DISTRIBUTION-CONDITIONED TASK ROUTING FOR CLASS-INCREMENTAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Parameter-efficient adaptation enables continual learners to acquire task-specific knowledge through compact model updates while maintaining strong within-task performance. However, class-incremental inference requires each input to be classified among all classes seen so far without access to its task identity. For learners equipped with task-specific parameter-efficient modules, this introduces a critical task-routing challenge beyond catastrophic forgetting. We study post-hoc task routing without retraining the learner or introducing a separately trained router. Such training-free inference-time calibration remains comparatively underexplored in parameter-efficient class-incremental learning. We identify three sources of routing error—feature-level, task-level, and class-level misalignment—and propose Feature Distribution Calibration (FDC). Its three components address these misalignments: Task Subspace Filtering (TSF) suppresses feature components outside each task's principal subspace, Residual Likelihood Calibration (RLC) evaluates the typicality of its subspace residual, and Prototype Affinity Calibration (PAC) measures compatibility with the task's class prototypes. Experiments demonstrate plug-and-play applicability to eight parameter-efficient class-incremental methods using a shared encoder. With one component configuration selected per method across all five benchmarks, FDC improves final accuracy in all 40 method–dataset pairs by 4.39 percentage points on average. Enabling all components improves 35 of the 40 pairs, with an average gain of 4.45 points. When applied to a simple baseline, FDC achieves strong overall performance.

1 INTRODUCTION

Pre-trained models (PTMs) (Zhou et al., 2024) have achieved remarkable success and substantially reshaped the paradigm of continual learning (CL). Traditional CL methods (Khademi Nori et al., 2025; Wang et al., 2026) often adapt a large portion of model parameters while explicitly addressing catastrophic forgetting, whereas PTMs enable more parameter-efficient alternatives, such as prompt tuning (Wang et al., 2022b; Hong et al., 2025) and Low-Rank Adaptation (LoRA) (Hu et al., 2022), to preserve previously acquired knowledge while maintaining strong task-specific adaptation. Consequently, LoRA-based class-incremental learning (CIL) has attracted increasing attention in recent years.

Existing LoRA-based CIL methods (Wu et al., 2025; Liu & Chang, 2025; Qiu et al., 2026) have devoted substantial effort to mitigating catastrophic forgetting as the label space continually expands, achieving competitive performance across a variety of benchmarks. However, we highlight another complementary yet critical challenge: *task routing*, i.e., identifying the appropriate task-specific knowledge for an input when task identity is unavailable at inference time. This challenge becomes particularly important because, as empirically demonstrated in Figure 2, within-task discrimination is largely preserved, whereas discrimination across tasks remains substantially impaired. This suggests that routing, rather than within-task recognition itself, can become a key bottleneck in class-incremental inference. While task routing itself is not new (Kim et al., 2022; Wang et al., 2023), existing solutions typically rely on learned routers (Wang et al., 2025a), out-of-distribution (OOD)-aware continual training (Jia et al., 2026), or prototype- and distribution-based prediction mechanisms (Lin et al., 2024), all of which introduce additional training requirements or modify the original learning procedure. In contrast, whether task routing can be effectively improved *after* a LoRA-based CIL model has already been trained remains largely unexplored.

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

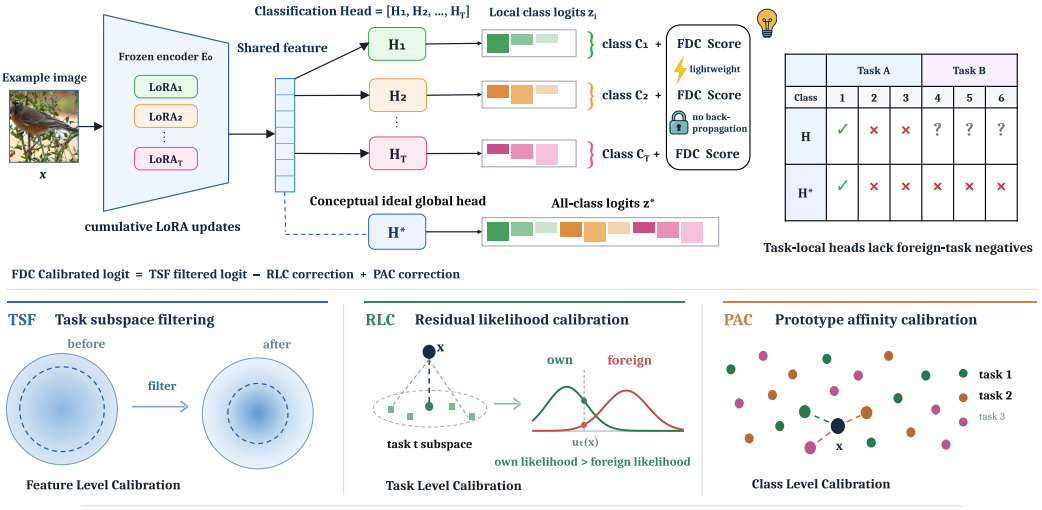

Figure 1: Post-hoc task routing with FDC. Top: task-local heads lack foreign-task negatives; H^* illustrates an ideal global classifier (Section 3). Bottom: TSF, RLC, and PAC calibrate routing through feature support, task membership, and class affinity, respectively, without additional training.

In this paper, we investigate this practical yet challenging setting, termed *post-hoc task routing* for LoRA-based CIL. Given an already trained CIL learner, our goal is to improve task routing without updating the encoder or classifier parameters and without training an additional router. Meanwhile, the additional storage and computational overhead should remain modest to preserve the efficiency benefits of parameter-efficient adaptation. These constraints require us to extract discriminative routing evidence directly from the established feature space. To this end, we analyze routing errors from a coarse-to-fine perspective and identify three distinct forms of misalignment: *feature-level*, *task-level*, and *class-level* misalignment. We empirically verify their prevalence across existing benchmarks in Sec. 3. At the feature level, residual components outside a task's principal support tend to make positive logit contributions for foreign-task inputs, while contributing little or even negatively to own-task inputs. At the task level, own-task and foreign-task inputs exhibit distinct residual-energy distributions, providing useful evidence for task membership. At the class level, individual class prototypes preserve fine-grained structure that can be obscured by task-level means.

Building on these observations, we propose a unified post-hoc calibration framework, FDC (Figure 1), to systematically address the three levels of routing misalignment. Specifically, *Task Subspace Filtering* (TSF) suppresses feature components outside each task's principal subspace before applying its classifier, thereby mitigating feature-level misalignment. *Residual Likelihood Calibration* (RLC) evaluates whether the residual geometry of an input is more typical of the candidate task or of foreign tasks, providing task-level evidence. Finally, *Prototype Affinity Calibration* (PAC) measures the input's affinity to each task's class prototypes relative to its training distribution, complementing routing decisions with class-level evidence. Together, these components progressively calibrate routing from feature support to task membership and class affinity, while contributing to the scores of all task classification heads without additional gradient-based training.

Our contributions are threefold:

- We investigate a practical yet underexplored setting, *post-hoc task routing* for LoRA-based CIL, where routing is improved after training without modifying the learned encoder or classifier. We further identify and empirically characterize three complementary sources of routing error: feature-, task-, and class-level misalignment.
- We propose FDC, a lightweight post-hoc calibration framework that jointly addresses these misalignments through feature support projection, task-level residual evidence, and class-prototype affinity. FDC requires neither additional gradient-based optimization nor a replay buffer, while keeping the learned encoder and classifier parameters fixed.
- Across five benchmarks and three training seeds, FDC improves Cumulative-LoRA by 3.58–10.88 percentage points. Moreover, plug-and-play evaluation on eight additional CIL

learners yields positive average gains across all 40 method–dataset combinations, with an overall improvement of 4.39 percentage points using a single component configuration per learner across the five benchmarks.

2 Related Work

2.1 Parameter-Efficient Class-Incremental Learning

CIL requires recognizing all previously encountered classes without task labels at inference. Pretrained vision transformers provide transferable representations (Dosovitskiy et al., 2021), which continual learners adapt through compact prompts, adapters, or low-rank updates. RainbowPrompt aggregates task-specific prompts through learned transformations and alignment (Hong et al., 2025), while adapter-based approaches explore task-specific and shared adaptation (Wang et al., 2025c; Zhao et al., 2026). Among these approaches, LoRA parameterizes weight updates with low-rank factors (Hu et al., 2022). InfLoRA and SplitLoRA constrain update spaces to manage interference (Liang & Li, 2024; Qiu et al., 2026); SD-LoRA decouples update magnitude and direction (Wu et al., 2025); and LoDA separates knowledge-sharing and task-specific subspaces (He et al., 2026). Complementary strategies regularize low-rank updates (Zheng et al., 2026), allocate capacity proactively (Wang et al., 2025b), or construct drift-resistant representations (Liu & Chang, 2025). FDC complements these training strategies by using feature statistics to calibrate competition among task heads, while keeping the learned encoder and classifier parameters fixed.

2.2 Task Routing and Classifier Calibration

Kim et al. (2022) connect within-task recognition and task inference to OOD detection; HiDe-Prompt further separates within-task, task-identity, and task-adaptive prediction (Wang et al., 2023). Modular learners address task inference through learned adapter weights (SEMA) (Wang et al., 2025a), training-free self-refined retrieval (MOS) (Sun et al., 2025b), or entropy-based selection over candidate-specific predictions (GR-LoRA) (Lin et al., 2026). FDC instead calibrates competition among heads sharing one adapted representation.

Weight Aligning corrects classifier bias by rescaling weights without further optimization (Zhao et al., 2020); SLCA optimizes the classifier using sampled class-wise features (Zhang et al., 2023). T-CIL targets confidence calibration through temperature optimization with perturbed memory exemplars and new-task validation data (Hwang et al., 2025). FDC targets cross-task classification errors; Section 5.3 tests its compatibility with classifier alignment through InfLoRA+CA.

Feature statistics also support inference: FeCAM uses class means and covariances for Mahalanobis classification (Goswami et al., 2023), and AnaCP analytically learns a contrastive projection and classifier (Momeni et al., 2025). TPL predicts tasks using likelihood ratios with task-specific models and replay-based estimates (Lin et al., 2024); ViM combines subspace residuals with logits for OOD detection (Wang et al., 2022a). GOD instead shapes feature geometry during training through ETF and ArcFace losses (Jia et al., 2026). FDC combines subspace support, residual typicality, and prototype affinity to calibrate existing heads, using compact statistics without replay or additional gradient-based training.

3 Preliminary and Motivation

3.1 Preliminary

Consider T tasks with training sets $\mathcal{D}_1, \ldots, \mathcal{D}_T$ and mutually disjoint class sets $\mathcal{C}_1, \ldots, \mathcal{C}_T$. Let E_0 be the frozen pretrained encoder and E_t the encoder after learning task t. For each adapted matrix, low-rank updates accumulate as $W_{\text{enc}}^{(t)} = W_{\text{enc}}^{(0)} + \sum_{k=1}^{t} B_k A_k$. Task t learns a head $g_t(h) = W_t h + b_t$ for its classes. Its current-task training minimizes empirical classification loss:

$$\hat{\theta}_t \in \arg\min_{\theta_t} \frac{1}{|\mathcal{D}_t|} \sum_{(x,y)\in\mathcal{D}_t} \ell(g_t(E_t(x)), y), \tag{1}$$

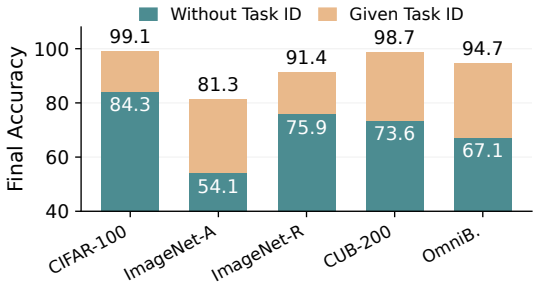

Figure 2: Final accuracy comparison between w/ and w/o task identity.

Dataset	Accuracy change (pp)
CIFAR-100	$+0.35 \pm 0.85$
ImageNet-A	-0.02 ± 0.34
ImageNet-R	-0.79 ± 0.34
CUB-200	-0.54 ± 1.23
OmniBenchmark	$+0.11 \pm 0.23$

Table 1: Accuracy change after LS.

TSF	RLC	PAC
$\mathcal{O}(Td^2)$	$\mathcal{O}(Td^2)$	$\mathcal{O}(Cd)$

Table 2: Extra storage for FDC statistics. T: number of tasks; C: total number of classes; d: classification head input dimension.

where θ_t comprises the new LoRA increment and current head. Historical classifier blocks remain fixed. At final inference, all heads share E_T, and prediction spans all seen classes:

$$z_t(x) = W_t E_T(x) + b_t, \qquad (\hat{t}, \hat{c}) = \arg\max_{t,\, c \in \mathcal{C}_t} z_{t,c}(x). \tag{2}$$

The local argmax in the selected head yields the global class prediction; the maximum logit indicates the head's preferred class, but training on $\mathcal{D}_t$ does not directly constrain its response to another task.

3.2 Motivation

Task routing is a key challenge. We compare two prediction settings for the same final model and test inputs. Without Task ID, the model selects the class with the highest logit across all seen classes; Given Task ID, it selects the class with the highest logit only within the true task's classes. Both settings use the same final shared encoder and concatenated classifier heads, with all accumulated LoRA updates active. Providing task identity therefore does not undo any forgetting caused by subsequent updates, which can still affect recognition within the true task. Nevertheless, Figure 2 shows a substantial accuracy gap across all five datasets, with task identity alone correcting around 80% of errors on average. Thus, much of the required recognition capability remains available in the shared representation but is obscured by cross-head competition. This makes routing calibration a central priority alongside mitigating catastrophic forgetting during training. We want to clarify that we are re-identifying the task routing misalignment during the post-hoc stage in CIL, rather than presenting this observation as our original contribution.

Limited gains from logit standardization (LS). A natural first hypothesis is that the separately trained heads produce logits with different offsets and scales, potentially biasing their competition. Although the encoder is shared through cumulative LoRA updates, the heads are simply concatenated at inference:

$$z(x) = \begin{bmatrix} z_1(x) \\ \vdots \\ z_T(x) \end{bmatrix} = \begin{bmatrix} W_1 \\ \vdots \\ W_T \end{bmatrix} E_T(x) + \begin{bmatrix} b_1 \\ \vdots \\ b_T \end{bmatrix}. \tag{3}$$

To test this hypothesis, each head's maximum logit is standardized as:

$$s_t^{\mathrm{LS}}(x) = \frac{s_t^{\mathrm{raw}}(x) - \mu_t^s}{\max(\sigma_t, \epsilon)}, \qquad \hat{t}_{\mathrm{std}}(x) = \arg\max_t s_t^{\mathrm{LS}}(x), \tag{4}$$

where μ_t^s and σ_t are the mean and standard deviation of head t's maximum logit on its own training data, and $\epsilon = 10^{-12}$ ensures numerical stability. This transformation changes head selection while preserving each head's local class prediction. However, joint centering and scaling changes CIL accuracy by only $+0.35$, -0.02, -0.79, -0.54, and $+0.11$ percentage points on the five datasets, respectively (see Table 1), providing no consistent improvements. Correcting these score statistics alone therefore appears insufficient for reliable routing. These results motivate keeping learned logits while supplementing them with evidence about the current input's compatibility with each task's training distribution.

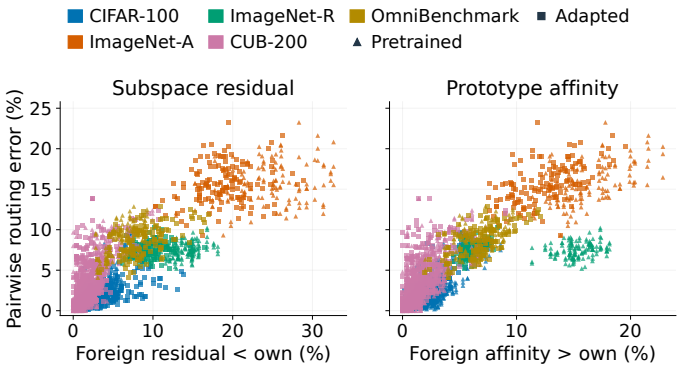

Figure 3: Feature similarity versus routing error. Pre-trained/adapted use the frozen encoder/encoder after each task. Similarity uses subspace residuals (left) and prototype affinities (right).

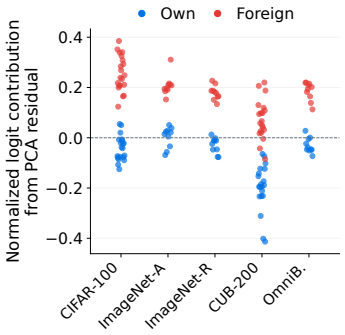

Figure 4: PCA-residual logit contributions on own and foreign inputs. Each point denotes one head averaged over its own or foreign test samples.

Head input similarity and routing confusion. We further empirically identify that head-input similarity is positively correlated with routing confusion. As shown in Figure 3, task pairs that are more confusable in terms of subspace residuals (Eq. (11)) or affinities to the class prototypes (Eq. (13)) tend to exhibit more routing errors. Here, head-input similarity is assessed by comparing how well an input fits its own task's feature distribution with how well it fits another task's. Subspace residuals measure its fit to each task's principal subspace, while prototype affinities measure its resemblance to each task's class mean prototypes.

Routing errors at three levels. To analyze the sources of the error gap in Figure 2, we compare the task-locally trained classifier H with a hypothetical jointly supervised reference H^* serving as the calibration target. Since both settings use the same model, we assume that H and H^* receive identical inputs. H trains each head using only within-task class labels. H^* additionally uses cross-task labels under an all-class classification loss, providing negative supervision for foreign inputs.

For a foreign class $c \in \mathcal{C}_{t'}$, $t' \neq t$, write $\bar{t}$ for "not task t." The probability chain rule gives a three-level factorization, exposing the cross-task supervision available to H^* but absent from H:

$$P(c \mid x) = \underbrace{P(\bar{t} \mid x)}_{\text{not task } t}\ \underbrace{P(t' \mid \bar{t}, x)}_{\text{which task}}\ \underbrace{P(c \mid t', x)}_{\text{which class}}. \tag{5}$$

To translate these evidence levels into score corrections, let $r_t^{\text{feat}}, r_t^{\text{task}}, r_t^{\text{class}} > 0$ denote their multiplicative calibration factors. Let s_t, s_t^* be the maximum class logits of H, H^* and $\sigma_t > 0$ the head's own-training maximum-logit standard deviation. We model calibration on the scale s_t/σ_t as

$$\begin{aligned} \frac{\exp(s_t(x)/\sigma_t)}{\exp(s_t^*(x)/\sigma_t)} &= r_t^{\text{feat}}(x)\, r_t^{\text{task}}(x)\, r_t^{\text{class}}(x), \\ \xRightarrow{\log} \quad \frac{s_t(x) - s_t^*(x)}{\sigma_t} &= \log r_t^{\text{feat}}(x) + \log r_t^{\text{task}}(x) + \log r_t^{\text{class}}(x). \end{aligned} \tag{6}$$

Writing $\Delta_t^{\ell}(x) = \log r_t^{\ell}(x)$ for $\ell \in \{\text{feat}, \text{task}, \text{class}\}$ and multiplying by σ_t gives

$$s_t^*(x) = s_t(x) - \sigma_t\big[\Delta_t^{\text{feat}}(x) + \Delta_t^{\text{task}}(x) + \Delta_t^{\text{class}}(x)\big]. \tag{7}$$

We examine these three levels through the following diagnostics. At the *feature level*, a head's high response to foreign inputs may arise from directions rarely used by its own-task inputs. We measure the normalized logit contribution from features outside the task's principal component analysis (PCA) (Pearson, 1901) subspace (Appendix A.3). Figure 4 shows generally positive mean residual contributions on foreign inputs, versus small or negative contributions on own-task inputs. This observation motivates TSF (Section 4.2).

At the *task level*, PCA also provides a measure of how well an input fits a task's feature distribution. Fitted to samples across the task's classes, its principal subspace summarizes their dominant

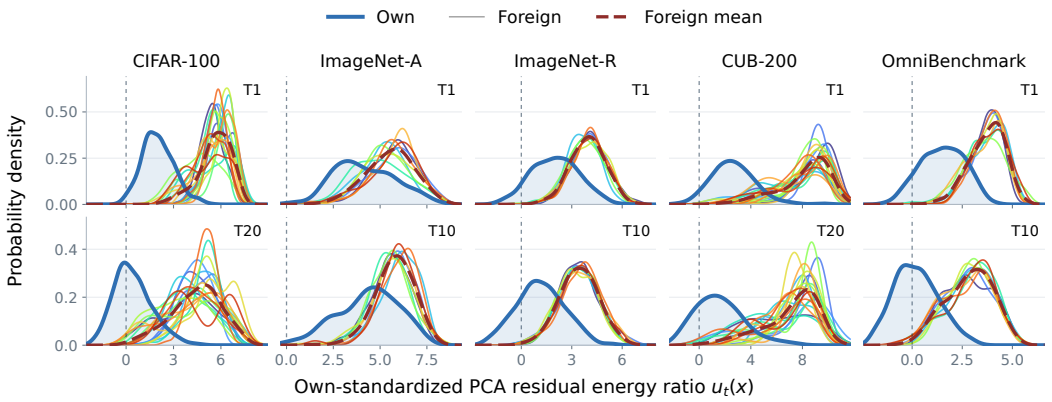

Figure 5: Own-standardized PCA residual distributions for the first and last task heads. Dashed curves show the equal-task mean of foreign densities.

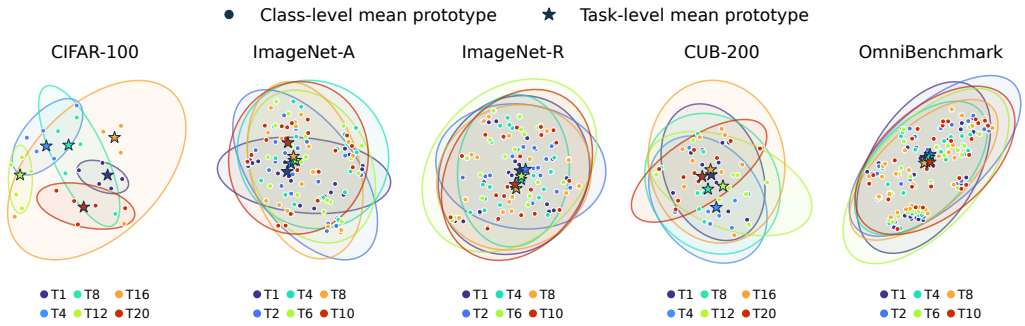

Figure 6: Mean prototype geometry across five datasets. Dots are class mean prototypes and stars their task-level mean prototype, jointly embedded with t-SNE.

feature variation. The fraction of centered feature energy outside this subspace then measures the input's deviation from that task. Figure 5 shows that foreign inputs generally have higher residual energy ratios than own-task inputs, with each task's training statistics used for standardization. This difference motivates the residual-based task calibration in RLC (Section 4.3).

At the *class level*, an input's resemblance to a particular class in another task may help explain why that task's classification head competes with the true task's head. Figure 6 illustrates how task mean prototypes can cluster closely together while class mean prototypes retain more distinct locations. Averaging classes into a single task prototype can obscure this finer structure, making class-level affinity useful for assessing which task contains a class compatible with the input. PAC (Section 4.4) uses this class-level evidence for calibration.

4 Principal Design

4.1 Overview

To address the three levels of routing misalignment in Section 3, FDC uses three calibration modules: Task Subspace Filtering (TSF) for feature-level correction, Residual Likelihood Calibration (RLC) for task-level correction, and Prototype Affinity Calibration (PAC) for class-level correction. It calibrates task classification heads without back-propagation or any separately trained routing mechanism by observing the d-dimensional input features of task classification heads across T tasks and $C = \sum_{t=1}^{T} |\mathcal{C}_t|$ classes. Figure 1 summarizes the routing problem and the three calibration modules.

For classification head input features $g(x)$, task-level variation is captured by PCA:

$$U_t = \mathrm{PCA}_\eta\big(\{g(x) : x \in \mathcal{D}_t\}\big), \qquad \Pi_t = U_t U_t^\top. \tag{8}$$

PCA centers the features by subtracting their task mean and retains the fewest leading components explaining at least a fraction η of the total variance (Appendix E.1). Class-level structure is captured by averaging the unit-normalized features $\widehat{g}(x) = g(x)/\|g(x)\|_2$ within each class:

$$p_{t,c} = \mathrm{norm}\left(\frac{1}{|\mathcal{D}_{t,c}|}\sum_{x\in\mathcal{D}_{t,c}} \widehat{g}(x)\right), \qquad \mathrm{norm}(u) = \frac{u}{\|u\|_2}. \tag{9}$$

Here $\mathcal{D}_{t,c}$ contains task t's training samples of class c. Statistics are collected as tasks are learned.

4.2 Task Subspace Filtering (TSF)

As observed in Figure 4, feature components outside a task's principal subspace can inflate its logits on foreign inputs. To attenuate these contributions, TSF uses the adapted-view projection Π_t^t from Eq. (8) to transform the task's classifier weights:

$$\widetilde{W}_t = W_t - \gamma W_t(I - \Pi_t^t), \tag{10}$$

where $\gamma \in [0, 1]$ controls the attenuation strength (Appendix E.1). This transformation scales the contribution of features orthogonal to the task subspace by $1 - \gamma$, while preserving the contribution within the subspace. With the learned head bias b_t unchanged, the filtered logits are $\widetilde{z}_t(x) = \widetilde{W}_t E_T(x) + b_t$, giving the task logit (the maximum class logit within head t) $\widetilde{s}_t(x) = \max_{c\in\mathcal{C}_t} \widetilde{z}_{t,c}(x)$.

4.3 Residual Likelihood Calibration (RLC)

RLC compares an input's subspace residual under the task's own distribution and a foreign-task reference. For a feature $g(x)$ from either encoder, we use the task projection Π_t from Eq. (8) to compute

$$\rho_t(x) = \frac{\|(I - \Pi_t)(g(x) - m_t)\|_2^2}{\|g(x) - m_t\|_2^2}, \qquad u_t(x) = \frac{\rho_t(x) - \mu_t^\rho}{\delta_t}, \tag{11}$$

where m_t is the mean training feature, and μ_t^ρ and δ_t are the mean and standard deviation of training residuals, computed separately for each encoder.

We approximate the standardized own-task residuals by $p_O = \mathcal{N}(0, 1)$. The foreign reference $p_F = \mathcal{N}(\mu_F, V_F)$ is fitted by evaluating task t's features $g(x)$ against the stored statistics of earlier tasks. The resulting likelihood correction is

$$S_t^{\mathrm{RLC}}(x) = \sigma_t \tanh\left(\frac{1}{\tau}\log\frac{p_F(u_t(x))}{p_O(u_t(x))}\right), \tag{12}$$

where τ is the median absolute log-likelihood ratio over the foreign samples. RLC subtracts $S_t^{\mathrm{RLC}}(x)$ from the task logit. A higher foreign likelihood decreases the score, whereas a higher own-task likelihood increases it.

4.4 Prototype Affinity Calibration (PAC)

PAC assesses task compatibility through the closest class prototype, retaining the class-level structure that a single task prototype can obscure. For each candidate task t, PAC performs a nearest-neighbor search ($K = 1$) over the class prototypes $p_{t,c}$ from Eq. (9), using cosine similarity. The similarity to the retrieved prototype serves as the task affinity, which we standardize using the task's training statistics:

$$A_t(x) = \max_{c\in\mathcal{C}_t} \frac{g(x)^\top p_{t,c}}{\|g(x)\|_2}, \qquad q_t(x) = \frac{A_t(x) - \mu_t^a}{\sigma_t^a}, \tag{13}$$

where μ_t^a and σ_t^a are the mean and standard deviation of training affinities computed with the corresponding training-time encoder for $x \in \mathcal{D}_t$. Setting $K = 1$ yields the best PAC accuracy in the neighborhood-size sweep (Appendix E.2). The bounded affinity correction is

$$S_t^{\mathrm{PAC}}(x) = \sigma_t \tanh(q_t(x)). \tag{14}$$

PAC adds $S_t^{\mathrm{PAC}}(x)$ to the task logit. Affinity above the task's training mean increases the score, while affinity below the mean decreases it.

FDC Framework. After learning all T tasks, FDC first applies TSF to obtain the base task logit $\widetilde{s}_t(x)$ from the final adapted features $E_T(x)$. RLC and PAC then calibrate this score using the input's compatibility with each task. In the adapted view, they compare $E_T(x)$ with statistics collected from $E_t(\mathcal{D}_t)$ when task t was learned. The pretrained view uses E_0 for both statistics collection and inference. For a single view, the calibrated task logit is

$$S_t(x) = \widetilde{s}_t(x) - \alpha S_t^{\mathrm{RLC}}(x) + \beta S_t^{\mathrm{PAC}}(x), \tag{15}$$

where α and β control the strengths of residual and affinity calibration.

Dual-view calibration combines corrections from the adapted and pretrained encoders while retaining the same TSF base score:

$$\begin{aligned} S_t^{\mathrm{dual}}(x) = \widetilde{s}_t(x) &+ w_T\big[\beta S_t^{\mathrm{PAC},T}(x) - \alpha S_t^{\mathrm{RLC},T}(x)\big] \\ &+ w_0\big[\beta S_t^{\mathrm{PAC},0}(x) - \alpha S_t^{\mathrm{RLC},0}(x)\big], \end{aligned} \tag{16}$$

where superscripts T and 0 denote the final adapted encoder E_T and the frozen pretrained encoder E_0, weighted by w_T and w_0, respectively. Across all datasets and baseline methods, we fix the RLC/PAC correction weights and the two view weights to one (i.e., $\alpha = \beta = w_T = w_0 = 1$), the PCA retained-energy threshold to $\eta = 0.75$, and the TSF attenuation strength to $\gamma = 0.5$. These shared settings avoid dataset- or baseline-specific hyperparameter tuning.

Plug-and-play. FDC can be integrated into existing continual learning methods using the shared hyperparameter settings above, without method-specific tuning. We organize FDC into three independently selectable components: TSF, pretrained-view calibration (E_0 FDC), and adapted-view calibration (E_T FDC). Both E_0 FDC and E_T FDC combine RLC and PAC and can be used individually or together.

5 Evaluation

5.1 Implementation

Following the dataset setup of PILOT (Sun et al., 2025a), we evaluate on CIFAR-100 (Krizhevsky, 2009), ImageNet-A (Hendrycks et al., 2021b), ImageNet-R (Hendrycks et al., 2021a), CUB-200 (Wah et al., 2011), and OmniBenchmark (Zhang et al., 2022). CIFAR-100 and CUB-200 are divided into 20 tasks with 5 and 10 classes per task, respectively. ImageNet-A, ImageNet-R, and OmniBenchmark use 10 tasks with 20, 20, and 30 classes per task. We report final CIL accuracy as mean $\pm$ sample standard deviation over three training seeds with a fixed class order; gains are measured in percentage points (pp). Data partitions and seed settings are detailed in Appendix A.1.

Cumulative-LoRA baseline. Our reference baseline, *Cumulative-LoRA* (Section 3), trains a new LoRA increment and head per task while freezing previous updates and heads. The increments accumulate in a shared encoder used by all heads at inference.

All methods use ViT-B/16 (Dosovitskiy et al., 2021) with ImageNet-21K AugReg pretraining (Steiner et al., 2022). LoRA runs use rank-10 key/value increments, batch size 64, 20 main-training epochs per task, Adam, zero weight decay, and LoRA/head learning rates of $5 \times 10^{-4}/5 \times 10^{-3}$. We reproduce each method as closely as possible under this protocol; implementation details are provided in Appendix A.2. Each comparison with and without FDC reuses the same trained checkpoint and test examples. Computational and memory costs are reported in Appendix C. Component and view ablations are reported in Appendix D; average incremental accuracy for complete training trajectories is provided in Appendix F.

5.2 Main Results

Table 3 reports final CIL accuracy before and after FDC, with mean $\pm$ sample standard deviation over three seeds. On our reference baseline, Cumulative-LoRA, FDC improves accuracy by

Table 3: Final CIL accuracy (%) before and after FDC. Bold marks the best mean per dataset.

Method	CIFAR-100	ImageNet-A	ImageNet-R	CUB-200	OmniB.
Cumulative-LoRA	84.34±0.15	54.11±0.17	75.91±0.34	73.55±1.70	67.15±0.21
+ FDC	88.62±0.13	60.08±1.02	79.48±0.24	84.44±0.26	73.49±0.39
EWC-LoRA	82.50±0.26	51.39±0.60	74.31±0.33	70.77±0.11	66.88±0.37
+ FDC	87.31±0.43	56.33±0.63	77.50±0.29	83.31±0.25	74.56±0.30
SD-LoRA	84.87±0.85	53.35±0.86	75.77±0.23	71.67±1.09	68.13±0.19
+ FDC	89.20±0.25	59.05±0.23	78.98±0.28	82.99±0.68	74.39±0.61
SplitLoRA	88.20±0.28	53.70±0.71	78.68±0.13	75.56±0.36	70.35±0.38
+ FDC	**89.82±0.12**	56.37±0.44	80.13±0.04	81.27±0.15	73.08±0.17
InfLoRA	84.82±0.58	51.55±2.18	75.06±0.28	72.53±1.11	66.71±0.62
+ FDC	89.03±0.31	58.04±1.07	78.81±0.22	83.09±0.60	74.36±0.10
InfLoRA+CA	84.41±0.70	58.77±0.96	76.66±0.30	84.34±0.67	71.89±0.44
+ FDC	88.96±0.30	60.90±0.57	79.13±0.25	86.57±0.51	**75.10±0.26**
LoRA-Subtraction	34.61±3.91	56.92±0.93	76.09±0.18	87.21±0.32	42.38±1.56
+ FDC	39.03±6.65	57.14±1.36	76.88±0.32	87.63±0.06	48.69±1.41
LoDA	87.41±0.62	62.78±0.27	80.69±0.17	77.76±0.23	69.83±0.15
+ FDC	88.96±0.48	**64.76±0.40**	**81.41±0.24**	82.85±0.30	72.77±0.28
RSIAT	54.81±0.29	60.87±0.76	60.46±0.48	87.59±0.40	61.96±0.56
+ FDC	66.93±0.65	61.73±0.34	66.64±0.69	**88.71±0.51**	67.42±0.49

4.28, 5.97, 3.58, 10.88, and 6.35 pp on CIFAR-100, ImageNet-A, ImageNet-R, CUB-200, and OmniBenchmark, respectively, without changing the trained model. Improvements are consistent across all 15 dataset–seed pairs, with the largest gain on CUB-200.

5.3 Plug-and-Play Calibration

We apply FDC to EWC-LoRA (Zheng et al., 2026), SD-LoRA (Wu et al., 2025), SplitLoRA (Qiu et al., 2026), InfLoRA and its classifier-alignment variant (InfLoRA+CA) (Liang & Li, 2024), LoRA-Subtraction (Liu & Chang, 2025), LoDA (He et al., 2026), and RSIAT (Zhao et al., 2026), using each method's trained checkpoint without retraining. For each method, we enable all components whose standalone gains, averaged over seeds, are positive on every dataset. The resulting configuration is shared across all five datasets and three seeds. Appendix G reports the selected component configurations and their full-component counterparts, together with paired gains in Table 10. With selective configurations, FDC improves all 40 method–dataset pairs by 4.39 pp on average. Enabling all components for every method yields a comparable 4.45 pp gain, with improvements in 35 of the 40 pairs and a positive five-dataset average for every method. Thus the overall benefit persists without method-specific component selection. FDC also improves InfLoRA+CA by 2.92 pp, supporting its compatibility with classifier alignment under the same numerical hyperparameters.

6 Limitations

Although FDC improves inference without task identity, a substantial gap to the Given Task ID accuracy remains. Our calibration study is restricted to methods that share a single adapted encoder across task heads, with Cumulative-LoRA as the reference baseline. We have not studied how to integrate FDC into inference over multiple candidate experts or task-specific subnetworks, as used by GR-LoRA (Lin et al., 2026) and TPL (Lin et al., 2024), although they incur substantial additional computational and storage costs.

7 Conclusion

FDC improves task routing in class-incremental learners sharing an adapted encoder through subspace filtering, residual likelihood, and prototype affinity, without additional training. Its consistent gains on Cumulative-LoRA and broad applicability to existing methods show that feature-statistic calibration helps learners better use their retained knowledge when task identity is unavailable.

AI USE STATEMENT

AI-assisted tools were used to help organize the manuscript, draft and edit text and equations, inspect related-work sources, and prepare tables from existing experiment outputs. Scientific interpretations, bibliographic accuracy, and the reported results remain the authors' responsibility.

REFERENCES

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost van de Weijer. FeCAM: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/15294ba2dcfb4521274f7aa1c26f4dd4-Abstract-Conference.html.

Lingfeng He, De Cheng, Huaijie Wang, Xi Yang, Nannan Wang, and Xinbo Gao. Task-driven subspace decomposition for knowledge sharing and isolation in LoRA-based continual learning. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/63718.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8349, 2021a. URL https://openaccess.thecvf.com/content/ICCV2021/html/Hendrycks_The_Many_Faces_of_Robustness_A_Critical_Analysis_of_Out-of-Distribution_ICCV_2021_paper.html.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15262–15271, 2021b. URL https://openaccess.thecvf.com/content/CVPR2021/html/Hendrycks_Natural_Adversarial_Examples_CVPR_2021_paper.html.

Kiseong Hong, Gyeong hyeon Kim, and Eunwoo Kim. RainbowPrompt: Diversity-enhanced prompt-evolving for continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1130–1140, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Hong_RainbowPrompt_Diversity-Enhanced_Prompt-Evolving_for_Continual_Learning_ICCV_2025_paper.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Seong-Hyeon Hwang, Minsu Kim, and Steven Euijong Whang. T-CIL: Temperature scaling using adversarial perturbation for calibration in class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15339–15348, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Hwang_T-CIL_Temperature_Scaling_using_Adversarial_Perturbation_for_Calibration_in_Class-Incremental_CVPR_2025_paper.html.

Wangwang Jia, Zijian Gao, Tianjiao Wan, Yuan Cao, Yong Dou, and Kele Xu. Geometry-driven OOD detectors are class-incremental learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34638–34649, 2026. URL

https://openaccess.thecvf.com/content/CVPR2026/html/Jia_Geometry-driven_OOD_Detectors_Are_Class-Incremental_Learners_CVPR_2026_paper.html.

Milad Khademi Nori, Il Min Kim, and Guanghui Wang. Autoencoder-based hybrid replay for class-incremental learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 29753–29767, 2025. URL https://proceedings.mlr.press/v267/khademi-nori25a.html.

Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, Zixuan Ke, and Bing Liu. A theoretical study on solving continual learning. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/20f44da80080d76bbc35bca0027f14e6-Abstract-Conference.html.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Yan-Shuo Liang and Wu-Jun Li. InfLoRA: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Liang_InfLoRA_Interference-Free_Low-Rank_Adaptation_for_Continual_Learning_CVPR_2024_paper.html.

Haowei Lin, Yijia Shao, Weinan Qian, Ningxin Pan, Yiduo Guo, and Bing Liu. Class incremental learning via likelihood ratio based task prediction. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/de0e668df3fe63ec89e5a7e68f3d350f-Abstract-Conference.html.

Yipeng Lin, Fengqiang Wan, and Yang Yang. GR-LoRA: Gradient-recycling low-rank adaptation for class-incremental learning. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/64527.

Xuan Liu and Xiaobin Chang. LoRA subtraction for drift-resistant space in exemplar-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/papers/Liu_LoRA_Subtraction_for_Drift-Resistant_Space_in_Exemplar-Free_Continual_Learning_CVPR_2025_paper.pdf.

Saleh Momeni, Changnan Xiao, and Bing Liu. AnaCP: Toward upper-bound continual learning via analytic contrastive projection. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2444. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/699b19e638a086ed3a6d1710c5aea504-Abstract-Conference.html.

Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. doi: 10.1080/14786440109462720. URL https://doi.org/10.1080/14786440109462720.

Haomiao Qiu, Miao Zhang, Ziyue Qiao, Weili Guan, Min Zhang, and Liqiang Nie. SplitLoRA: Balancing stability and plasticity in continual learning through gradient space splitting. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5035a409f5798e188079e236f437e522-Abstract-Conference.html.

Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your ViT? data, augmentation, and regularization in vision transformers. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=4nPswr1KcP.

Hai-Long Sun, Da-Wei Zhou, De-Chuan Zhan, and Han-Jia Ye. PILOT: A pre-trained model-based continual learning toolbox. *Science China Information Sciences*, 68(4):147101, 2025a. doi: 10.1007/s11432-024-4276-4. URL https://scis.scichina.com/en/2025/147101.pdf.

Hai-Long Sun, Da-Wei Zhou, Hanbin Zhao, Le Gan, De-Chuan Zhan, and Han-Jia Ye. MOS: Model surgery for pre-trained model-based class-incremental learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20699–20707, 2025b. doi: 10.1609/aaai.v39i19.34281. URL https://ojs.aaai.org/index.php/AAAI/article/view/34281.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical report, California Institute of Technology, 2011. URL https://authors.library.caltech.edu/records/cvm3y-5hh21.

Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. ViM: Out-of-distribution with virtual-logit matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022a. URL https://openaccess.thecvf.com/content/CVPR2022/html/Wang_ViM_Out-of-Distribution_With_Virtual-Logit_Matching_CVPR_2022_paper.html.

Huiyi Wang, Haodong Lu, Lina Yao, and Dong Gong. Self-expansion of pre-trained models with mixture of adapters for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025a. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_Self-Expansion_of_Pre-trained_Models_with_Mixture_of_Adapters_for_Continual_CVPR_2025_paper.html.

Liyuan Wang, Jingyi Xie, Xingxing Zhang, Mingyi Huang, Hang Su, and Jun Zhu. Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d9f8b5abc8e0926539ecbb492af7b2f1-Abstract-Conference.html.

Xiequn Wang, Zhan Zhuang, and Yu Zhang. PLAN: Proactive low-rank allocation for continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025b. URL https://openaccess.thecvf.com/content/ICCV2025/papers/Wang_PLAN_Proactive_Low-Rank_Allocation_for_Continual_Learning_ICCV_2025_paper.pdf.

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*, 2025c. URL https://openaccess.thecvf.com/content/ICCV2025/html/Wang_Integrating_Task-Specific_and_Universal_Adapters_for_Pre-Trained_Model-based_Class-Incremental_Learning_ICCV_2025_paper.html.

Zheng Wang, Wanhao Yu, Li Yang, and Sen Lin. Rethinking continual learning with progressive neural collapse. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/4ed3645a042ec800a474a6a312575c5e-Abstract-Conference.html.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 139–149, 2022b. URL https://openaccess.thecvf.com/content/CVPR2022/html/Wang_Learning_To_Prompt_for_Continual_Learning_CVPR_2022_paper.html.

Yichen Wu, Hongming Piao, Long-Kai Huang, Renzhen Wang, Wanhua Li, Hanspeter Pfister, Deyu Meng, Kede Ma, and Ying Wei. SD-LoRA: Scalable decoupled low-rank adaptation for class incremental learning. In *International Conference on Learning Representations*, 2025.

URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/92f43b1d33fae4aa1958f75317f0cec1-Abstract-Conference.html.

Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. SLCA: Slow learner with classifier alignment for continual learning on a pre-trained model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Zhang_SLCA_Slow_Learner_with_Classifier_Alignment_for_Continual_Learning_on_ICCV_2023_paper.html.

Yuanhan Zhang, Zhenfei Yin, Jing Shao, and Ziwei Liu. Benchmarking omni-vision representation through the lens of visual realms. In *European Conference on Computer Vision*, 2022. URL https://www.ecva.net/papers/eccv_2022/papers_ECCV/html/287_ECCV_2022_paper.php.

Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Zhao_Maintaining_Discrimination_and_Fairness_in_Class_Incremental_Learning_CVPR_2020_paper.html.

Jiarui Zhao, Libo Huang, Xiangqi Li, Zhulin An, Chuanguang Yang, Yu Wang, Boyu Diao, and Yongjun Xu. Representation-steered incremental adapter-tuning for class-incremental learning with pre-trained models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18010–18020, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zhao_Representation-Steered_Incremental_Adapter-Tuning_for_Class-Incremental_Learning_with_Pre-Trained_Models_CVPR_2026_paper.html.

Yaoyue Zheng, Yin Zhang, Joost van de Weijer, Gido M. van de Ven, Shaoyi Du, Xuetao Zhang, and Zhiqiang Tian. Revisiting weight regularization for low-rank continual learning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/47197750b407ef48daa1438e0c0af242-Abstract-Conference.html.

Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. Continual learning with pre-trained models: A survey. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 8363–8371, 2024. doi: 10.24963/ijcai.2024/924. URL https://www.ijcai.org/proceedings/2024/924.

Appendix

A Experimental Setup and Evaluation Protocol **15**

A.1 Datasets, Hyperparameters, and Metrics 15

A.2 Backbone and Training Settings 15

A.3 Feature-Level Routing Diagnostic 16

B FDC Algorithm **16**

B.1 Training-Time Statistics Collection 16

B.2 Test-Time Calibration 17

C Computational and Memory Costs of FDC **17**

C.1 Training-Time Costs 17

C.2 Inference-Time Costs 18

D FDC Component Ablation Studies **18**

E Hyperparameter Analysis of TSF and PAC **18**

E.1 TSF Hyperparameters 18

E.2 PAC Neighborhood Size 19

F Average Incremental Accuracy **20**

G Plug-and-Play Component Configurations **20**

A Experimental Setup and Evaluation Protocol

A.1 Datasets, Hyperparameters, and Metrics

Table 4 summarizes the class-disjoint, equal-sized task partitions and processed image counts. CIFAR-100 uses its canonical split; the other datasets follow the processed PILOT splits (Sun et al., 2025a), including a CUB-200 split that differs from the official one.

Table 4: Dataset partitions and task sequences.

Dataset	Classes	Tasks	Classes/task	Train	Test
CIFAR-100	100	20	5	50,000	10,000
ImageNet-A	200	10	20	5,981	1,519
ImageNet-R	200	10	20	24,000	6,000
CUB-200	200	20	10	9,430	2,358
OmniBenchmark	300	10	30	89,697	5,985

Task identity is available during training and statistics collection, but not at test time.

Table 5 lists the common training and FDC hyperparameters. Method-specific mechanisms are retained as described below; binary component switches are separate from the numerical hyperparameters.

Table 5: Common training and FDC hyperparameters.

Setting	Value
Main-training epochs per task / batch size	20 / 64
Optimizer / weight decay	Adam / 0
LoRA learning rate / head learning rate	5×10^{-4} / 5×10^{-3}
LoRA rank / adapted projections	10 / key and value
Training seeds / class-order seed	$1, 2, 3$ / 1993
PCA retained energy η	0.75
TSF attenuation γ	0.5
RLC and PAC weights (α, β)	$(1, 1)$
Enabled-view weights (w_T, w_0)	$(1, 1)$ for dual-view calibration
PAC prototype neighbors K	1

We report mean final CIL accuracy and sample standard deviation over three seeds, with Raw/FDC gains paired within each seed and datasets weighted equally. Given Task ID (Oracle) restricts prediction to the true task's classes using the same final model; task-routing accuracy measures correct task assignments.

A.2 Backbone and Training Settings

All methods use ImageNet-21K AugReg-pretrained ViT-B/16 with 768-dimensional CLS features. Cumulative-LoRA trains only the current task's rank-10 key/value increments and head; all accumulated increments are active at inference.

Method-specific implementation. EWC-LoRA (Zheng et al., 2026) estimates per-sample Fisher importance with regularization strength 10^7 and accumulation factor one. SplitLoRA (Qiu et al., 2026) retains gradient splitting with key/value updates. InfLoRA (Liang & Li, 2024) retains SVD initialization, historical-space projection, and its native cosine schedule; the learning-rate trajectories are therefore not identical across methods. InfLoRA+CA adds five classifier-alignment epochs on later tasks, using Adam with learning rate 5×10^{-3} and zero weight decay. LoRA-Subtraction (Liu & Chang, 2025) uses its projected Adam update. LoDA (He et al., 2026) retains its two branches with optional classifier alignment disabled. SD-LoRA (Wu et al., 2025) uses its standard forward inference and native historical state. RSIAT (Zhao et al., 2026) retains its native adapter, paired drift estimation, and ten classifier-calibration epochs on later tasks, with synthetic-feature batch size

256 and three warmup epochs. FDC preserves each method's classifier, including cosine scoring for LoDA and RSIAT, EWC-LoRA's head-input transform, and LoRA-Subtraction's nearest-class-mean scoring. Raw/FDC comparisons share checkpoints and test examples; calibration statistics use training images with evaluation preprocessing and no gradients.

A.3 Feature-Level Routing Diagnostic

For Figure 4, we measure the contribution of feature components outside a head's own-training PCA subspace to its predicted-class logit:

$$A(x) = \frac{w^\top (I - \Pi) h}{\sigma}, \tag{17}$$

where $h = E_T(x)$, Π projects onto the head's own-training principal subspace, w is the weight vector of its predicted class before calibration, and σ is its own-training maximum-logit standard deviation. We average this contribution separately over own-task and foreign-task test inputs for each head. Positive values indicate that the out-of-subspace component raises the predicted-class logit; negative values indicate that it lowers the logit.

B FDC Algorithm

Algorithm 1 summarizes FDC for the Cumulative-LoRA reference. The two phases collect statistics after each task and calibrate predictions at test time. The baseline training procedure is unchanged.

Algorithm 1 Full dual-view FDC: statistics collection and inference

Require: Task stream $\{\mathcal{D}_t\}_{t=1}^T$, pretrained encoder E_0, baseline learner; $\eta = 0.75$, $\gamma = 0.5$, unit correction and view weights.

Training-time statistics: initialize $\mathcal{F}^A, \mathcal{F}^0 \leftarrow \emptyset$.
for $t = 1, \ldots, T$ **do**
  Train task t with the baseline learner to obtain E_t, W_t, b_t.
  Extract $G_t^A = E_t(\mathcal{D}_t)$ and $G_t^0 = E_0(\mathcal{D}_t)$ without gradients.
  for $v \in \{A, 0\}$ **do**
    Fit the task mean, PCA basis, class prototypes, and own residual/affinity moments.
    For each $j < t$, append standardized residuals $u_j^v(G_t^v)$ to $\mathcal{F}^v$ with task-pair labels (j, t).
  end for
  Fold TSF into $\widetilde{W}_t = (1-\gamma)W_t + \gamma(W_t U_t^A)(U_t^A)^\top$.
  Compute $\sigma_t = \mathrm{Std}_{g \in G_t^A}[\max_c(\widetilde{W}_t g + b_t)_c]$ from the cached features.
  Retain task statistics; release current-task images and extracted features.
end for
If $T > 1$, for each view fit (μ_F^v, V_F^v) by Eq. (18) and τ^v as the median absolute foreign log-likelihood ratio; discard $\mathcal{F}^v$.
Test-time inference: compute $g^A = E_T(x)$ and $g^0 = E_0(x)$ once, sharing frozen backbone weights; cache each view's unit-normalized feature.
for $t = 1, \ldots, T$ **do**
  Compute $\widetilde{z}_t = \widetilde{W}_t g^A + b_t$ and $(\widetilde{s}_t, c_t) = (\max_c \widetilde{z}_{t,c}, \arg\max_c \widetilde{z}_{t,c})$.
  Compute PAC and RLC in both views using stored statistics and σ_t; set RLC to zero when $T = 1$ (no foreign-reference fit).
  Combine the corrections with $\widetilde{s}_t$ by Eq. (16) to obtain S_t.
end for
Select $\hat{t} = \arg\max_t S_t$ and return the global class indexed by $(\hat{t}, c_{\hat{t}})$.

B.1 Training-Time Statistics Collection

After learning task t, we extract $E_t(\mathcal{D}_t)$ and $E_0(\mathcal{D}_t)$ without gradients and fit each view's task mean, PCA basis, class prototypes, and own-task moments. The adapted-view statistics are indexed by A in Algorithm 1. In the checkpoint-based evaluation, they are reconstructed from each task's corresponding checkpoint and training partition, using evaluation preprocessing.

For the foreign residual reference, current-task features are evaluated against stored statistics of earlier tasks $j < t$. Let μ_{jt} and V_{jt} be the population mean and variance of the resulting standardized

residuals for task pair (j, t). With $\mathcal{P} = \{(j, t) : j < t\}$, each view fits

$$\mu_F = \frac{1}{|\mathcal{P}|} \sum_{(j,t)\in\mathcal{P}} \mu_{jt}, \qquad V_F = \frac{1}{|\mathcal{P}|} \sum_{(j,t)\in\mathcal{P}} \left[V_{jt} + (\mu_{jt} - \mu_F)^2\right]. \tag{18}$$

The foreign moments weight task pairs equally. The scale τ is the median absolute log-likelihood ratio over their concatenated residual samples. Current-task features can be released after statistics collection, and foreign residual scalars can be discarded after fitting τ.

B.2 Test-Time Calibration

Each test input is encoded once by E_T and, when enabled, once by E_0. TSF transforms the linear classifier weights as in Eq. (10); PAC and RLC use the original encoder features and the stored task statistics. Both views use the same adapted-head own-training score scale σ_t. The calibrated task logit determines the head, and that head's local argmax determines the class.

When TSF is disabled, both the local logits and their own-training score scales revert to Raw. For normalized or nonlinear baseline heads, the evaluator applies the TSF linear map at the head input and retains the method's subsequent scoring operations. Affinity, residual, and score moments use population standard deviations. Their scales, foreign variances, and LLR scales are floored at 10^{-6}; feature normalization and centered-energy denominators use 10^{-12} safeguards. With one task, RLC is skipped.

Store U_t^v rather than dense projectors Π_t^v; apply projections as $(hU_t^v)(U_t^v)^\top$. Adapted-only calibration omits all E_0 computations and statistics. Calibration using only E_0 still requires E_T for class prediction.

C Computational and Memory Costs of FDC

We measure Cumulative-LoRA on all five datasets using the seed-1 configurations, one NVIDIA A100 80GB PCIe GPU, FP32, and batch size 64. Statistics fitting uses four CPU threads. Data loading uses two workers, except for OmniBenchmark, which uses eight.

C.1 Training-Time Costs

Table 6 compares baseline optimization with optimization plus FDC statistics collection, reporting the baseline cost and the increase due to FDC. We replay 20 training epochs per task from the corresponding preceding checkpoint. Since FDC leaves optimization unchanged, the combined time is the sum of the measured optimization and statistics-collection times; combined peak GPU memory is the maximum across these sequential phases. The frozen encoder is kept off GPU during optimization. Timings include data loading and statistics fitting but exclude checkpoint loading, test evaluation, and artifact serialization. Peak GPU memory counts allocated tensors. FDC storage counts factorized PCA bases, prototypes, feature means, and scalar statistics, excluding model weights.

Table 6: Training-stage time, GPU memory, and FDC storage.

	Time (min)		Peak GPU memory (GiB)		FDC storage
Dataset	Baseline	+ FDC	Baseline	+ FDC	(MiB)
CIFAR-100	132.2	+5.1	6.51	+0.00	11.86
ImageNet-A	18.7	+1.1	6.51	+0.00	10.05
ImageNet-R	64.8	+5.1	6.51	+0.00	11.30
CUB-200	29.3	+3.0	6.51	+0.00	7.68
OmniB.	239.3	+9.2	6.51	+0.00	12.03

C.2 Inference-Time Costs

Table 7 compares Cumulative-LoRA with single-view (E_T) and dual-view (E_T, E_0) FDC on the same final checkpoint. We time three repeats of 20 batches after 10 warm-up batches, using 64 test images preloaded on GPU and synchronizing at each timing boundary. Image loading is excluded. Latency columns report total batch time; memory columns report the baseline peak and increases over it. TSF is folded into the heads before timing; Raw retains no FDC-specific GPU tensors. Dual reuses the frozen backbone weights for its two sequential forward passes, disabling LoRA for E_0. Each view's normalized feature is reused across task heads. Parentheses in the latency columns give increases over the baseline. Single adds 3.7–5.9 MiB and 0.7–1.7% latency. Dual adds 7.8–12.1 MiB, with latency 1.88–1.90 times the baseline.

Table 7: Inference latency and GPU memory at batch size 64.

	Total latency (ms/batch)			Peak GPU memory (MiB)		
Dataset	Baseline	Single	Dual	Baseline	Δ Single	Δ Dual
CIFAR-100	170.5	173.3 (+1.7%)	323.1 (+89.5%)	980.9	+4.7	+11.9
ImageNet-A	171.0	172.5 (+0.8%)	321.5 (+88.0%)	981.4	+5.0	+10.1
ImageNet-R	170.5	171.7 (+0.7%)	320.1 (+87.7%)	981.4	+5.9	+11.3
CUB-200	171.3	174.1 (+1.6%)	324.2 (+89.3%)	981.4	+3.7	+7.8
OmniB.	169.9	171.3 (+0.8%)	319.2 (+87.9%)	982.0	+5.7	+12.1

TSF and adapted-view RLC share a stored PCA basis. For feature dimension d, total class count C, and per-task/view rank r_t^v, statistical storage scales as $\mathcal{O}(d\sum_{v,t} r_t^v + Cd + Td)$, excluding encoder and classifier weights.

D FDC Component Ablation Studies

All ablations use the same Cumulative-LoRA checkpoints and test examples. Unless varied, we use $\eta = 0.75$, $\gamma = 0.5$, $K = 1$, and unit correction and view weights.

Table 8 reports Raw accuracy and paired gains over Raw for all eight component combinations, with dual-view RLC and PAC. Values are means and sample standard deviations over three seeds; gain deviations are computed after subtracting Raw within each seed. Each component alone improves Raw on all five datasets, and the full combination performs best. Component gains interact: adding PAC to TSF+RLC changes the gain on ImageNet-R by only 0.01 pp, compared with 3.83 pp on CUB-200.

Single- and dual-view calibration. The final two rows of Table 8 retain TSF and use PAC and RLC from either E_T or E_0 alone. TSF acts on the adapted head; E_0-only calibration still uses E_T for class prediction. Combining both views improves over adapted-view calibration by 1.44, 2.05, and 1.90 pp on CIFAR-100, CUB-200, and OmniBenchmark, respectively; ImageNet-A and ImageNet-R change little.

E Hyperparameter Analysis of TSF and PAC

We analyze the TSF attenuation and PCA retained-energy threshold, followed by the PAC neighborhood size, using five datasets and three training seeds.

E.1 TSF Hyperparameters

We vary $\gamma \in \{0, 0.25, 0.375, 0.5, 0.625, 0.75, 1\}$ at $\eta = 0.75$ and $\eta \in \{0.5, 0.65, 0.70, 0.75, 0.80, 0.85, 0.95\}$ at $\gamma = 0.5$, using full dual-view FDC on the same Cumulative-LoRA checkpoints for all five datasets and three seeds. Other settings remain fixed. Statistics and calibration scales are fitted for each setting using the same causal collection protocol, without retraining the models. Curves show the three-seed mean with one sample standard deviation shaded. Vertical axes are scaled independently, and dotted lines mark the shared defaults.

Table 8: Component and view ablations: Raw accuracy (%) and gains over Raw (pp).

Method	CIFAR-100	ImageNet-A	ImageNet-R	CUB-200	OmniB.
Raw	84.34±0.15	54.11±0.17	75.91±0.34	73.55±1.70	67.15±0.21
TSF	+0.85±0.22	+3.29±0.43	+1.45±0.28	+2.19±0.84	+0.37±0.32
Dual PAC	+2.89±0.21	+2.96±0.23	+1.63±0.21	+8.14±1.64	+4.83±0.26
Dual RLC	+2.76±0.14	+3.45±1.25	+2.34±0.20	+4.95±0.83	+5.02±0.12
TSF + Dual PAC	+3.21±0.27	+4.56±0.42	+2.63±0.25	+9.44±1.46	+4.90±0.23
TSF + Dual RLC	+3.21±0.16	+5.66±0.86	+3.57±0.14	+7.05±0.84	+5.03±0.19
Dual PAC + Dual RLC	+4.18±0.21	+3.91±0.40	+2.73±0.26	+9.85±1.36	+6.23±0.27
Full (Dual FDC)	+4.28±0.13	+5.97±0.91	+3.58±0.13	+10.88±1.48	+6.35±0.31
FDC (E_T calibration)	+2.84±0.35	+5.92±0.59	+3.58±0.32	+8.84±1.57	+4.44±0.12
FDC (E_0 calibration)	+3.76±0.12	+4.50±0.56	+2.09±0.12	+8.71±0.92	+5.70±0.04

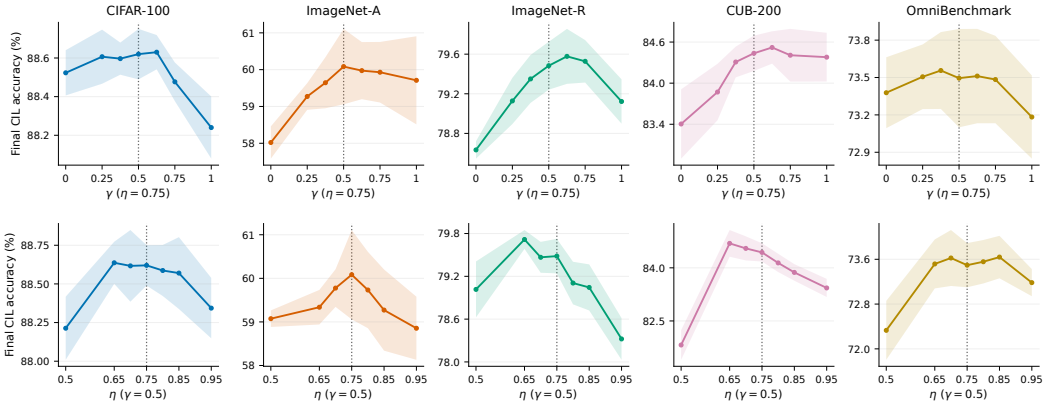

Figure 7: TSF sensitivity: γ (top) and η (bottom), one dataset per column.

TSF attenuation. In Figure 7, the default $\gamma = 0.5$ averages 77.22% across datasets, exceeding both no attenuation (76.39%) and full attenuation (76.93%); $\gamma = 0.75$ is only 0.06 pp lower.

PCA retained energy. The default $\eta = 0.75$ averages 77.22%, close to 77.18% at $\eta = 0.65$. Dataset-specific maxima differ. Both defaults were fixed before this analysis, without dataset-specific selection; these results do not establish universal optimality.

E.2 PAC Neighborhood Size

For each candidate task, PAC averages its K highest class-prototype cosine similarities:

$$A_{t,K}(x) = \frac{1}{K} \sum_{c \in \mathcal{N}_{t,K}(x)} \frac{g(x)^\top p_{t,c}}{\|g(x)\|_2}. \tag{19}$$

At $K = 1$, this reduces to Eq. (13). We compare two reference distributions for standardizing $A_{t,K}$: *Distributional matching* uses own-task training statistics recomputed for each K, whereas *Nearest* uses statistics across candidate tasks for the current input. Both apply the same $\sigma_t \tanh(q_{t,K})$ correction; prototypes, TSF, dual-view RLC, head-score scales, and local predictions remain fixed.

Figure 8 sweeps all feasible K, showing three-seed means and sample standard deviations. Matching performs best at $K = 1$ and exceeds Nearest on all five datasets there. Nearest can lead at larger K, especially on OmniBenchmark, but remains below the default matching result there.

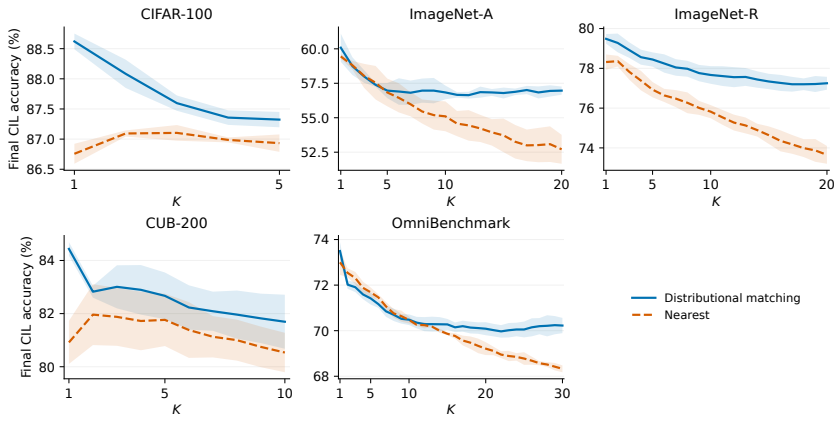

Figure 8: PAC accuracy across neighborhood sizes.

F Average Incremental Accuracy

For CIL accuracy A_k over all encountered classes after task k, average incremental accuracy is $\text{AIA} = T^{-1} \sum_{k=1}^{T} A_k$. Table 9 reports the mean and sample standard deviation over three seeds for the compared shared-encoder methods. FDC is evaluated at the final checkpoint, so its AIA is not reported.

Table 9: Average incremental accuracy (%), mean ± standard deviation.

Method	CIFAR-100	ImageNet-A	ImageNet-R	CUB-200	OmniB.
Cumulative-LoRA	90.56±0.16	63.20±0.35	82.09±0.31	82.76±1.36	77.17±0.13
EWC-LoRA	88.88±0.40	61.39±0.40	81.14±0.24	80.95±0.54	76.58±0.27
SD-LoRA	90.81±0.44	62.81±0.64	81.94±0.08	80.96±1.79	77.53±0.17
SplitLoRA	92.49±0.29	63.83±0.63	84.48±0.18	82.55±0.26	78.66±0.30
InfLoRA	90.80±0.23	61.67±1.84	81.65±0.16	82.65±1.17	76.23±0.38
InfLoRA+CA	91.33±0.12	68.42±0.57	82.36±0.14	90.43±0.37	80.36±0.09
LoRA-Subtraction	63.06±2.27	67.03±0.86	81.94±0.25	92.08±0.22	67.07±0.63
LoDA	92.51±0.35	72.80±0.61	86.63±0.05	85.89±0.18	78.98±0.06
RSIAT	77.13±0.53	72.73±0.36	77.43±0.09	92.86±0.17	77.34±0.24

G Plug-and-Play Component Configurations

Table 10 gives the configurations used in Table 3. Following Section 5.3, each method enables components with positive standalone mean gains on all five datasets, then uses this configuration across datasets and seeds. This selection uses the reported evaluation results; Cumulative-LoRA enables all components.

The TSF, E_0, and E_T switches denote filtering, pretrained-view calibration, and adapted-view calibration (1: enabled; 0: disabled). Each calibration view includes RLC and PAC. Where selection disables a component, a second row reports full FDC. Gains are paired with Raw within each seed and reported as mean ± sample standard deviation; Avg. weights datasets equally.

Table 10: FDC component configurations and paired accuracy gains (pp).

Baseline	TSF	E_0	E_T	CIFAR-100	ImageNet-A	ImageNet-R	CUB-200	OmniB.	Avg.
EWC-LoRA	1	1	1	+4.81±0.41	+4.94±0.63	+3.19±0.23	+12.54±0.35	+7.68±0.17	+6.63
SD-LoRA	1	1	1	+4.33±0.60	+5.71±0.95	+3.21±0.15	+11.32±0.88	+6.26±0.72	+6.16
SplitLoRA	0	0	1	+1.62±0.16	+2.68±0.55	+1.46±0.09	+5.71±0.32	+2.74±0.21	+2.84
	1	1	1	+1.76±0.22	+1.95±0.44	+0.75±0.14	+9.10±0.35	+3.48±0.23	+3.41
InfLoRA	1	1	1	+4.21±0.28	+6.50±1.15	+3.74±0.09	+10.56±0.51	+7.66±0.72	+6.53
InfLoRA+CA	1	1	1	+4.55±0.94	+2.13±0.47	+2.47±0.09	+2.23±0.47	+3.22±0.50	+2.92
LoRA-Subtraction	0	0	1	+4.41±2.98	+0.22±0.55	+0.79±0.16	+0.42±0.36	+6.31±0.15	+2.43
	1	1	1	+6.28±2.77	−2.09±0.21	−0.32±0.20	−0.99±0.28	+11.09±0.23	+2.80
LoDA	0	0	1	+1.55±0.17	+1.98±0.17	+0.72±0.21	+5.09±0.07	+2.94±0.28	+2.45
	1	1	1	+1.07±0.17	+1.71±0.37	+0.11±0.05	+5.08±0.17	+3.53±0.37	+2.30
RSIAT	0	1	0	+12.12±0.45	+0.86±0.46	+6.18±0.23	+1.12±0.19	+5.46±0.24	+5.15
	1	1	1	+13.07±0.19	−0.09±0.83	+6.68±0.55	−0.18±0.10	+4.71±0.53	+4.84