigating interference than relying on adapter retrieval mechanisms. While retrieval-based methods like MOS are effective, their performance heavily depends on the accuracy of the selector, which can be unstable under domain shifts. DAF avoids this dependency by rigorously fusing historical knowledge into a single global adapter using PAC-Bayes bounds.

To further evaluate robustness with a weaker backbone, we conduct experiments using ViT-B/16-IN1K. Figure 3 illustrates the evolution of Average Accuracy ($\bar{\mathcal{A}}$) as the number of observed classes increases. Consistent with the IN21K results, DAF maintains a clear advantage throughout the incremental sequence across all benchmarks. Notably, on the difficult ObjectNet dataset, DAF exhibits a significantly slower decay in $\bar{\mathcal{A}}$ compared to the runner-up method MOS, underscoring its superior stability. These results confirm that the effectiveness of our dynamical fusion mechanism remains agnostic to the strength of the pre-trained backbone. Detailed computational efficiency comparisons with MOS and EASE are provided in Appendix K. Scalability experiments on 50-task and 100-task sequences and domain-incremental learning results are provided in Appendix L and Appendix M, respectively.



(a) CIFAR-100 (b) ImageNet-R (c) ObjectNet

Figure 4: Comparison with rehearsal-based methods. Rehearsal-based methods use 20 exemplars per class. Using ViT-B/16-IN21K as the backbone, DAF significantly outperforms strong replay baselines (iCaRL, MEMO, FOSTER) across CIFAR-100, ImageNet-R, and ObjectNet.



(a) Comparison of Fusion Methods (b) Sensitivity of Hyperparameter α (c) Impact of Prior Weight γ

Figure 5: Analysis of DAF on ImageNet-R using the ViT-B/16-IN21K backbone. **(a)** Performance comparison of different fusion mechanisms, demonstrating the superiority of DAF over EMA, CoMA, and CoFiMA baselines. **(b)** Dual-axis analysis of hyperparameter α, illustrating the explicit trade-off between Stability (Green) and Plasticity (Red). **(c)** Evaluation of the balancing weight γ, confirming that the theoretically derived equal weighting ($\gamma = 0.5$) of historical priors yields optimal performance.

Comparison with Strong Rehearsal-Based Methods. We also compare DAF against strong rehearsal-based methods, despite our method being exemplar-free. As shown in Figure 4, DAF surpasses these methods on CIFAR-100, ImageNet-R, and ObjectNet. On the 10-task ObjectNet sequence, DAF achieves a final average accuracy of 76.11%, establishing a significant lead of 7.03% over the state-of-the-art replay method, FOSTER. Notably, the performance gap widens as more tasks are learned. This suggests that our dynamical fusion method, which rigorously integrates historical priors, is a more scalable solution for lifelong learning than relying on the overhead of storing and replaying limited exemplars.

4.3 Ablation Study

To dissect the contribution of each key component in DAF, we conduct a systematic ablation study on ImageNet-R using the ViT-B/16-IN21K backbone. Table 2 presents results under four configurations, where the baseline uses random initialization and the last task adapter for inference.

As shown in Table 2, both Robust Initialization and Dynamical Fusion independently yield substantial gains over the baseline. Specifically, Robust Initialization alone improves $\mathcal{A}_T$ from 80.49% to 83.57% (+3.08%), while Dynamical Fusion alone achieves 83.48% (+2.99%). Crucially, these two components are complementary: their combination achieves the best performance of 84.01% / 79.63%, confirming that each module addresses a distinct aspect of the plasticity-stability dilemma and that their synergy is essential for optimal results.

4.4 Discussion

Comparison of Fusion Mechanisms. To validate the design of our fusion strategy, we compare DAF against three alternative mechanisms on ImageNet-R: Exponential Moving Average (EMA), CoMA, and CoFiMA [22]. As shown in Figure 5a, DAF achieves the best performance ($\mathcal{A}_T$=79.63%),

Table 2: Ablation study of DAF components on ImageNet-R (ViT-B/16-IN21K). ✓/× denotes whether the component is enabled. The best performance is highlighted in **bold**.

Robust Init.	Dyn. Fusion	$\bar{\mathcal{A}}$ (%)	$\mathcal{A}_T$ (%)
×	×	80.49	74.03
✓	×	83.57	77.90
×	✓	83.48	77.93
✓	✓	**84.01**	**79.63**

Table 3: Impact of Initialization on ImageNet-R using the ViT-B/16-IN21K backbone. We compare Random, Previous Task, and our Robust Initialization under two inference paradigms. The best performance is highlighted in **bold**.

	Paradigm 1: Last Task Adapter		Paradigm 2: Global Fused Adapter (DAF)	
Initialization Method	$\bar{\mathcal{A}}$ (%)	$\mathcal{A}_T$ (%)	$\bar{\mathcal{A}}$ (%)	$\mathcal{A}_T$ (%)
Random Initialization	80.50	74.03	78.50	71.52
Previous Task (θ_{t-1})	82.65	77.10	82.98	78.58
Robust Initialization (Ours)	83.57	77.90	**84.01**	**79.63**

outperforming EMA (76.55%) and CoMA (78.38%). Notably, CoFiMA performs significantly worse (60.95%) in our exemplar-free PEFT setting, as its recent-task-only Fisher estimation becomes highly biased without access to historical data. In contrast, DAF's fusion coefficients are rigorously derived via constrained Lagrangian optimization, explicitly accounting for initialization, task-specific, and global adapter parameters. Implementation details and a discussion of the performance gap are provided in Appendix O.

Hyperparameter Sensitivity. Figure 5b analyzes the impact of the constraint strength α. As α increases from 1.25 to 3.0, the relaxed historical constraint improves Plasticity but slightly reduces Stability. Despite this trade-off, overall performance remains highly robust (see Appendix H), indicating that DAF is not sensitive to precise tuning. We set $\alpha = 1.25$ to prioritize stability.

Balancing Prior Knowledge. Our derivation assigns equal weight to θ_p and θ^*_{t-1}. To validate this, we introduce a coefficient γ: $\theta^*_t = 2\gamma\beta_t\theta_p + 2(1-\gamma)\beta_t\theta^*_{t-1} + (1-2\beta_t)\theta_t$. As shown in Figure 5c, the balanced setting ($\gamma = 0.5$) achieves optimal accuracy (79.63%), while shifting to either extreme ($\gamma = 0$: 79.33%; $\gamma = 1$: 79.28%) degrades performance, confirming that θ_p and θ^*_{t-1} capture complementary aspects of historical knowledge.

Effectiveness of Robust Initialization. Table 3 compares initialization strategies under two inference paradigms. DAF with random initialization (71.52%) underperforms even the Last Task Adapter baseline (74.03%), as random parameters introduce destructive noise into the fusion process. Equipping DAF with Robust Initialization recovers this gap and achieves 79.63%, demonstrating that a high-quality prior θ_p is essential for constructive fusion.

5 Conclusion

To address the high inference latency and potential retrieval errors inherent in current Pre-Trained Model-based CIL methods, we propose Dynamical Adapter Fusion, a novel framework designed to construct a single robust global adapter. DAF formulates the parameter fusion process as a constrained optimization problem derived from PAC-Bayes theorem. By employing the Method of Lagrange Multipliers, we derive optimal weights to dynamically fuse task-specific and global parameters, ensuring an optimal balance between stability and plasticity without storing historical adapters. Furthermore, we propose a Robust Initialization strategy that leverages recursive historical knowledge to effectively capture global patterns. Extensive experiments on major CIL benchmarks demonstrate that DAF achieves superior accuracy over SOTA methods while achieving high training and inference efficiency. We discuss the limitations and future work in Appendix A, and broader impact in Appendix B.

References

[1] Rahaf Aljundi et al. "Gradient based sample selection for online continual learning". In: *Advances in neural information processing systems* 32 (2019).

[2] Rahaf Aljundi et al. "Memory aware synapses: Learning what (not) to forget". In: *Proceedings of the European conference on computer vision*. 2018, pp. 139–154.

[3] Pierre Alquier, James Ridgway, and Nicolas Chopin. "On the properties of variational approximations of Gibbs posteriors". In: *Journal of Machine Learning Research* 17.236 (2016), pp. 1–41.

[4] Andrei Barbu et al. "Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models". In: *NeurIPS* 32 (2019).

[5] Pietro Buzzega et al. "Dark experience for general continual learning: a strong, simple baseline". In: *Advances in neural information processing systems* 33 (2020), pp. 15920–15930.

[6] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. "Co2l: Contrastive continual learning". In: *Proceedings of the IEEE/CVF International conference on computer vision*. 2021, pp. 9516–9525.

[7] Arslan Chaudhry et al. "Efficient lifelong learning with a-gem". In: *International Conference on Learning Representations* (2019).

[8] Arslan Chaudhry et al. "Riemannian walk for incremental learning: Understanding forgetting and intransigence". In: *Proceedings of the European conference on computer vision*. 2018, pp. 532–547.

[9] Arslan Chaudhry et al. "Using hindsight to anchor past knowledge in continual learning". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 8. 2021, pp. 6993–7001.

[10] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *ICLR*. 2020.

[11] Sayna Ebrahimi et al. "Adversarial continual learning". In: *Proceedings of the European conference on computer vision*. Springer. 2020, pp. 386–402.

[12] Dan Hendrycks et al. "Natural adversarial examples". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 15262–15271.

[13] Dan Hendrycks et al. "The many faces of robustness: A critical analysis of out-of-distribution generalization". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 8340–8349.

[14] Tianyu Huai et al. "CL-MoE: Enhancing Multimodal Large Language Model with Dual Momentum Mixture-of-Experts for Continual Visual Question Answering". In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 19608–19617.

[15] Zixuan Ke, Bing Liu, and Xingchang Huang. "Continual learning of a mixed sequence of similar and dissimilar tasks". In: *Advances in neural information processing systems* 33 (2020), pp. 18493–18504.

[16] James Kirkpatrick et al. "Overcoming catastrophic forgetting in neural networks". In: *Proceedings of the national academy of sciences* 114.13 (2017), pp. 3521–3526.

[17] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009).

[18] Qiwei Li, Yuxin Peng, and Jiahuan Zhou. "Fcs: Feature calibration and separation for non-exemplar class incremental learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, pp. 28495–28504.

[19] Zhizhong Li and Derek Hoiem. "Learning without forgetting". In: *TPAMI* 40.12 (2017), pp. 2935–2947.

[20] Hangda Liu et al. "A resource-aware workload scheduling method for unbalanced GEMMs on GPUs". In: *The Computer Journal* (2024), bxae110.

[21] Noel Loo, Siddharth Swaroop, and Richard E Turner. "Generalized variational continual learning". In: *International Conference on Learning Representations* (2021).

[22] Imad Eddine Marouf et al. "Weighted ensemble models are strong continual learners". In: *European Conference on Computer Vision*. Springer. 2024, pp. 306–324.

[23] Michael McCloskey and Neal J Cohen. "Catastrophic interference in connectionist networks: The sequential learning problem". In: *Psychology of learning and motivation*. Vol. 24. Elsevier, 1989, pp. 109–165.

[24] Mark D McDonnell et al. "Ranpac: Random projections and pre-trained models for continual learning". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 12022–12053.

[25] Adam Paszke et al. "Pytorch: An imperative style, high-performance deep learning library". In: *NeurIPS*. 2019, pp. 8026–8037.

[26] Jingyang Qiao et al. “Large Continual Instruction Assistant”. In: *Forty-second International Conference on Machine Learning*. 2025.

[27] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. “Learning multiple visual domains with residual adapters”. In: *NeurIPS* 30 (2017).

[28] Sylvestre-Alvise Rebuffi et al. “icarl: Incremental classifier and representation learning”. In: *CVPR*. 2017, pp. 2001–2010.

[29] Joan Serra et al. “Overcoming catastrophic forgetting with hard attention to the task”. In: *International conference on machine learning*. PMLR. 2018, pp. 4548–4557.

[30] James Smith et al. “Always be dreaming: A new approach for data-free class-incremental learning”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 9374–9384.

[31] James Seale Smith et al. “CODA-Prompt: COntinual Decomposed Attention-based Prompting for Rehearsal-Free Continual Learning”. In: *CVPR*. 2023, pp. 11909–11919.

[32] Andrea Soltoggio et al. “A collective AI via lifelong learning and sharing at the edge”. In: *Nature Machine Intelligence* 6.3 (2024), pp. 251–264.

[33] Hai-Long Sun et al. “Mos: Model surgery for pre-trained model-based class-incremental learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 39. 19. 2025, pp. 20699–20707.

[34] Hai-Long Sun et al. “PILOT: A Pre-Trained Model-Based Continual Learning Toolbox”. In: *arXiv preprint arXiv:2309.07117* (2023).

[35] Yuwen Tan et al. “Semantically-shifted incremental adapter-tuning is a continual vitransformer”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 23252–23262.

[36] Hemanth Venkateswara et al. “Deep hashing network for unsupervised domain adaptation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 5018–5027.

[37] Xinrui Wang et al. “Forgetting, ignorance or myopia: Revisiting key challenges in online continual learning”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 58341–58375.

[38] Yan Wang, Da-Wei Zhou, and Han-Jia Ye. “Integrating Task-Specific and Universal Adapters for Pre-Trained Model-based Class-Incremental Learning”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2025, pp. 806–816.

[39] Fu-Yun Wang et al. “Foster: Feature boosting and compression for class-incremental learning”. In: *European conference on computer vision*. Springer. 2022, pp. 398–414.

[40] Zifeng Wang et al. “DualPrompt: Complementary Prompting for Rehearsal-free Continual Learning”. In: *ECCV* (2022).

[41] Zifeng Wang et al. “Learn-prune-share for lifelong learning”. In: *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2020, pp. 641–650.

[42] Zifeng Wang et al. “Learning to prompt for continual learning”. In: *CVPR*. 2022, pp. 139–149.

[43] Chuanguang Yang et al. “CLIP-KD: An Empirical Study of CLIP Model Distillation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 15952–15962.

[44] Lu Yu et al. “Semantic drift compensation for class-incremental learning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 6982–6991.

[45] Gengwei Zhang et al. “SLCA: Slow Learner with Classifier Alignment for Continual Learning on a Pre-trained Model”. In: *ICCV*. 2023.

[46] Tingting Zhao et al. “Deep bayesian unsupervised lifelong learning”. In: *Neural Networks* 149 (2022), pp. 95–106.

[47] Da-Wei Zhou et al. “A Model or 603 Exemplars: Towards Memory-Efficient Class-Incremental Learning”. In: *ICLR*. 2023.

[48] Da-Wei Zhou et al. “Expandable subspace ensemble for pre-trained model-based class-incremental learning”. In: *CVPR*. 2024, pp. 23554–23564.

[49] Da-Wei Zhou et al. “Revisiting Class-Incremental Learning with Pre-Trained Models: Generalizability and Adaptivity are All You Need”. In: *International Journal of Computer Vision* (2024).

[50] Yuhao Zhou et al. “Ferret: An Efficient Online Continual Learning Framework under Varying Memory Constraints”. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2025, pp. 4850–4861.

A Limitations and Future Work

Despite the strong performance and efficiency of DAF, we acknowledge certain limitations that offer avenues for future research.

First, similar to most standard Class-Incremental Learning (CIL) methods, our current framework relies on the availability of clear task boundaries during the training phase to perform the dynamical fusion. While this assumption holds for many structured learning scenarios, extending DAF to a fully Task-Free setting, where task boundaries are ambiguous or unknown, remains a challenging but promising direction.

Second, as a PEFT method, the upper bound of DAF's performance is naturally influenced by the representational power of the frozen pre-trained backbone. While we have verified efficacy on Vision Transformers, exploring how our dynamical fusion strategy generalizes to other architectures, such as larger Foundation Models, warrants further investigation.

Third, regarding our theoretical formulation, the derivation of the optimal fusion coefficient β_t relies on a second-order Taylor expansion of the loss function and approximates the Hessian diagonal using the Fisher Information Matrix. The claim of theoretical optimality therefore rests on the accuracy of these approximations. In scenarios with highly irregular loss landscapes where the local quadratic assumption may not hold, the derived coefficients could potentially deviate from the ideal solution, although our empirical results suggest robust performance in standard settings.

We plan to address these aspects in future work to enhance the versatility and autonomy of our CIL framework.

B Broader Impact

Our proposed DAF framework presents a significant advancement in Class-Incremental Learning by enabling efficient, lifelong knowledge accumulation without the burden of catastrophic forgetting. By eliminating the need for expanding architectures or inference-time retrieval, DAF drastically reduces computational and memory overhead. This efficiency is particularly relevant for real-world applications on resource-constrained edge devices, such as autonomous robots and mobile assistants, where low latency is critical.

Ethically, our method adheres to a strict exemplar-free protocol. By avoiding the storage of historical samples, DAF mitigates privacy risks associated with sensitive data retention, promoting the responsible deployment of AI in privacy-critical domains like healthcare. Furthermore, the high training and inference throughput of our method contributes to sustainable AI development by minimizing the energy footprint required for continuous model evolution.

C Prototype-based Classifier and Alignment

Following the established settings in MOS [33] and TUNA [38], we adopt a non-parametric strategy to construct the classification head. Let $\mathcal{D}_c$ denote the set of training samples available for a specific class c in the current task. We calculate the class prototype $\mathbf{p}_c$ by computing the centroid of the feature representations for all samples in this set:

$$\mathbf{p}_c = \frac{1}{|\mathcal{D}_c|} \sum_{\mathbf{x} \in \mathcal{D}_c} \phi(\mathbf{x}; \mathcal{A}_t), \tag{18}$$

where $\phi(\mathbf{x}; \mathcal{A}_t)$ represents the feature extraction function of the backbone model parameterized by the current task's adapter $\mathcal{A}_t$, and $|\mathcal{D}_c|$ is the number of samples in class c. The corresponding classifier weight vector $\mathbf{w}_c$ for class c is subsequently derived by applying L_2 normalization to the prototype: $\mathbf{w}_c = \mathbf{p}_c / \|\mathbf{p}_c\|_2$.

To address the challenge of *subspace misalignment*, where historical prototypes diverge from the current feature space, we utilize the feature distribution modeling method from TUNA [38] and MOS [33]. Specifically, we model the feature distribution of each previously learned class k as a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. During the training of the current task t, we synthesize pseudo-features by sampling from these stored distributions. These features are then

used to update the prototypes of old classes, resulting in a fully aligned global classifier matrix $W_t = [\mathbf{w}_1, \ldots, \mathbf{w}_{|\mathcal{Y}_t|}]$, where $|\mathcal{Y}_t|$ denotes the total number of accumulated classes.

D Theoretical Analysis for Task-Specific and Global Adapters

We first reintroduce the classical PAC-Bayes adapted from [3, 37] as the Lemma.

Lemma D.1 (Adapted from [3], Thm 4.1)**.** *Let $\mathcal{D} = (x_1, ..., x_m)$ be an iid set sampled from the law μ. For any data-free prior P, for any loss function ℓ bounded by K, any $\lambda > 0, \delta \in [0, 1]$, one has with probability $1 - \delta$ for any posterior $Q \in \mathcal{M}_1(\mathcal{H})$:*

$$\mathbb{E}_{h\sim Q}\mathbb{E}_{x\sim\mu}[\ell(h, x)] \leq \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}_{h\sim Q}[\ell(h, x_i)] + \frac{\mathrm{KL}(Q\|P) + \log(1/\delta)}{\lambda} + \frac{\lambda K^2}{2m}, \tag{19}$$

where $\mathcal{M}_1(\mathcal{H})$ denotes the set of all probability distributions on $\mathcal{H}$.

D.1 Setup and Definitions

To adapt the PAC-Bayes framework to our method, we formalize the structural decomposition of the parameter space and the distributions used in our algorithm.

Definition D.2 (Factorized Parameter Space and Distributions)**.** Let the hypothesis space be decomposable into $\mathcal{H} = \mathcal{H}_s \times \mathcal{H}_g$, where $\mathcal{H}_s$ represents the task-specific parameters (special adapter) and $\mathcal{H}_g$ represents the global parameters (global adapter). For a specific task t, we define the distributions as follows:

- **Prior** P_t: Constructed before observing $\mathcal{D}_t$, defined as the product measure of a fixed pre-trained initialization P_{init} and the previous global posterior Q^g_{t-1}:

$$P_t(h_s, h_g) = P_{\text{init}}(h_s) \otimes Q^g_{t-1}(h_g). \tag{20}$$

- **Posterior** Q_t: The variational family is restricted to product measures (Mean-Field approximation) of the learned task-specific adapter Q^s_t and the updated global adapter Q^g_t:

$$Q_t(h_s, h_g) = Q^s_t(h_s) \otimes Q^g_t(h_g). \tag{21}$$

Crucially, P_{init} is data-independent, and Q^g_{t-1} depends only on past tasks $\mathcal{D}_{1:t-1}$, making P_t a valid data-free prior relative to $\mathcal{D}_t$.

Based on this setup, we derive the generalization bound for the current task t.

Theorem D.3. *Let $\mathcal{D}_t = (x_1, ..., x_{m_t})$ be an iid set sampled from the distribution μ_t for task t. For any loss function ℓ bounded by K, any $\lambda > 0, \delta \in [0, 1]$, following the setting defined in Definition D.2, the following inequality holds with probability $1 - \delta$:*

$$\begin{aligned}\mathbb{E}_{h\sim Q_t}\left[\mathbb{E}_{x\sim\mu_t}[\ell(h, x)]\right] \leq & \frac{1}{m_t}\sum_{j=1}^{m_t}\mathbb{E}_{h\sim Q_t}\left[\ell(h, x_j)\right] + \frac{\mathrm{KL}(Q^s_t\|P_{init})}{\lambda} + \frac{\mathrm{KL}(Q^g_t\|Q^g_{t-1})}{\lambda} \\ & + \frac{\log(1/\delta)}{\lambda} + \frac{\lambda K^2}{2m_t}.\end{aligned} \tag{22}$$

Proof. We apply Lemma D.1 to the task t. Let $P = P_t$ and $Q = Q_t$. Since P_t is independent of the current samples $\mathcal{D}_t$, the standard PAC-Bayes bound holds with probability $1 - \delta$:

$$\mathbb{E}_{h\sim Q_t}\mathbb{E}_{x\sim\mu}[\ell(h, x)] \leq \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}_{h\sim Q_t}[\ell(h, x_i)] + \frac{\mathrm{KL}(Q_t\|P_t)}{\lambda} + \frac{\log(1/\delta)}{\lambda} + \frac{\lambda K^2}{2m_t}. \tag{23}$$

The key step is to decompose the divergence term $\mathrm{KL}(Q_t\|P_t)$. Following Definition D.2, both the prior and the posterior are product measures on the space $\mathcal{H}_s \times \mathcal{H}_g$. By the additivity property of the

Kullback-Leibler divergence for independent distributions, we have:

$$\begin{aligned}
\mathrm{KL}(Q_t \| P_t) &= \mathrm{KL}(Q_t^s \otimes Q_t^g \| P_{\text{init}} \otimes Q_{t-1}^g) \\
&= \int_{\mathcal{H}_s} \int_{\mathcal{H}_g} Q_t^s(h_s) Q_t^g(h_g) \log \left(\frac{Q_t^s(h_s) Q_t^g(h_g)}{P_{\text{init}}(h_s) Q_{t-1}^g(h_g)} \right) dh_g dh_s \\
&= \int_{\mathcal{H}_s} Q_t^s(h_s) \log \frac{Q_t^s(h_s)}{P_{\text{init}}(h_s)} dh_s \cdot \underbrace{\int_{\mathcal{H}_g} Q_t^g(h_g) dh_g}_{=1} \\
&\quad + \int_{\mathcal{H}_g} Q_t^g(h_g) \log \frac{Q_t^g(h_g)}{Q_{t-1}^g(h_g)} dh_g \cdot \underbrace{\int_{\mathcal{H}_s} Q_t^s(h_s) dh_s}_{=1} \\
&= \mathrm{KL}(Q_t^s \| P_{\text{init}}) + \mathrm{KL}(Q_t^g \| Q_{t-1}^g). \qquad (24)
\end{aligned}$$

Substituting this result back into Eq. (23) and separating the terms concludes the proof. □

Remark D.4. Theorem D.3 theoretically grounds the three objectives of our method to achieve a balance between plasticity and stability:

1. The empirical risk term $\frac{1}{m_t} \sum_{j=1}^{m_t} \mathbb{E}_{h \sim Q_t} [\ell(h, z_j)]$ encourages the adapter to learn from current data, ensuring the model's **plasticity**.
2. The term $\frac{\mathrm{KL}(Q_t^s \| P_{\text{init}})}{\lambda}$ regularizes the task-specific adapter towards the robust pre-trained initialization. This not only prevents overfitting but also helps align the current model distribution with potential future distributions, contributing to the model's **stability**.
3. The term $\frac{\mathrm{KL}(Q_t^g \| Q_{t-1}^g)}{\lambda}$ constrains the global adapter to remain close to the knowledge acquired from previous tasks. This explicitly ensures **stability** by mitigating catastrophic forgetting.

E Proof of Relationship between θ_t, θ_t^* and $\Delta\theta$

From the definition of the stability constraint parameter $\Delta\theta$ in Eq. (4) and the fusion strategy in Eq. (3), we first expand $\Delta\theta$:

$$\Delta\theta = \theta_t^* - \theta_{t-1}^* + \theta_t - \theta_p. \qquad (25)$$

Substitute θ_t^* with the explicit fusion formula $\beta_t\theta_p + \beta_t\theta_{t-1}^* + (1 - 2\beta_t)\,\theta_t$:

$$\begin{aligned}
\Delta\theta &= \left[\beta_t\theta_p + \beta_t\theta_{t-1}^* + (1-2\beta_t)\theta_t\right] - \theta_{t-1}^* + \theta_t - \theta_p \\
&= (\beta_t - 1)\theta_p + (\beta_t - 1)\theta_{t-1}^* + (1 - 2\beta_t + 1)\theta_t \qquad (26) \\
&= (\beta_t - 1)\theta_p + (\beta_t - 1)\theta_{t-1}^* + (2 - 2\beta_t)\theta_t.
\end{aligned}$$

By factoring out $(\beta_t - 1)$, we obtain:

$$\Delta\theta = (\beta_t - 1)(\theta_p + \theta_{t-1}^* - 2\theta_t). \qquad (27)$$

Next, we calculate the difference between the updated global adapter and the task-specific adapter, $\theta_t^* - \theta_t$:

$$\theta_t^* - \theta_t = \left[\beta_t\theta_p + \beta_t\theta_{t-1}^* + (1-2\beta_t)\theta_t\right] - \theta_t. \qquad (28)$$

Simplify the equation by grouping terms:

$$\begin{aligned}
\theta_t^* - \theta_t &= \beta_t\theta_p + \beta_t\theta_{t-1}^* + (1 - 2\beta_t - 1)\theta_t \\
&= \beta_t\theta_p + \beta_t\theta_{t-1}^* - 2\beta_t\theta_t. \qquad (29)
\end{aligned}$$

Factor out β_t:

$$\theta_t^* - \theta_t = \beta_t(\theta_p + \theta_{t-1}^* - 2\theta_t). \qquad (30)$$

Finally, comparing Eq. (27) and Eq. (30), we substitute the common term $(\theta_p + \theta_{t-1}^* - 2\theta_t) = \frac{\Delta\theta}{\beta_t - 1}$ into Eq. (30):

$$\theta_t^* - \theta_t = \beta_t \left(\frac{\Delta\theta}{\beta_t - 1} \right) = \frac{\beta_t}{\beta_t - 1} \Delta\theta. \qquad (31)$$

F Detailed Deduction of the Optimal Fusion Coefficient β_t

In this section, we provide the step-by-step deduction of the optimal fusion coefficient β_t presented in Eq. (12). We start with the unified optimization objective defined in Eq. (5), which aims to minimize the performance gap and parameter shift under the stability constraint.

The Lagrangian function $\mathcal{J}$ is formulated as:

$$\begin{aligned}\mathcal{J} = &\underbrace{\mathcal{L}(\theta_t^*) - \mathcal{L}(\theta_t)}_{\text{Loss Difference}} + \underbrace{\frac{1}{2}\left(\theta_t^* - \theta_{t-1}^* + \theta_t - \theta_p\right)^2}_{\text{Parameter Shift}} \\ &+ \lambda\left(\Delta\theta + \theta_{t-1}^* + \theta_p - \theta_t - \theta_t^*\right).\end{aligned} \tag{32}$$

First, we apply the second-order Taylor expansion to the loss difference and substitute the definitions derived in Appendix E. Specifically, we use the relationship $\theta_t^* - \theta_t = \frac{\beta_t}{\beta_t - 1}\Delta\theta$. The Lagrangian can be rewritten in terms of β_t and $\Delta\theta$:

$$\begin{aligned}\mathcal{J} \approx &\mathcal{L}'(\theta_t)\left(\frac{\beta_t}{\beta_t - 1}\Delta\theta\right) + \frac{\mathcal{L}''(\theta_t)}{2}\left(\frac{\beta_t}{\beta_t - 1}\Delta\theta\right)^2 + \frac{1}{2}\Delta\theta^2 \\ &+ \lambda\left[\Delta\theta + (1-\beta_t)(\theta_p + \theta_{t-1}^* - 2\theta_t)\right].\end{aligned} \tag{33}$$

To find the optimal β_t, we calculate the partial derivatives of $\mathcal{J}$ with respect to β_t and $\Delta\theta$, and set them to zero.

1. Derivative with respect to β_t:

$$\begin{aligned}\frac{\partial\mathcal{J}}{\partial\beta_t} = &-\frac{1}{(\beta_t - 1)^2}\mathcal{L}'(\theta_t)\Delta\theta - \frac{\beta_t}{(\beta_t - 1)^3}\mathcal{L}''(\theta_t)\Delta\theta^2 \\ &- \lambda(\theta_p + \theta_{t-1}^* - 2\theta_t) = 0.\end{aligned} \tag{34}$$

2. Derivative with respect to $\Delta\theta$:

$$\frac{\partial\mathcal{J}}{\partial\Delta\theta} = \frac{\beta_t}{\beta_t - 1}\mathcal{L}'(\theta_t) + \frac{\beta_t^2}{(\beta_t - 1)^2}\mathcal{L}''(\theta_t)\Delta\theta + \Delta\theta + \lambda = 0. \tag{35}$$

From the constraint equation $\Delta\theta = (\beta_t - 1)(\theta_p + \theta_{t-1}^* - 2\theta_t)$, we can express the term involving parameters as:

$$\theta_p + \theta_{t-1}^* - 2\theta_t = \frac{1}{\beta_t - 1}\Delta\theta. \tag{36}$$

Now, we substitute λ derived from Eq. (35) and the relationship from Eq. (36) into Eq. (34). This eliminates λ:

$$\begin{aligned}0 = &-\frac{1}{(\beta_t - 1)^2}\mathcal{L}'(\theta_t)\Delta\theta - \frac{\beta_t}{(\beta_t - 1)^3}\mathcal{L}''(\theta_t)\Delta\theta^2 \\ &- \left[-\frac{\beta_t}{\beta_t - 1}\mathcal{L}'(\theta_t) - \frac{\beta_t^2}{(\beta_t - 1)^2}\mathcal{L}''(\theta_t)\Delta\theta - \Delta\theta\right]\left(\frac{1}{\beta_t - 1}\Delta\theta\right).\end{aligned} \tag{37}$$

Expanding and grouping the terms:

$$\begin{aligned}0 = &-\frac{1}{(\beta_t - 1)^2}\mathcal{L}'(\theta_t)\Delta\theta - \frac{\beta_t}{(\beta_t - 1)^3}\mathcal{L}''(\theta_t)\Delta\theta^2 \\ &+ \frac{\beta_t}{(\beta_t - 1)^2}\mathcal{L}'(\theta_t)\Delta\theta + \frac{\beta_t^2}{(\beta_t - 1)^3}\mathcal{L}''(\theta_t)\Delta\theta^2 + \frac{1}{\beta_t - 1}\Delta\theta^2.\end{aligned} \tag{38}$$

We can combine the coefficients for $\mathcal{L}'(\theta_t)$ and $\mathcal{L}''(\theta_t)$:

$$\begin{aligned}0 = &\left[\frac{\beta_t - 1}{(\beta_t - 1)^2}\right]\mathcal{L}'(\theta_t)\Delta\theta + \left[\frac{\beta_t^2 - \beta_t}{(\beta_t - 1)^3}\right]\mathcal{L}''(\theta_t)\Delta\theta^2 \\ &+ \frac{1}{\beta_t - 1}\Delta\theta^2.\end{aligned} \tag{39}$$

Simplifying the fractions, we obtain:

$$0 = \frac{\Delta\theta}{\beta_t - 1}\left[\mathcal{L}'(\theta_t) + \frac{\beta_t}{\beta_t - 1}\mathcal{L}''(\theta_t)\Delta\theta + \Delta\theta\right]. \tag{40}$$

By observation, we can find one trivial solution where $\Delta\theta = 0$. This implies $\theta_t^* - \theta_{t-1}^* + \theta_t - \theta_p = 0$, representing that the model has not acquired new knowledge or shifted its parameters effectively. Obviously, this is not the global optimal solution we seek.

Furthermore, since the operational range of β_t is constrained to $(0, 0.5)$, we have $\beta_t - 1 \neq 0$. Consequently, to find the non-trivial optimal solution, we can safely divide the equation by $\frac{\Delta\theta}{\beta_t - 1}$:

$$0 = \mathcal{L}'(\theta_t) + \frac{\beta_t}{\beta_t - 1}\mathcal{L}''(\theta_t)\Delta\theta + \Delta\theta. \tag{41}$$

Next, we substitute $\Delta\theta$ back using the relationship $\Delta\theta = (\beta_t - 1)(\theta_p + \theta_{t-1}^* - 2\theta_t)$:

$$\begin{aligned} 0 = \mathcal{L}'(\theta_t) &+ \frac{\beta_t}{\beta_t - 1}\mathcal{L}''(\theta_t)\left[(\beta_t - 1)(\theta_p + \theta_{t-1}^* - 2\theta_t)\right] \\ &+ (\beta_t - 1)(\theta_p + \theta_{t-1}^* - 2\theta_t). \end{aligned} \tag{42}$$

Simplifying further:

$$\begin{aligned} 0 = \mathcal{L}'(\theta_t) &+ \beta_t\mathcal{L}''(\theta_t)(\theta_p + \theta_{t-1}^* - 2\theta_t) \\ &+ (\beta_t - 1)(\theta_p + \theta_{t-1}^* - 2\theta_t). \end{aligned} \tag{43}$$

Rearranging terms to isolate β_t:

$$\begin{aligned} 0 = \mathcal{L}'(\theta_t) &- (\theta_p + \theta_{t-1}^* - 2\theta_t) \\ &+ \beta_t(\theta_p + \theta_{t-1}^* - 2\theta_t)\left[\mathcal{L}''(\theta_t) + 1\right]. \end{aligned} \tag{44}$$

Finally, solving for β_t, we obtain the optimal fusion coefficient:

$$\beta_t = \frac{(\theta_p + \theta_{t-1}^* - 2\theta_t) - \mathcal{L}'(\theta_t)}{(\theta_p + \theta_{t-1}^* - 2\theta_t)(\mathcal{L}''(\theta_t) + 1)}. \tag{45}$$

F.1 Verification of Constraint Satisfaction

According to Eq. (27) derived in Appendix E, we have established the relationship:

$$\Delta\theta = (\beta_t - 1)(\theta_p + \theta_{t-1}^* - 2\theta_t). \tag{46}$$

Thus, we can verify that our proposed solution satisfies the *s.t.* constraint (Eq. (4)) through the following derivation:

$$\begin{aligned} \Delta\theta + \theta_{t-1}^* + \theta_p - \theta_t - \theta_t^* &= (\beta_t - 1)(\theta_p + \theta_{t-1}^* - 2\theta_t) + \theta_{t-1}^* + \theta_p - \theta_t - \theta_t^* \\ &= \beta_t\theta_p + \beta_t\theta_{t-1}^* - 2\beta_t\theta_t - \theta_p - \theta_{t-1}^* + 2\theta_t + \theta_{t-1}^* + \theta_p - \theta_t - \theta_t^* \\ &= \beta_t\theta_p + \beta_t\theta_{t-1}^* - 2\beta_t\theta_t + \theta_t - \theta_t^* \\ &= \beta_t\theta_p + \beta_t\theta_{t-1}^* - 2\beta_t\theta_t + \theta_t - [\beta_t\theta_p + \beta_t\theta_{t-1}^* + (1 - 2\beta_t)\theta_t] \\ &= \beta_t\theta_p + \beta_t\theta_{t-1}^* - 2\beta_t\theta_t + \theta_t - \beta_t\theta_p - \beta_t\theta_{t-1}^* - \theta_t + 2\beta_t\theta_t \\ &= 0. \end{aligned} \tag{47}$$

This confirms that the fusion mechanism strictly adheres to the stability constraint defined in our optimization problem.

G Full Implementation Details

Datasets and Task Protocols. We follow the standard evaluation protocol of [49]. We evaluate on four benchmarks: CIFAR-100 [17] is split into 20 tasks with 5 classes each; ImageNet-R [13],

ImageNet-A [12], and ObjectNet [4] each contain 200 classes split into 10 tasks of 20 classes. Class order is shuffled with a fixed random seed of 1993 [28].

Training Details. All experiments are implemented in PyTorch [25] using the PILOT codebase [34] on NVIDIA GeForce RTX 2080 Ti GPUs. We use two backbones: ViT-B/16-IN21K (pre-trained on ImageNet-21K) and ViT-B/16-IN1K (pre-trained on ImageNet-1K). Each task is trained for 20 epochs with batch size 48, using SGD with momentum, initial learning rate 0.01, and cosine annealing decay. Adapter projection dimension is set to 16 for all adapter-based methods. The scaling factor α in Eq. (14) is set to 1.25.

Evaluation Metrics. We report Average Accuracy $\bar{\mathcal{A}} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{t}\sum_{j=1}^{t} A_{t,j}$ and Final Average Accuracy $\mathcal{A}_T = \frac{1}{T}\sum_{j=1}^{T} A_{T,j}$, where $A_{t,j}$ is the accuracy on task j after training on task t. We additionally report Stability $S = \frac{1}{T-1}\sum_{j=1}^{T-1} A_{T,j}$ and Plasticity $P = \frac{1}{T}\sum_{j=1}^{T} A_{j,j}$ to analyze the plasticity-stability trade-off.

H Extended Hyperparameter Analysis

In the main paper, we discussed the trade-off between Stability and Plasticity with varying constraint strengths α. Here, we provide a detailed visualization of the overall **Average Accuracy** ($\bar{\mathcal{A}}$) to further substantiate the robustness of DAF.

As illustrated in Figure 6a, increasing α from 1.25 to 3.0 results in negligible fluctuations in $\bar{\mathcal{A}}$, with values ranging tightly between 79.63% and 79.40%. This stability confirms that while α shifts the internal balance between remembering old tasks and learning new ones, the aggregate performance of the global adapter remains remarkably consistent. Consequently, DAF demonstrates strong resilience to hyperparameter variations, alleviating the need for meticulous tuning in practical deployments.

In DAF, the Fisher approximation is only used in the second-order estimate $\mathcal{L}''(\theta_t) \approx \alpha\frac{F(\theta_t)-F_{\min}}{F-F_{\min}}+1$ in Eq. (13), where α controls the weight of the Fisher term in β_t. Therefore, varying α also evaluates the sensitivity of DAF to the Fisher approximation. In Table 4, we further compare DAF with $\alpha = 0$ (which removes the Fisher term and uses a fixed coefficient) and a variant that replaces the Fisher approximation with the exact Hessian on ImageNet-R. Across $\alpha \in [1.25, 3.0]$, $\bar{\mathcal{A}}$ varies by only 0.23% (from 79.40% to 79.63%). Replacing the Fisher approximation with the exact Hessian changes $\bar{\mathcal{A}}$ by only 0.07% (79.70% vs. 79.63%), which shows that our Fisher approximation is close to the exact Hessian. When $\alpha = 0$, $\bar{\mathcal{A}}$ drops by 1.03% to 78.60%, showing that the Fisher term improves $\bar{\mathcal{A}}$ while DAF is not sensitive to the approximation error between the Fisher approximation and the exact Hessian.

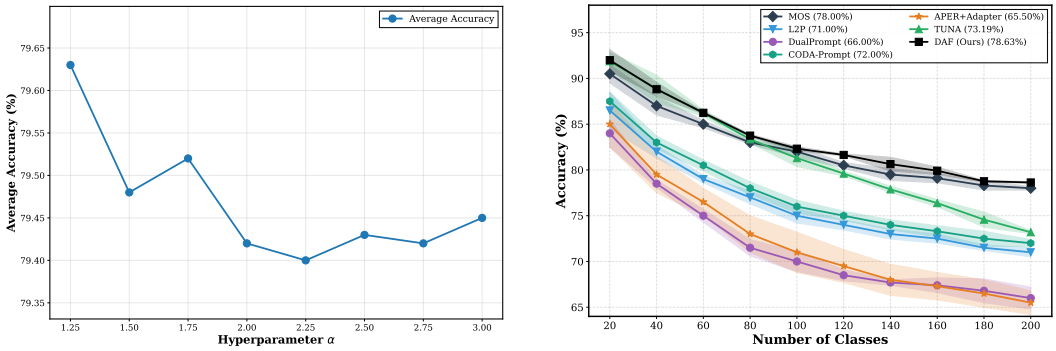
Table 4: Sensitivity of $\bar{\mathcal{A}}$ (%) to α and the Fisher approximation on ImageNet-R.

Setting	$\alpha = 0$	$\alpha = 1.25$	$\alpha = 1.5$	$\alpha = 1.75$	$\alpha = 2.0$	$\alpha = 2.25$	$\alpha = 2.5$	$\alpha = 2.75$	$\alpha = 3.0$	**Exact Hessian**
$\bar{\mathcal{A}}$ (%)	78.60	**79.63**	79.48	79.52	79.42	79.40	79.43	79.42	79.45	79.70

Plasticity-Stability Trade-off on CIFAR-100. To further illustrate the role of α in controlling the plasticity-stability trade-off, Table 5 reports the per-task $\bar{\mathcal{A}}$ on CIFAR-100 for $\alpha = 1.25$ and $\alpha = 3.0$. A higher α relaxes the historical constraint, yielding marginally better early-task accuracy (plasticity) but slightly degraded final performance (stability). The performance difference is at most 0.07%, confirming that DAF is robust to α variations. We fix $\alpha = 1.25$ to prioritize long-term stability.

Table 5: Impact of α on per-task $\bar{\mathcal{A}}$ (%) on CIFAR-100.

Setting	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	**T10**	**T11**	**T12**	**T13**	**T14**	**T15**	**T16**	**T17**	**T18**	**T19**	**T20**
$\alpha = 1.25$	99.60	99.15	98.77	98.45	98.21	97.92	97.60	97.34	97.08	96.80	96.50	96.27	96.03	95.83	95.62	95.39	95.16	94.94	94.76	**94.58**
$\alpha = 3.0$	99.60	99.20	98.84	98.49	98.25	97.96	97.63	97.35	97.08	96.81	96.49	96.24	96.00	95.79	95.58	95.35	95.12	94.90	94.71	94.53
Δ	+0.00	+0.05	+0.07	+0.04	+0.04	+0.04	+0.03	+0.01	+0.00	+0.01	−0.01	−0.03	−0.03	−0.04	−0.04	−0.04	−0.04	−0.04	−0.05	−0.05



(a) Robustness to Hyperparameter α

(b) Stability Across Random Seeds

Figure 6: Extended stability analysis of DAF on ImageNet-R. **(a)** Evaluation of the Final Average Accuracy ($\bar{\mathcal{A}}$) as the constraint strength α varies from 1.25 to 3.0. The performance remains remarkably stable, validating that DAF is insensitive to precise hyperparameter tuning. **(b)** Mean Average Accuracy curves over five different random seeds. DAF consistently outperforms SOTA methods, demonstrating strong robustness to variations in class order.

I Performance Stability Across Multiple Runs

While our main experiments adhere to the standard protocol of [28] using a fixed seed (1993), real-world scenarios often present unpredictable task sequences. To rigorously verify the consistency of DAF, we extend our evaluation on ImageNet-R by conducting five independent runs with distinct random seeds: {1993, 1994, 1995, 1996, 1997}. Each seed generates a unique permutation of class orders, creating diverse task compositions.

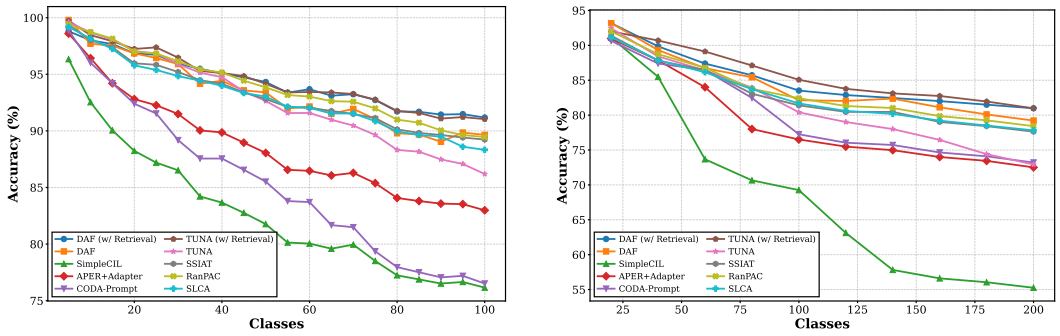
The averaged performance across these runs is presented in Figure 6b. The results demonstrate that DAF maintains a consistent and significant advantage over all competing baselines regardless of the task sequence. This stability confirms that our dynamical fusion mechanism is robust to variations in the incremental learning curriculum and does not rely on specific favorable class orderings.

J Enhancing DAF with Task-Specific Retrieval

While DAF is designed as an efficient, retrieval-free framework utilizing a Single Global Adapter, its flexible architecture allows it to be integrated with task-aware mechanisms. Although we evaluate TUNA using only its global adapter in the main paper, the original TUNA [38] also introduces a combined inference strategy. To fully evaluate extensibility, we adopt this strategy: retrieving the most relevant task-specific adapter via an entropy-based selector and combining its prediction with the global adapter. We compare four distinct variants: standard **DAF** and **TUNA** (Global adapter only), alongside their retrieval-enhanced counterparts, **DAF (w/ Retrieval)** and **TUNA (w/ Retrieval)**.

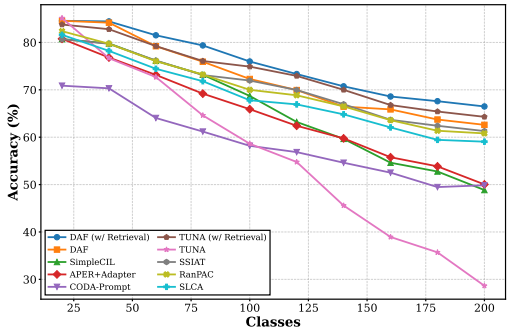
As illustrated in Figure 7, incorporating a retrieval module consistently yields significant performance improvements for both frameworks across all benchmarks. Specifically, **DAF (w/ Retrieval)** achieves the highest accuracy, outperforming **TUNA (w/ Retrieval)** by a clear margin. This confirms that our dynamically fused global adapter serves as a superior foundation for ensemble-based inference compared to TUNA.

However, it is crucial to acknowledge the cost of this improvement. While retrieval mechanisms effectively boost accuracy, they introduce substantial computational overhead during inference, as the model must search through a growing pool of adapters for every input sample. Consequently, although **DAF (w/ Retrieval)** offers the best absolute performance, the standard **DAF** remains the optimal choice for real-world applications where inference speed and memory efficiency are paramount.

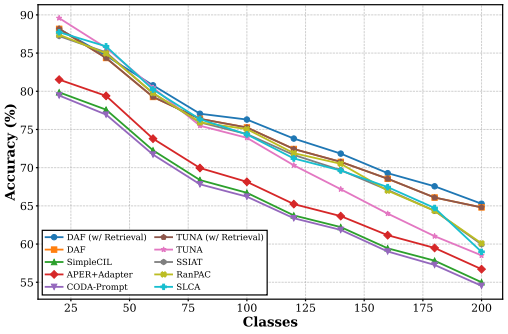


(a) CIFAR-100

(b) ImageNet-R



(c) ImageNet-A



(d) ObjectNet

Figure 7: **Impact of Task-Specific Retrieval on DAF and TUNA.** We compare four variants: DAF and TUNA with and without retrieval modules. While integrating retrieval mechanisms significantly boosts accuracy for both methods, it comes at the expense of high inference latency. Notably, **DAF (w/ Retrieval)** consistently outperforms **TUNA (w/ Retrieval)**, demonstrating the superiority of our underlying fusion mechanism.

K Computational Complexity and Efficiency Analysis

DAF Component Overhead. Table 6 details the computational time breakdown of each DAF component. The primary overhead is the computation of the Fisher Information Matrix and gradients, which requires one additional forward-backward pass over the current task data. In practice, this cost is negligible compared to the overall training time (less than 5% on all benchmarks). The Robust Initialization update and the parameter fusion step incur virtually no overhead.

Table 6: Computational time breakdown of DAF components (in seconds).

Dataset	Fisher & Grad.	Robust Init.	Fusion	DAF Train	DAF Infer
CIFAR-100	386.17	0.41	0.40	8545.03	486.89
ImageNet-A	39.48	0.10	0.18	1647.00	45.57
ImageNet-R	157.28	0.18	0.18	3064.82	151.32
ObjectNet	420.50	0.12	0.20	11420.10	862.79

Comparison with MOS and EASE. Unlike retrieval-based (MOS) or aggregation-based (EASE) methods that require $\mathcal{O}(T)$ parameter expansion, DAF maintains a strict $\mathcal{O}(1)$ parameter footprint by dynamically merging all knowledge into a single global adapter. Table 7 compares training and inference times across methods. DAF achieves the most efficient training and significantly faster inference across all datasets, as it completely bypasses the heavy inference-time computations of routing or aggregation.

Table 7: Comparison of training and inference time (seconds) and extra parameter overhead across methods.

Dataset	Metric	MOS	EASE	DAF (Ours)
CIFAR-100	Extra Params (MB)	23.22	23.22	**1.16**
	Train (s)	8612.45	8750.32	**8545.03**
	Infer (s)	6837.58	6890.15	**486.89**
ImageNet-R	Extra Params (MB)	11.61	11.61	**1.16**
	Train (s)	3156.78	3240.55	**3064.82**
	Infer (s)	1145.12	1172.34	**151.32**
ImageNet-A	Extra Params (MB)	11.61	11.61	**1.16**
	Train (s)	1794.53	1845.21	**1647.00**
	Infer (s)	408.98	425.60	**45.57**
ObjectNet	Extra Params (MB)	11.61	11.61	**1.16**
	Train (s)	11561.07	11785.40	**11420.10**
	Infer (s)	2342.12	2380.55	**862.79**

L Scalability on Long Task Sequences

To evaluate DAF's scalability under extended task sequences, we conduct experiments on 50-task and 100-task splits of ImageNet-A. Table 8 presents the results for DAF and TUNA, both with and without retrieval augmentation.

Table 8: Performance on ImageNet-A under 50-task and 100-task sequence settings.

Method	50 Tasks		100 Tasks	
	$\bar{\mathcal{A}}$ (%)	$\mathcal{A}_T$ (%)	$\bar{\mathcal{A}}$ (%)	$\mathcal{A}_T$ (%)
TUNA	60.42	47.33	64.42	52.40
TUNA (w/ Retrieval)	70.51	60.43	70.49	60.37
DAF (Ours)	69.87	57.80	70.13	57.95
DAF (w/ Retrieval)	**71.08**	**61.75**	**71.28**	**61.49**

DAF's single global adapter remains robust across extended sequences, consistently outperforming TUNA. Notably, retrieval augmentation yields only marginal gains for DAF (compared to the significant boost for TUNA), indicating that our dynamically fused adapter already retains knowledge effectively without external reliance. The retrieval-augmented variant of DAF is described in Appendix J.

M Domain-Incremental Learning Experiments

While DAF is primarily designed for Class-Incremental Learning, we evaluate its transferability to Domain-Incremental Learning (DIL) using the Office-Home dataset [36], which contains four domains (Art, Clipart, Product, Real World) with significant visual style shifts.

Table 9: Performance under domain-incremental learning on Office-Home.

Method	$\bar{\mathcal{A}}$ (%)	$\mathcal{A}_T$ (%)
TUNA	75.36	60.21
DAF (w/o Robust Init.)	76.38	62.19
DAF (Ours)	**80.61**	**69.93**

DAF outperforms TUNA by a significant margin (+5.25% in $\bar{\mathcal{A}}$). Crucially, removing Robust Initialization causes substantial degradation, highlighting its importance in bridging domain gaps.

We hypothesize that since the label space remains identical across DIL tasks, sequentially optimized models lie in the same loss basin, allowing our historical average to serve as an effective prior across varying domains.

N Sensitivity Analysis of Clipping Bounds for β_t

The fusion coefficient β_t is clipped to $[0.001, 0.499]$ in practice to ensure numerical stability and maintain valid fusion weights (see Section 3.4). To verify the robustness of this choice, we evaluate DAF under different clipping ranges on ImageNet-R.

Table 10: Sensitivity of DAF to different clipping bounds for β_t on ImageNet-R.

Clipping Bounds	$\bar{\mathcal{A}}$ (%)	$\mathcal{A}_T$ (%)
[0.001, 0.499]	**84.01**	**79.63**
[0.01, 0.49]	**84.01**	**79.63**
[0.1, 0.4]	**84.01**	**79.63**

Results are strictly identical across all valid clipping ranges, demonstrating that DAF is highly robust to the specific choice of clipping bounds. This stability arises because the theoretically derived β_t values naturally fall well within the valid range $(0, 0.5)$ for standard CIL benchmarks.

O Implementation Details of Fusion Method Comparison

Fusion Baselines. To provide a fair comparison in Figure 5a, we implement the following fusion baselines within our PEFT framework, all evaluated under the same exemplar-free PILOT benchmark setting with identical hyperparameters:

- **EMA**: The global adapter is updated as an exponential moving average of the task-specific adapter: $\theta_t^* = (1-\mu)\theta_{t-1}^* + \mu\theta_t$, with $\mu = 0.1$.
- **CoMA** [22]: Similar to EMA but with equal weighting between the current and previous global adapter: $\theta_t^* = (1-\mu)\theta_{t-1}^* + \mu\theta_t$, with $\mu = 0.5$.
- **CoFiMA** [22]: A Fisher-weighted ensemble where each adapter's contribution is weighted by its Fisher Information. For each task t, the Fisher matrix of the previous global adapter is estimated using the data from the most recent completed task $\mathcal{D}_{t-1}$, while the Fisher matrix of the current task adapter is estimated using the current task data $\mathcal{D}_t$.

Why CoFiMA Underperforms in Our Setting. CoFiMA is also an exemplar-free method; however, it was originally designed and evaluated under a full fine-tuning paradigm. When adapted to the restricted and sparse parameter space of PEFT adapters, its Fisher-weighted fusion suffers from high estimation bias. Specifically, the Fisher matrix estimated from a single task's data is a poor proxy for the true curvature of the global loss landscape, as it captures only the local geometry of the most recently seen distribution. This leads to severely biased fusion weights that over-emphasize the current task adapter and under-weight historical knowledge, resulting in a substantial 18.68% drop in $\bar{\mathcal{A}}$ compared to DAF (60.95% vs. 79.63% on ImageNet-R).

Furthermore, even when compared directly against the highest full fine-tuning results reported in the original CoFiMA paper under the same pre-trained model (ViT-B/16-IN21K) and dataset split, our DAF still achieves a 2.53% higher $\bar{\mathcal{A}}$ (84.01% vs. 81.48%) while updating substantially fewer parameters. This demonstrates that DAF's principled PAC-Bayesian derivation, which explicitly accounts for generalization regularization (via θ_p) and forgetting regularization (via θ_{t-1}^*), provides a fundamentally more robust fusion mechanism than heuristic Fisher weighting.

P Analysis of Forgetting and Backward Transfer

To further evaluate how well DAF retains knowledge from previous tasks during continual adaptation, we measure Forgetting and Backward Transfer (BWT) and compare DAF with TUNA under identical

experimental settings across four benchmarks: CIFAR-100, ImageNet-R, ImageNet-A, and ObjectNet. Lower Forgetting (↓) indicates less performance drop on previously learned tasks, while higher BWT (↑) indicates better preservation of past task performance after learning new tasks.

Table 11: Comparison of Forgetting (↓) and Backward Transfer (BWT ↑) between TUNA and DAF across four benchmarks under identical settings. Best results are in **bold**.

Dataset	Method	Forgetting ↓	BWT ↑
CIFAR-100	TUNA	8.5158	-8.0842
	DAF (Ours)	**4.5263**	**-4.4526**
ImageNet-R	TUNA	10.6733	-10.6233
	DAF (Ours)	**7.8922**	**-7.8922**
ImageNet-A	TUNA	31.6144	-29.5389
	DAF (Ours)	**20.8933**	**-20.8933**
ObjectNet	TUNA	14.2600	-13.5544
	DAF (Ours)	**8.1744**	**-8.0400**

As shown in Table 11, DAF consistently achieves lower Forgetting and higher BWT than TUNA across all four datasets. Specifically, DAF reduces Forgetting by nearly half on CIFAR-100 (from 8.5158 to 4.5263) and ObjectNet (from 14.2600 to 8.1744), with corresponding improvements in BWT (from -8.0842 to -4.4526 on CIFAR-100, and from -13.5544 to -8.0400 on ObjectNet). Clear improvements also hold on more challenging benchmarks, where Forgetting drops from 10.6733 to 7.8922 on ImageNet-R and from 31.6144 to 20.8933 on ImageNet-A. These results are consistent with our optimization objective in Eq. (3), where the Forgetting Regularization term keeps the updated fused parameters θ_t^* close to the previous fused parameters θ_{t-1}^*, thereby reducing forgetting on earlier tasks.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the three contributions (PAC-Bayes-grounded fusion, Robust Initialization, SOTA performance), all supported by Section 3.2, Section 3, and Section 4.

   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: Limitations are discussed in Appendix A, covering task-boundary assumptions, backbone dependency, and the approximation accuracy of the Taylor expansion and FIM.

   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: Full proofs for Theorem 3.1 and the derivation of β_t are provided in Appendix D, E, and F.

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: Full implementation details (datasets, optimizer, hyperparameters) are in Appendix G.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We provided the code in the supplementary materials.

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: Key settings are summarized in Section 4; full details are in Appendix G. Sensitivity analyses for α and β_t clipping are in Appendix H and N.

   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: [No]

   Justification: Following standard CIL practice [49, 33], results use a fixed seed. Robustness across five seeds is shown in Appendix I.

   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: GPU type (NVIDIA GeForce RTX 2080 Ti) is stated in Appendix G. Training and inference times are detailed in Appendix K.

   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: This work uses only publicly available image classification benchmarks and involves no human subjects or sensitive data.

   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: Broader impacts are discussed in Appendix B, covering efficiency for edge devices, privacy benefits of exemplar-free learning, and the absence of significant negative societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper proposes a continual learning algorithm and releases no new datasets or high-risk models.

    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 datasets, the ViT backbone, PyTorch [25], and the PILOT codebase [34] are properly cited and used for academic research purposes.

    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: No new datasets or model checkpoints are released. The DAF algorithm is fully described in the paper.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: No human subjects are involved; IRB approval is not required.

    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: LLMs are not part of the proposed method.

    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.