

Compensating Distribution Drifts in Class-incremental Learning of Pre-trained Vision Transformers

Xuan Rao¹, Simian Xu², Zheng Li³, Bo Zhao^{1*}
Derong Liu⁴, Mingming Ha⁵, Cesare Alippi^{6,7}

¹ School of Systems Science, Beijing Normal University ² School of Physics, Peking University

³ School of Automation and Intelligent Manufacturing, Southern University of Science and Technology

⁴ School of Artificial Intelligence, Anhui University ⁵ Kuaishou Technology

⁶ Politecnico di Milano ⁷ Università Della Svizzera italiana

{raoxuan98@mail.bnu.edu.cn, zhaobo@bnu.edu.cn, hamingming@kuaishou.com}

Abstract

Recent advances have shown that sequential fine-tuning (SeqFT) of pre-trained vision transformers (ViTs), followed by classifier refinement using approximate distributions of class features, can be an effective strategy for class-incremental learning (CIL). However, this approach is susceptible to distribution drift, caused by the sequential optimization of shared backbone parameters. This results in a mismatch between the distributions of the previously learned classes and that of the updated model, ultimately degrading the effectiveness of classifier performance over time. To address this issue, we introduce a latent space transition operator and propose Sequential Learning with Drift Compensation (SLDC). SLDC aims to align feature distributions across tasks to mitigate the impact of drift. First, we present a linear variant of SLDC, which learns a linear operator by solving a regularized least-squares problem that maps features before and after fine-tuning. Next, we extend this with a weakly nonlinear SLDC variant, which assumes that the ideal transition operator lies between purely linear and fully nonlinear transformations. This is implemented using learnable, weakly nonlinear mappings that balance flexibility and generalization. To further reduce representation drift, we apply knowledge distillation (KD) in both algorithmic variants. Extensive experiments on standard CIL benchmarks demonstrate that SLDC significantly improves the performance of SeqFT. Notably, by combining KD to address representation drift with SLDC to compensate distribution drift, SeqFT achieves performance comparable to joint training across all evaluated datasets. Code: <https://github.com/raoxuan98-hash/sldc.git>.

Introduction

There is a growing interest in applying continual learning (CL) to pre-trained models (PTMs) (Dosovitskiy et al. 2021; Radford et al. 2021) by leveraging their rich representations (Zheng et al. 2023; Li et al. 2024; Zhou et al. 2025). Researchers have shown that sequentially fine-tuning (SeqFT) the backbones of pre-trained vision transformers (ViTs) on downstream tasks, followed by the refinement of the classifier using the approximate distributions of class-wise deep features, offers an effective strategy to class in-

cremental learning (CIL) (Zhang et al. 2023, 2024; Marouf et al. 2024). Notably, unlike methods that introduce task-specific lightweight adaptation to mitigate interference from new tasks (Li et al. 2024; Wang et al. 2025), SeqFT is more computationally efficient, as it eliminates the need of task identification (Zhang et al. 2024; Marouf et al. 2024).

However, the sequential optimization of shared parameters inevitably introduces representation drifts, which leads to a mismatch between the learned distributions of previous classes and those of the updated model.

Unlike previous works which mitigate distribution drifts through approaches like distillation, model ensemble, and gradient projection (Zhao et al. 2024; Xiao et al. 2023; Lu et al. 2024), our work takes a novel perspective by focusing on compensating for the negative effects of representation drifts once they occur. To this end, we resolve to *model the transformation that occurs in the feature space between consecutive tasks*. In particular, the latent space transition operator that captures how the feature mapping function evolves during task adaptation is defined as:

Definition 1 (Latent Space Transition Operator). *A latent space transition operator is a mapping $\mathcal{P}_{t-1 \rightarrow t} : \mathcal{F}_{t-1} \rightarrow \mathcal{F}_t$, where $\mathcal{F}_{t-1} : \mathcal{X} \rightarrow \mathbb{R}^d$ and $\mathcal{F}_t : \mathcal{X} \rightarrow \mathbb{R}^d$ are (here) neural network-based feature extractors (e.g., backbones of ViTs) that map inputs from the input space $\mathcal{X}$ to a d -dimensional feature space at tasks $t-1$ and t , respectively.*

Ideally, when the approximate distributions are multi-variate Gaussian, the operator $\mathcal{P}_{t-1 \rightarrow t}$ enables the propagation of their first-order (mean) and second-order (covariance) moments from the previous feature space to the new one, which enables consistent classifier refinement despite the representation drift. However, learning the exact operator $\mathcal{P}_{t-1 \rightarrow t}$ would typically require access to the entire input space $\mathcal{X}$ (e.g., the normalized RGB space), which is not available in exemplar-free CIL settings where previous data cannot be preserved. To overcome this limitation, we introduce a practical approximation strategy that estimates $\mathcal{P}_{t-1 \rightarrow t}$ using only the current task data $\mathcal{D}_t$ and the frozen models $\mathcal{F}_{t-1}$ and $\mathcal{F}_t$.

Accordingly, the Sequential Learning with Drift Compensation (SLDC) method is proposed. First, we propose the α_1 -SLDC method, which learn a linear operator by solving a

*Corresponding author

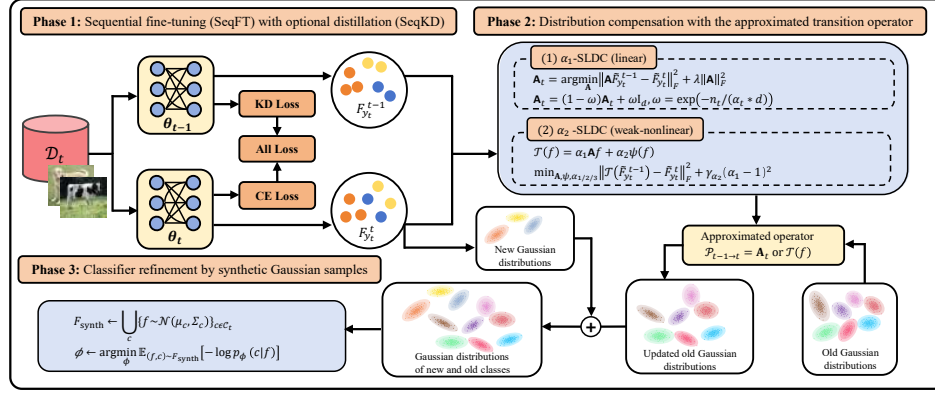


Figure 1: Overview of the SLDC framework. The framework consists of three phases: (1) Sequential fine-tuning with optional distillation (SeqFT/SeqKD); (2) Distribution compensation using an approximated transition operator, either linear (α_1 -SLDC) or weak-nonlinear (α_2 -SLDC), to align (compensate) previous feature distributions with the new one; (3) Classifier refinement using synthetic Gaussian features sampled from the compensated Gaussian distributions.

regularized least-squares problem between the deep features of models $\mathcal{F}_{t-1}$ and $\mathcal{F}_t$ on $\mathcal{D}_t$ ¹. The empirical results show that the linear operator can compensate for the distribution drift appropriately, but it still yields large prediction residuals when predicting the post-optimization deep features, implying that a nonlinear mapping is required. However, the direct implementation of popular nonlinear transformation such as multilayer perceptrons (MLPs) leads to overfitting and produces distributions that are less accurate than those obtained with linear operators.

Motivated by these empirical observations, *we assume that an ideal operator approximation lies between purely linear and fully nonlinear transformations*. Correspondingly, we propose the α_2 -SLDC method by constructing a weak-nonlinear transformation to learn the transition operator. Building upon $\alpha_{1,2}$ -SLDC methods, the distillation-enhanced SLDC variants, $\beta_{1,2}$ -SLDC, are further developed by constraining model’s representation updatings with knowledge distillation (KD).

Notably, the evaluation results show that the combination of distillation (to preserve previous knowledge) with SLDC (to compensate for distribution drifts) enables PTM-based CIL to nearly match the performance of joint training (i.e., training the model using all training data simultaneously), which can be regarded as an empirical upper bound of optimal performance of CIL (Sun et al. 2025). It emerges how SLDC achieves near-parity on 10-task CIL scenarios with joint training across two PTMs and four different datasets, with accuracy discrepancies within +0.50% to -3.29%, proving the effectiveness of the proposed approach. The novel contributions are:

1. An effective novel CIL methodology is proposed based on a learned transition operator that models the feature space evolution across successive tasks.
2. Two novel learnable transition operators, the α_1 -SLDC and α_2 -SLDC, along with their distillation-enhanced

¹Here, α/β denote without or with distillation; subscripts 1/2 denote linear or weak-nonlinear SLDC methods.

variants $\beta_{1,2}$ -SLDC, are developed based on linear and weak-nonlinear transformations, respectively. The proposed methods can be implemented and integrated with existing approaches.

Methodologies

SeqFT-based CIL with pre-trained ViTs and classifier refinement

CIL formalization. A sequence of training datasets is defined as $\mathcal{D} = \{\mathcal{D}_1, \dots, \mathcal{D}_T\}$, where the t th dataset is $\mathcal{D}_t = \{(x_{(t,n)}, y_{(t,n)})\}_{n=1}^{n_t}$. Each $\mathcal{D}_t$ contains n_t pairs of input samples $x_{(t,n)} \in \mathcal{X}$ and their corresponding labels $y_{(t,n)} \in \mathcal{Y}_t$, where $\mathcal{X}$ represents the shared input space and $\mathcal{Y}_t$ represents the label space of task t . Specifically, $\mathcal{Y}_t \cap \mathcal{Y}_{t'} = \emptyset$ for $t \neq t'$. The cumulative set of observed classes up to task t is denoted as $\mathcal{C}_t = \bigcup_{t'=1}^t \mathcal{Y}_{t'}$.

ViT architectures. The ViT is defined as $\mathcal{G}_\varphi(x) = \mathcal{C}_\phi(\mathcal{F}_\theta(x))$, where $\mathcal{F}_\theta : \mathcal{X} \rightarrow \mathbb{R}^d$ is the pre-trained backbone, $\mathcal{C}_\phi : \mathbb{R}^d \rightarrow \mathbb{F}^{|\mathcal{C}_t|}$ is a linear classifier, and $\varphi = \{\phi, \theta\}$ denotes all trainable parameters (Dosovitskiy et al. 2021). In this paper, we adopt the configuration of SLCA++ and fine-tune the backbones of ViTs using low-rank adaptation (LoRA) (Hu et al. 2022; Zhang et al. 2024), thus θ denotes the parameters of the LoRA adapters.

For any label subspace $\mathcal{S} \subseteq \mathcal{C}_t$ (e.g., $\mathcal{Y}_t$ or $\mathcal{C}_t$), the ViT’s softmax output is given by

$$p_\varphi(x; \mathcal{S})_i = \frac{\exp([\mathcal{G}_\varphi(x)]_i)}{\sum_{j=1}^{|\mathcal{S}|} \exp([\mathcal{G}_\varphi(x)]_j)}, \quad (1)$$

where $i \in \{1, \dots, |\mathcal{S}|\}$. At the task t , the model is trained by minimizing the task-specific cross-entropy loss

$$\mathcal{L}_{\text{CE}}(\varphi; \mathcal{D}_t) = -\frac{1}{B} \sum_{n=1}^B \log p_\varphi(x_n; \mathcal{Y}_t)_{y_n} \quad (2)$$

where $(x_n, y_n) \sim \mathcal{D}_t$, and B denotes the batch size.

Post-hoc classifier refinement. After the training procedure on task t , for each new class $c \in \mathcal{Y}_t$, we assume that its deep features under the PTM mapping $\mathcal{F}_\theta$ follow a Gaussian distribution, and its deep feature distribution is approximated by

$$\mu_c = \frac{1}{n_c} \sum_{i=1}^{n_c} f_c^{(i)}, \quad (3)$$

$$\Sigma_c = \frac{1}{n_c} \sum_{i=1}^{n_c} (f_c^{(i)} - \mu_c)(f_c^{(i)} - \mu_c)^\top, \quad (4)$$

where $f_c^{(i)} = \mathcal{F}_\theta(x^{(i)})$ is the feature of sample $x^{(i)}$ with label c . Let $\mathcal{H}_t = \{\mathcal{N}(\mu_c, \Sigma_c) \mid c \in \mathcal{C}_t\}$ be the set of all Gaussian distributions up to task t . The classifier is refined in a post-hoc manner after learning each new task by the synthetic samples from $\mathcal{H}_t$ to improve cross-task decision boundaries as

$$\min_{\phi} \mathcal{L}_{\text{CR}}(\phi; \mathcal{H}_t) = -\frac{1}{|\mathcal{C}_t|} \sum_{c \in \mathcal{C}_t} \mathbb{E}[\log p_\phi(f_{\text{synth}}; \mathcal{C}_t)_c], \quad (5)$$

where $f_{\text{synth}} \sim \mathcal{N}(\mu_c, \Sigma_c)$, and $p_\phi(f; \mathcal{C}_t)$ denotes the classifier's softmax output over $\mathcal{C}_t$.

SLDC for Distribution Drift Compensation

Figure 1 provides a visual illustration of SLDC's underlying mechanisms. Let $\mathcal{F}_{\theta_{t-1}}$ and $\mathcal{F}_{\theta_t}$ be the ViT backbones after training on tasks $t-1$ and t , respectively. Given the current task dataset $\mathcal{D}_t$, we define $F_{\mathcal{Y}_t}^{t-1} = [\mathcal{F}_{\theta_{t-1}}(x_{t,1}), \dots, \mathcal{F}_{\theta_{t-1}}(x_{t,n_t})] \in \mathbb{R}^{d \times n_t}$ and $F_{\mathcal{Y}_t}^t = [\mathcal{F}_{\theta_t}(x_{t,1}), \dots, \mathcal{F}_{\theta_t}(x_{t,n_t})] \in \mathbb{R}^{d \times n_t}$ by the feature matrices extracted by backbones $\mathcal{F}_{\theta_{t-1}}$ and $\mathcal{F}_{\theta_t}$ on $\mathcal{D}_t$, respectively.

Derivation of linear α_1 -SLDC. The α_1 -SLDC estimates a linear transition operator by solving a least-square problem between normalized features. Specifically, let $\tilde{F}_{\mathcal{Y}_t}^{t-1} \in \mathbb{R}^{d \times n_t}$ and $\tilde{F}_{\mathcal{Y}_t}^t \in \mathbb{R}^{d \times n_t}$ be the column-wise L_2 -normalized versions of $F_{\mathcal{Y}_t}^{t-1}$ and $F_{\mathcal{Y}_t}^t$, respectively. The linear operator $\mathbf{A}_t \in \mathbb{R}^{d \times d}$ for approximating $\mathcal{P}_{t-1 \rightarrow t}$ is obtained by solving the regularized least-square solution

$$\mathbf{A}_t = \arg \min_{\mathbf{A}} \|\mathbf{A} \tilde{F}_{\mathcal{Y}_t}^{t-1} - \tilde{F}_{\mathcal{Y}_t}^t\|_F^2 + \gamma_{\alpha_1} \|\mathbf{A}\|_F^2 \quad (6)$$

$$= \tilde{F}_{\mathcal{Y}_t}^t (\tilde{F}_{\mathcal{Y}_t}^{t-1})^\top \left(\tilde{F}_{\mathcal{Y}_t}^{t-1} (\tilde{F}_{\mathcal{Y}_t}^{t-1})^\top + \gamma_{\alpha_1} I_d \right)^{-1}, \quad (7)$$

where γ_{α_1} is the regularization coefficient and $I_d \in \mathbb{R}^{d \times d}$ is the identity matrix. In addition, there are some cases where the number of task-specific samples n_t is too small to obtain a robust estimation of the linear operator. To avoid this problem, we regularize $\mathbf{A}_t$ by a heuristic re-weighting process based on sample complexity as

$$\mathbf{A}_t = (1 - w) \mathbf{A}_t + w I_d, \quad (8)$$

where $w = \exp\left(-\frac{n_t}{\alpha_{\text{temp}} d}\right)$ and α_{temp} are the weighting and temperature coefficients, respectively.

Once $\mathbf{A}_t$ is obtained, the Gaussian distributions of previous tasks' classes $c \in \mathcal{C}_{t-1}$ are compensated by

$$\mu_c \leftarrow \mathbf{A}_t \mu_c \quad \Sigma_c \leftarrow \mathbf{A}_t \Sigma_c \mathbf{A}_t^\top. \quad (9)$$

This process is applied recursively as new tasks arrive. In Statement 1 of the appendix, it is proved that the above updating formulations follow the close-formed solution to the linear transformation of a Gaussian distribution.

Derivation of weak-nonlinear α_2 -SLDC. Although the task-wise linear operator $\mathbf{A}_t$ in α_1 -SLDC can mitigate distribution drifts to some extent, residual errors between the predicted and actual features still remain. While nonlinear MLPs could address the under-fitting problem, they suffer from over-fitting and yield less accurate transformed distributions than linear transformations.

Based on these empirical observations, we assume that the ideal transition operator $\mathcal{P}_{t-1 \rightarrow t}$ for SeqFT-based CIL with pre-trained ViTs resides between purely linear and fully nonlinear transformations, i.e., $\mathcal{P}_{t-1 \rightarrow t}$ is weak-nonlinear.

Motivated by the assumption, the α_2 -SLDC is proposed by defining the weak-nonlinear transformation

$$\mathcal{T}(f) = c_1 \mathbf{A} f + c_2 \psi(f). \quad (10)$$

Specifically, $c_{1/2}$ are learnable contribution coefficients which satisfies $c_{1/2} \geq 0$ and $c_1 + c_2 = 1$. In particular, we instantiate $\mathbf{A}$ as a learnable matrix and $\psi(f)$ as a two-layer MLP with ReLU activation. To optimize $\mathcal{T}(f)$, a regularized optimization objective is defined by

$$\min_{\mathbf{A}, \psi, c_{1/2}} \left\| \mathcal{T}(\tilde{F}_{\mathcal{Y}_t}^{t-1}) - \tilde{F}_{\mathcal{Y}_t}^t \right\|_F^2 + \gamma_{\alpha_2} (c_1 - 1)^2, \quad (11)$$

where $\gamma_{\alpha_2} (c_1 - 1)^2$ is the regularization term controlling the contribution of nonlinear $\psi(f)$.

In practice, the optimization process for $\mathcal{T}(f)$ is end-to-end by the gradient optimizer, and the training details are presented in the experiment section. Specifically, in Statements 3 and 4, some theoretical claims on the characteristics of transition operator are given based on the neural tangent kernel (NTK) theory (Jacot, Gabriel, and Hongler 2018).

After obtaining the weak-nonlinear transformation $\mathcal{T}(f)$, the Monte Carlo sampling is used to estimate the updated Gaussian distributions for previous classes $c \in \mathcal{C}_{t-1}$. Specifically, for each class c , we generate $N \gg d$ synthetic samples from its original Gaussian distribution $\mathcal{N}(\mu_c, \Sigma_c)$

$$f_c^{(i)} \sim \mathcal{N}(\mu_c, \Sigma_c), \quad i = 1, \dots, N \quad (12)$$

These samples are then compensated by the weak-nonlinear transformation as

$$\tilde{f}_c^{(i)} = \mathcal{T}(f_c^{(i)}), \quad i = 1, \dots, N \quad (13)$$

Hereafter, the mean μ_c and covariance Σ_c for $c \in \mathcal{C}_{t-1}$ are compensated by re-calculating (3) and (4) using the transformed samples in (13). Finally, distributions of old classes in $\mathcal{H}_t$ are replaced by the updated ones before executing classifier refinement.

Distillation-enhanced SLDC variants. Typically, the unconstrained optimization for ViT backbones makes the performance of SeqFT for CIL sensitive to several hyperparameters such as batch size, learning rate and tuning epochs. Considering these issues, the distillation-enhanced variants of $\alpha_{1,2}$ -SLDC are proposed by incorporating a feature-based distillation loss, i.e.,

$$\mathcal{L}_{\text{KD}} = -\frac{1}{B} \sum_{n=1}^B \|\mathcal{F}_{\theta_{t-1}}(x_n) - \mathcal{F}_{\theta}(x_n)\|^2, \quad (14)$$

In addition, a regularization loss is considered to maintain the L_2 -norm of feature vectors as

$$\mathcal{L}_{\text{Norm}} = -\frac{1}{B} \sum_{n=1}^B (\|\mathcal{F}_{\theta_{t-1}}(x_n)\| - \|\mathcal{F}_{\theta}(x_n)\|)^2, \quad (15)$$

Consequently, the overall loss for optimizing the ViT backbone in β -SLDC is

$$\mathcal{L}_{\text{All}} = \mathcal{L}_{\text{CE}} + \gamma_{\text{kd}} \mathcal{L}_{\text{KD}} + \gamma_{\text{norm}} \mathcal{L}_{\text{Norm}}, \quad (16)$$

where γ_{KD} and γ_{Norm} are the balance coefficients. In particular, we refer $\beta_{1,2}$ -SLDC to the distillation-enhanced $\alpha_{1,2}$ -SLDC variants, respectively. We also refer SeqKD to the distillation-enhanced SeqFT in the following sections.

Improved operator estimation with auxiliary unlabeled data. In certain scenarios, limited dataset size and insufficient sample diversity can lead to inaccurate approximations of the transfer operator. To address this challenge, this paper proposes auxiliary data enrichment (ADE) to improve the prediction by leveraging unlabeled auxiliary data from arbitrary sources. *Crucially, ADE operates without requiring labeled data and remains consistent with the exemplar-free continual learning (CIL) framework since it does not preserve any task-relevant data from previous tasks.*

Related Works

Based on strategies for dealing with representation drifts, existing research on ViT-based CIL approaches can be divided into four types.

The first category optimizes task-specific adapters for each new task and selects appropriate adapters during inference based on the characteristics of test samples (Wang et al. 2025; Li et al. 2024). Typically, these methods decompose the prediction process into two hierarchical stages, i.e., the task identity prediction and the within-task label prediction using the corresponding adapter. However, these methods rely heavily on task identity prediction accuracy, incur high computational overhead due to repeated forward passes, and face linearly scaling storage demands for adapters.

The second approach trains a shared backbone or lightweight adapter across tasks by using techniques like reduced learning rates, distillation, model merging, or gradient projection to mitigate catastrophic forgetting (Zhang et al. 2023; Gao et al. 2023; Marouf et al. 2024; Lu et al. 2024). For example, slow learner with classifier alignment (SLCA) adapts ViT backbones with lower learning rates to preserve pre-trained knowledge (Zhang et al. 2023). Enhancements like continual model averaging (CoMA) and

continual fisher-weighted model averaging (CoFiMA) improve SLCA by averaging current and past models (Marouf et al. 2024), which proportionally average current and past models to enhance SLCA’s performance. SLCA++ further integrates lightweight adapters in SLCA, and achieves comparable results with minimal parameter optimization (Zhang et al. 2024). However, these methods remain vulnerable to representation drifts from progressive optimization.

The third approach combines multiple shared adapters with instance-level feature adaptation. Learning to prompt (L2P) uses a fixed prompt pool and learnable query vectors to dynamically select prompts based on sample features (Wang et al. 2022b). DualPrompt extends L2P with supplementary task-specific prompts (Wang et al. 2022a), while CODA-Prompt employs an input-dependent key-value mechanism to achieve finer-grained prompts (Smith et al. 2023).

The fourth category freezes PTMs and leverages the pre-trained features only. First session adaptation (FSA) optimizes PTMs only in the first task and applies exemplar-free CIL by incremental linear discriminant analysis (LDA) (Panos et al. 2023). RanPAC enhances FSA by projecting ViT features into a 10,000-dimensional space with a non-linear ReLU mapping (McDonnell et al. 2023). LayUP enhances RanPAC’s performance by concatenating outputs from multiple feature layers (Ahrens et al. 2024).

Beyond PTM-based CIL, there were methods compensating the distribution drifts during CIL (Yu et al. 2020; Gomez-Villa et al. 2024). For example, AddGauss tackles task-recency bias by adapting class covariance matrices with nonlinear mappings (Rypešć et al. 2024). Meanwhile, DPCR quantifies feature space semantic drifts using linear task-wise semantic drift projections and categorical information projections (He et al. 2025), DS-AL constructs an analytic incremental classifier based on the recursive least-squares method (Zhuang et al. 2024). Notably, SLDC methods take insights from AddGauss and investigate the efficacy of linear, weak-nonlinear, and nonlinear transformations in the context of PTM-based CIL research.

Experiment Evaluations

Benchmarks. To comprehensively evaluate the CIL performance, we conduct experiments on four widely-used benchmark datasets, i.e., CIFAR-100 (Krizhevsky and Hinton 2009), ImageNet-R (Hendrycks et al. 2021), CUB-200 (Wah et al. 2011), and Cars-196 (Krause et al. 2013). Each dataset is uniformly partitioned into 10 disjoint tasks without any emphasis. The CIFAR-100 comprises 100 classes of natural images, with 500 training samples per class. The ImageNet-R contains images from 200 classes. Totally, it has 24,000 and 6,000 samples for training and test sets, respectively. Specifically, ImageNet-R is challenging for the PTMs because its images are either hard examples from ImageNet-21K or new images in diverse styles. CUB-200 contains 200 bird species with approximately 60 images per class. The training and test sets are split evenly. Cars-196 consists of 196 car types. It has 8,144 training and 8,040 testing images totally. Following the established protocols,

Table 1: State-of-the-art CIL performance comparison across CUB-200, Cars-196, CIFAR-100, and ImageNet-R by a self-supervised pre-trained ViT-B/16 with the MoCo-V3 approach.

Method	CUB-200		Cars-196		CIFAR-100		ImageNet-R	
	Last-Acc	Inc-Acc	Last-Acc	Inc-Acc	Last-Acc	Inc-Acc	Last-Acc	Inc-Acc
Joint-Training	81.82±0.29	-	81.16±0.06	-	88.86±0.14	-	75.95±0.23	-
BiC	74.39±1.12	82.13±0.33	65.57±0.93	73.95±0.29	80.57±0.86	89.39±0.33	57.36±2.68	68.07±0.22
LwF	61.66±1.95	73.90±1.91	52.45±0.48	63.87±0.31	77.94±1.00	86.90±0.90	60.74±0.30	68.55±0.65
RanPAC	74.43±0.43	83.63±0.01	63.21±0.02	74.01±0.47	86.47±0.52	90.81±1.05	69.11±0.69	75.20±0.34
SLCA	73.01±0.16	82.13±0.34	66.04±0.08	72.59±0.04	85.27±0.08	89.51±1.04	68.07±0.21	73.04±0.56
SLCA++	75.48±0.31	82.94±0.73	69.71±0.10	75.67±0.32	84.77±0.18	89.53±0.98	69.01±0.42	74.75±0.69
CoMA	75.12±0.27	82.76±0.16	67.48±0.19	74.90±0.87	86.59±0.51	91.02±0.47	69.33±0.22	75.64±0.13
CoFiMA	77.65±0.18	83.54±0.16	69.51±0.16	76.21±0.83	87.44±0.47	91.13±0.53	70.87±0.31	76.09±0.78
SeqFT	64.40±1.65	77.77±0.61	60.42±1.50	72.12±0.63	73.36±0.90	80.40±2.01	61.37±0.25	70.55±0.55
SeqFT + MLPDC	70.56±1.09 ^{16.16}	82.70±0.72	67.87±0.51 ^{17.45}	79.68±0.57	79.21±1.44 ^{15.85}	86.98±0.86	69.88±0.31 ^{18.51}	76.71±0.56
α_1 -SLDC (ours)	70.42±1.01 ^{16.02}	82.86±0.85	61.01±0.74 ^{15.59}	76.33±0.57	79.84±1.15 ^{16.48}	88.15±0.75	71.81±0.39 ^{110.44}	77.73±0.43
α_2 -SLDC (ours)	78.98±0.95 ^{114.58}	86.70±0.72	77.53±0.05 ^{117.11}	84.25±0.52	81.75±0.74 ^{18.39}	88.75±0.79	71.38±0.40 ^{110.01}	77.79±0.39
SeqFT + MLPDC + ADE	76.66±1.22 ^{113.26}	85.74±0.91	74.24±0.47 ^{113.82}	82.90±0.42	79.65±0.93 ^{16.29}	86.94±0.99	70.54±0.72 ^{19.17}	77.04±0.40
α_1 -SLDC + ADE (ours)	78.03±1.36 ^{113.63}	86.54±0.80	76.26±0.59 ^{115.84}	83.87±0.40	81.57±0.98 ^{18.21}	88.78±0.75	72.29±0.42 ^{110.92}	77.95±0.31
α_2 -SLDC + ADE (ours)	79.43±0.77 ^{115.03}	86.92±0.88	77.51±0.21 ^{117.09}	84.32±0.44	83.15±0.81 ^{19.79}	89.26±0.82	72.47±0.08 ^{111.10}	77.95±0.28
SeqKD	76.97±0.20	86.00±0.66	73.87±0.66	82.37±0.68	80.35±0.41	88.09±0.92	66.93±0.28	75.07±0.45
SeqKD + MLPDC	72.56±0.81 ^{14.41}	83.44±0.86	71.18±0.37 ^{12.69}	81.07±0.51	82.59±0.95 ^{12.24}	88.80±1.01	72.11±0.22 ^{15.18}	77.44±0.43
β_1 -SLDC (ours)	80.55±0.53 ^{14.85}	87.29±0.76	77.79±0.27 ^{13.92}	84.19±0.43	85.50±0.55 ^{15.15}	90.52±0.97	73.00±0.13 ^{16.07}	78.08±0.25
β_2 -SLDC (ours)	81.82±0.52 ^{14.85}	87.60±0.71	80.10±0.31 ^{16.23}	85.07±0.54	85.16±0.29 ^{14.81}	90.30±0.96	73.01±0.11 ^{16.08}	77.96±0.28
SeqKD + MLPDC + ADE	80.54±0.49 ^{15.24}	87.26±0.80	78.77±0.28 ^{14.90}	84.53±0.42	82.42±0.81 ^{12.07}	88.70±0.97	71.11±0.25 ^{14.18}	77.06±0.29
β_1 -SLDC + ADE (ours)	82.21±0.53 ^{15.24}	87.85±0.68	80.59±0.29 ^{16.72}	85.31±0.37	86.02±0.31 ^{15.67}	90.62±0.94	73.42±0.11 ^{16.49}	78.03±0.33
β_2 -SLDC + ADE (ours)	82.32±0.57 ^{15.35}	87.78±0.76	80.61±0.31 ^{16.74}	85.32±0.42	86.12±0.23 ^{15.77}	90.52±0.98	73.14±0.22 ^{16.21}	77.96±0.28

Table 2: State-of-the-art CIL performance comparison across CUB-200, Cars-196, CIFAR-100, and ImageNet-R by a supervised pre-trained ViT-B/16 on ImageNet-21K.

Method	CUB-200		Cars-196		CIFAR-100		ImageNet-R	
	Last-Acc	Inc-Acc	Last-Acc	Inc-Acc	Last-Acc	Inc-Acc	Last-Acc	Inc-Acc
Joint-Training	88.43±0.25	-	83.79±0.25	-	93.56±0.17	-	82.74±0.14	-
BiC	81.91±2.50	89.29±1.57	63.10±5.71	73.75±2.37	88.45±0.57	93.37±0.32	64.89±0.80	73.66±1.61
LwF	69.75±1.37	80.45±2.08	49.94±3.24	63.28±1.11	87.99±0.05	92.13±1.16	67.29±1.67	74.47±1.48
RanPAC	85.82±0.53	91.47±0.96	53.84±0.84	64.39±1.18	90.09±0.25	93.31±0.98	72.62±0.11	78.35±0.58
SLCA	84.71±0.40	90.94±0.68	67.73±0.85	76.93±1.21	91.53±0.28	94.09±0.87	77.00±0.33	81.17±0.64
SLCA++	86.59±0.29	91.63±0.72	73.97±0.22	79.49±0.80	91.46±0.18	94.20±0.71	78.09±0.22	82.95±0.78
CoMA	85.95±0.29	90.75±0.39	78.55±0.50	78.55±0.42	92.00±0.13	94.12±0.63	77.47±0.05	83.32±0.17
CoFiMA	87.11±0.56	91.87±0.69	76.96±0.64	82.65±0.96	92.77±0.24	94.89±0.94	78.25±0.26	81.48±0.56
SeqFT	76.57±1.62	85.84±0.47	54.53±1.75	69.48±0.83	83.14±1.37	88.06±1.03	68.56±0.94	77.46±0.31
SeqFT + MLPDC	68.32±1.79 ^{18.25}	84.29±0.95	64.65±0.41 ^{110.12}	78.54±0.43	87.20±1.00 ^{14.06}	91.96±0.57	73.38±0.30 ^{14.82}	81.45±0.65
α_1 -SLDC (ours)	71.49±2.54 ^{15.08}	84.65±1.01	46.78±1.80 ^{17.75}	68.64±1.34	87.45±1.09 ^{14.31}	92.41±0.50	76.85±0.20 ^{18.29}	82.85±0.57
α_2 -SLDC (ours)	78.65±2.18 ^{12.08}	88.72±1.01	74.07±0.78 ^{119.54}	83.32±0.55	88.69±0.44 ^{15.55}	93.02±0.56	77.05±0.04 ^{18.49}	82.96±0.46
SeqFT + MLPDC + ADE	75.44±1.71 ^{11.13}	86.69±1.09	69.26±0.47 ^{114.73}	81.15±0.32	88.44±0.48 ^{15.30}	92.21±0.82	76.98±0.15 ^{18.42}	82.73±0.44
α_1 -SLDC + ADE (ours)	77.02±2.34 ^{10.45}	87.93±0.97	73.01±0.97 ^{118.48}	82.82±0.52	88.73±0.86 ^{15.59}	92.92±0.45	78.14±0.10 ^{19.58}	83.38±0.45
α_2 -SLDC + ADE (ours)	77.56±2.00 ^{10.99}	88.20±0.94	73.04±0.57 ^{118.51}	83.02±0.44	89.83±0.53 ^{16.69}	93.43±0.61	78.82±0.26 ^{110.26}	83.61±0.36
SeqKD	86.75±0.29	92.22±0.55	75.62±0.32	83.36±0.63	88.03±0.62	92.85±0.91	74.04±0.38	81.25±0.32
SeqKD + MLPDC	75.76±1.23 ^{10.99}	87.22±0.60	70.19±0.65 ^{15.43}	80.69±0.86	89.65±0.56 ^{11.62}	93.26±0.75	78.57±0.17 ^{14.53}	83.27±0.73
β_1 -SLDC (ours)	83.76±1.41 ^{12.99}	91.06±0.72	73.71±1.03 ^{11.91}	82.48±0.71	91.21±0.45 ^{13.18}	94.27±0.69	79.56±0.44 ^{15.52}	83.82±0.55
β_2 -SLDC (ours)	85.85±0.49 ^{10.90}	91.92±0.60	79.91±0.47 ^{14.29}	85.11±0.49	90.98±0.27 ^{12.95}	94.20±0.72	79.54±0.02 ^{15.50}	83.96±0.48
SeqKD + MLPDC + ADE	85.05±0.80 ^{11.70}	91.43±0.76	77.30±0.56 ^{11.68}	84.03±0.56	89.48±0.58 ^{11.45}	93.17±0.70	78.31±0.17 ^{14.27}	83.15±0.58
β_1 -SLDC + ADE (ours)	87.18±0.50 ^{10.43}	92.42±0.59	80.61±0.36 ^{14.99}	85.51±0.41	91.36±0.34 ^{13.33}	94.37±0.73	79.78±0.24 ^{15.74}	83.91±0.40
β_2 -SLDC + ADE (ours)	87.15±0.50 ^{10.40}	92.38±0.57	80.50±0.30 ^{14.88}	85.45±0.41	91.48±0.24 ^{13.45}	94.38±0.69	80.00±0.29 ^{15.96}	84.01±0.46

CIFAR-100 and ImageNet-R serve as standard CIL benchmarks, while CUB-200 and Cars-196 evaluate fine-grained classification capabilities. All experiments are conducted using the PILOT framework (Sun et al. 2025) with consistent random seeds to ensure fair comparison.

Metrics. We report two key metrics, i.e., the average classification accuracy across all classes encountered after each incremental task, denoted as Inc-Acc (%), and the classification accuracy after completing the final task, denoted as Last-Acc (%). The first metric evaluates the balance of remembering old classes and learning new ones throughout the CIL process, while the second one shows the overall performance across all classes after all tasks are learned.

CIL baselines. Our proposed SLDC methods are compared against advanced PTM-based CIL approaches, including BiC (Wu et al. 2019), LwF (Li and Hoiem 2017), SLCA/SLCA++ (Gao et al. 2023; Zhang et al. 2024), RanPAC (McDonnell et al. 2023), and CoMA/CoFiMA (Marouf et al. 2024). Specifically, SeqKD denotes the distillation-

enhanced SeqFT. Since $\alpha_{1,2}$ -SLDC and $\beta_{1,2}$ -SLDC methods are implemented based on SeqFT and SeqKD, respectively, the relative improvements over SeqFT and SeqKD are reported. Notably, our methods can be further integrated with other techniques, such as CoMA and CoFiMA, where EMA is employed on model parameters to mitigate representation drifts. As an upper-bound reference, the performance of joint training is reported, where the model is trained on all incremental tasks simultaneously. Additionally, MLPDC, which refers to the MLP-based distribution compensation method, also serves as a baseline method to SLDC-based compensation.

Implementation details. Two PTMs, which are the ViT-B/16 pre-trained on ImageNet-21K supervisedly (Ridnik et al. 2021) and the ViT-B/16 pre-trained using the MoCo-V3 self-supervised technique on ImageNet-1K (Chen, Xie, and He 2021), are employed. The LoRA adapters are of rank 4 and optimized using the Adam optimizer with a learning rate of 10^{-4} and a weight decay of 3×10^{-5} . For α_1 -SLDC,

λ_{α_1} is set to 10^{-4} . In the case of α_2 -SLDC, $\mathbf{A}_t$ and $\psi(f)$ are initialized as an identity matrix and a three-layer MLP with ReLU activation, respectively, where the hidden dimension of $\psi(f)$ matches that of the [cls] token in the ViTs. The default value for λ_{α_2} is 0.5, and the coefficients (c_1, c_2) are set to (0.9, 0.1). Additional training details are provided in the Appendix. To re-estimate the class-specific mean and covariance in α_2 -SLDC through Gaussian sampling, we use $N = 10d$ samples per class, where d denotes the feature dimension. For feature-based distillation in $\beta_{1,2}$ -SLDC, let $\gamma_{\text{KD}} = 1.0$ and $\gamma_{\text{Norm}} = 0.1$ simply.

Main comparison results

Tables 1 and 2 present comprehensive comparisons between our proposed SLDC methods and state-of-the-art CIL approaches using both self-supervised (MoCo-V3) and supervised (ImageNet-21K) pre-trained ViT-B/16 backbones. In addition, the comparison results are visualized in Figs. 7 and 8 in the appendix. Some notable observations are as follows.

1. Vanilla SeqFT struggles with severe forgetting, as evidenced by its low Last-Acc values, such as 64.40% on CUB-200 and 61.37% on ImageNet-R (see Table 1). In contrast, SLDC methods significantly boost accuracy without regularizing the backbone optimization. For example, α_2 -SLDC lifts CUB-200 performance to 78.98% (a +14.58% absolute gain) with MoCo-V3 architecture.
2. When ADE is not employed, α_2 -SLDC consistently outperforms linear α_1 -SLDC and nonlinear MLPDC on fine-grained datasets, with notable gains on Cars-196 (77.53% vs. 61.01% with MoCo-V3) and CUB-200 (78.98% vs. 70.42%).
3. SeqKD improves SeqFT substantially, with a +12.57% Last-Acc gain on CUB-200 using Sup-21K. Notably, distillation pairs exceptionally well with SLDC: β_1 -SLDC (distillation-enhanced α_1 -SLDC) nearly matches α_2 -SLDC, such as 80.55% vs. 78.98% on CUB-200 with MoCo-V3.
4. α_2 -SLDC and β_2 -SLDC deliver robust performance across all datasets and pre-trained models. It outperforms MLPDC (nonlinear compensation) by +6.52% on Cars-196 and +2.17% on CIFAR-100 with Sup-21K pretraining, supporting our hypothesis that appropriate operators lie between linear and nonlinear extremes.
5. ADE significantly enhances the performance of SLDC methods on fine-grained datasets. For example, α_1 -SLDC shows instability with Sup-21K pretraining, with Last-Acc dropping on CUB-200 (71.49% vs. SeqFTs 76.57%) and Cars-196 (46.78% vs. 54.53%). Nonetheless, α_1 -SLDC + ADE achieves a striking +26.23% improvement on Cars-196 compared to its non-ADE counterpart. This confirms ADEs ability to mitigate approximation errors when task data is limited.

Ablation studies

Effectiveness to long-sequence CIL. Here, we extend the evaluation to 20 tasks to assess the effectiveness of SLDC methods on long-sequence CIL scenarios. The comparative

results with and without distillation on the MoCo-V3 architecture are presented in Figure 2, while the corresponding results for the Sup-21K architecture are provided in Figure 9 in the Appendix. Some noteworthy observations are listed as follows. 1) The α_2 -SLDC approach consistently outperforms α_1 -SLDC when neither distillation nor ADE is applied. The incorporation of both distillation and ADE techniques yields significant improvements across all SLDC variants. 2) MLPDC exhibits particularly poor performance on the Cars196 and CUB200 datasets. 3) The α_1 -SLDC still suffers from instability when it is implemented on the Sup-21K architecture, and it can be mitigated effectively through either distillation or ADE techniques.

Effectiveness to hybrid CIL datasets. To evaluate the robustness of SLDC methods in heterogeneous CIL scenarios, we construct a hybrid CIL benchmark where each evaluation dataset (CIFAR-100, Cars-196, CUB-200, and ImageNet-R) is treated as a distinct incremental task. Figure 3 presents the comparative results under both MoCo-V3 and Sup-21K pre-training strategies with and without distillation. Key findings include: 1) SLDC methods outperform both SeqFT and MLPDC baselines across all settings. 2) The performance gap between $\alpha_{1,2}$ -SLDC methods narrows significantly in this setting. It means that α_1 -SLDC achieves comparable stability to its weak-nonlinear counterpart when dealing with larger task-specific datasets. In practice, we have tried experiments with varied dataset orders, and the evaluation results are similar.

Influences of α_{temp} in α_1 -SLDC. This part analyzes the impact of the temperature parameter α_{temp} in α_1 -SLDC. Focusing on the MoCo-V3 architecture with distillation, we evaluate four α_{temp} values ([0.5, 1.0, 2.0, 5.0]), with results shown in Figure 4. Our experiments reveal two key findings. 1) When ADE is not employed, $\alpha_{\text{temp}} = 1.0$ achieves optimal performance on fine-grained datasets Cars-196 and CUB-200. 2) When ADE is employed, reducing α_{temp} below 1.0 becomes advantageous for effectively utilizing the unlabeled dataset. These findings suggest that the optimal temperature depends on whether ADE is implemented.

Influences of γ_{α_2} . Here, we investigate the influences of regularization coefficient γ_{α_2} in α_2 -SLDC by selecting values from [0.1, 0.5, 1.0, 2.0]. For simplicity, the results on the MoCo-V3 architecture with distillation are reported in Figure 5. The performance of α_2 -SLDC exhibits remarkable stability across the tested range of γ_{α_2} values. It suggests that the prior assumption governing the hypothesis space of the transition operator plays a more critical role than the specific choice of the regularization coefficient.

Sensitivity to sample selection in ADE. This section examines the impact of sample selection in the ADE process. We evaluate three ADE datasets (CIFAR-10, SVHN, and ImageNet) with varying sample sizes ranging from 512 to 2048. As shown in Figure 6, our analysis reveals distinct patterns across different benchmark datasets. For the fine-grained CUB-200 dataset, all ADE variants improve SLDC performance, with larger ADE sample sizes yielding progressively better results. In contrast, the ImageNet-R dataset

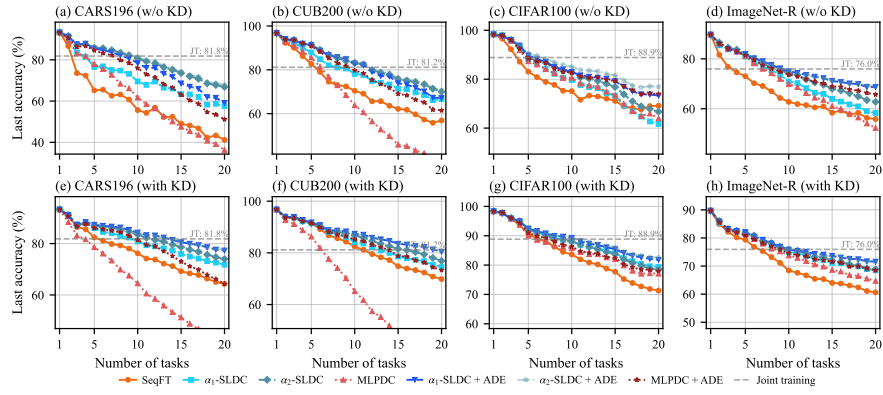


Figure 2: Performance comparison of SLDC methods on a 20-task sequence, demonstrating state-of-the-art results both with and without knowledge distillation.

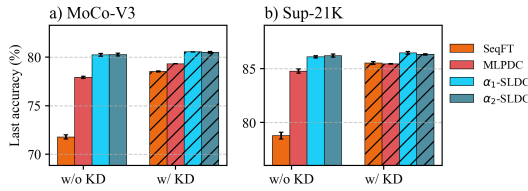


Figure 3: Comparative performance of SLDC methods on hybrid CIL tasks comprising four distinct datasets: CIFAR-100, Cars-196, CUB-200, and ImageNet-R

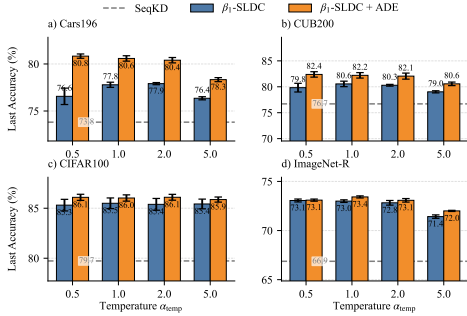


Figure 4: Performance comparison with varying temperature parameters α_{temp} in α_1 -SLDC

demonstrates stable performance without requiring ADE, suggesting that the training samples in ImageNet-R is sufficient to achieve robust performance.

Conclusions

In this paper, an in-depth exploration on pre-trained ViT-based CIL is conducted, and it is highlighted that effective approximation of the latent space transition operator is critical for mitigating the adverse effects of distribution drifts during sequential optimization. Accordingly, the linear α_1 -SLDC and weak-nonlinear α_2 -SLDC methods are introduced, along with their distillation-enhanced variants, β_1 -SLDC and β_2 -SLDC, to align the distributions of previous classes with the updated feature space. Extensive experi-

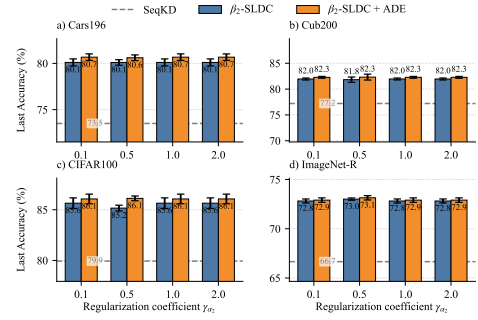


Figure 5: Performance evaluation of α_2 -SLDC with varying regularization coefficients $\gamma_{\alpha_2} \in \{0.1, 0.5, 1.0, 2.0\}$

ments demonstrate the efficacy of our methods. Notably, the synergy of distillation (to limit excessive optimization) and SLDC (to compensate for distribution drifts) significantly narrows the performance gap between CIL and joint learning, making CIL more practical for real-world applications.

However, we observed that α_1 -SLDC exhibits instability on certain fine-grained datasets with the Sup-21K architecture, and auxiliary unlabeled data are required to stabilize its performance. In addition, the applicability of SLDC methods to multi-modal models remains an open question, which we plan to explore in our future works.

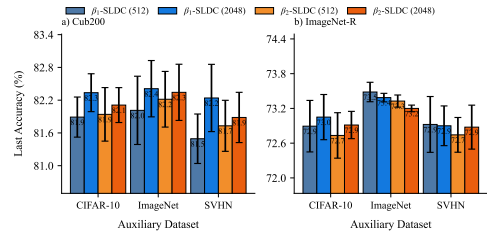


Figure 6: Performance comparison of SLDC methods with varying ADE datasets (CIFAR-10, SVHN, and ImageNet) and sample sizes (512 to 2048)

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 62573062, in part by the Shenzhen Science and Technology Program under grant JCYJ20230807093513027, and in part by the Fundamental Research Funds for the Central Universities under grant 1243300008.

References

- Ahrens, K.; Lehmann, H. H.; Lee, J. H.; and Wermter, S. 2024. Read Between the Layers: Leveraging Multi-Layer Representations for Rehearsal-Free Continual Learning with Pre-Trained Models. *Transactions on Machine Learning Research*.
- Chen, X.; Xie, S.; and He, K. 2021. An Empirical Study of Training Self-Supervised Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 9640–9649.
- Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations*.
- Gao, Q.; Zhao, C.; Sun, Y.; Xi, T.; Zhang, G.; Ghanem, B.; and Zhang, J. 2023. A Unified Continual Learning Framework with General Parameter-Efficient Tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 11483–11493.
- Gomez-Villa, A.; Goswami, D.; Wang, K.; Bagdanov, A. D.; Twardowski, B.; and van de Weijer, J. 2024. Exemplar-free continual representation learning via learnable drift compensation. In *European Conference on Computer Vision*, 473–490. Springer.
- He, R.; Fang, D.; Xu, Y.; Cui, Y.; Li, M.; Chen, C.; Zeng, Z.; and Zhuang, H. 2025. Semantic Shift Estimation via Dual-Projection and Classifier Reconstruction for Exemplar-Free Class-Incremental Learning. In *Proceedings of the International Conference on Machine Learning*, 1–24.
- Hendrycks, D.; Basart, S.; Mu, N.; Kadavath, S.; Wang, F.; Dorundo, E.; Desai, R.; Zhu, T.; Parajuli, S.; Guo, M.; et al. 2021. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 8340–8349.
- Hu, E. J.; yelong shen; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *Proceedings of the International Conference on Learning Representations*.
- Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In *Advances in Neural Information Processing Systems*, 8580–8589.
- Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 554–561.
- Krizhevsky, A.; and Hinton, G. 2009. Learning Multiple Layers of Features from Tiny Images. *Technical Report*, 1–60.
- Li, Z.; and Hoiem, D. 2017. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12): 2935–2947.
- Li, Z.; Zhao, L.; Zhang, Z.; Zhang, H.; Liu, D.; Liu, T.; and Metaxas, D. N. 2024. Steering Prototypes With Prompt-Tuning for Rehearsal-Free Continual Learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2523–2533.
- Lu, Y.; Zhang, S.; Cheng, D.; Xing, Y.; Wang, N.; Wang, P.; and Zhang, Y. 2024. Visual Prompt Tuning in Null Space for Continual Learning. In *Advances in Neural Information Processing Systems*, volume 37, 7878–7901.
- Marouf, I. E.; Roy, S.; Tartaglione, E.; and Lathuilière, S. 2024. Weighted Ensemble Models Are Strong Continual Learners. In *Proceedings of the European Conference on Computer Vision*, 306–324.
- McDonnell, M. D.; Gong, D.; Parvaneh, A.; Abbasnejad, E.; and van den Hengel, A. 2023. RanPAC: Random Projections and Pre-trained Models for Continual Learning. In *Advances in Neural Information Processing Systems*, volume 36, 12022–12053.
- Panos, A.; Kobe, Y.; Reino, D. O.; Aljundi, R.; and Turner, R. E. 2023. First Session Adaptation: A Strong Replay-Free Baseline for Class-Incremental Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 18820–18830.
- Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, 8748–8763.
- Ridnik, T.; Ben-Baruch, E.; Noy, A.; and Zelnik-Manor, L. 2021. ImageNet-21K Pretraining for the Masses. *arXiv preprint arXiv:2104.10972*.
- Rypeść, G.; Cygert, S.; Trzciński, T.; and Twardowski, B. o. 2024. Task-recency bias strikes back: Adapting covariances in Exemplar-Free Class Incremental Learning. In *Advances in Neural Information Processing Systems*, volume 37, 63268–63289.
- Smith, J. S.; Karlinsky, L.; Gutta, V.; Cascante-Bonilla, P.; Kim, D.; Arbelle, A.; Panda, R.; Feris, R.; and Kira, Z. 2023. CODA-Prompt: COntinual Decomposed Attention-Based Prompting for Rehearsal-Free Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11909–11919.
- Sun, H.-L.; Zhou, D.-W.; Zhan, D.-C.; and Ye, H.-J. 2025. PILOT: A Pre-Trained Model-Based Continual Learning Toolbox. *SCIENCE CHINA Information Sciences*, 68(4): 147101.
- Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset.

- Wang, L.; Xie, J.; Zhang, X.; Su, H.; and Zhu, J. 2025. HiDe-PET: Continual Learning via Hierarchical Decomposition of Parameter-Efficient Tuning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(8): 6687–6702.
- Wang, Z.; Zhang, Z.; Ebrahimi, S.; Sun, R.; Zhang, H.; Lee, C.-Y.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022a. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *Proceedings of European Conference on Computer Vision (ECCV)*, 631–648. Springer.
- Wang, Z.; Zhang, Z.; Lee, C.-Y.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022b. Learning To Prompt for Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 139–149.
- Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; and Fu, Y. 2019. Large Scale Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 374–382.
- Xiao, J.-W.; Zhang, C.-B.; Feng, J.; Liu, X.; van de Weijer, J.; and Cheng, M.-M. 2023. Endpoints Weight Fusion for Class Incremental Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7204–7213.
- Yu, L.; Twardowski, B.; Liu, X.; Herranz, L.; Wang, K.; Cheng, Y.; Jui, S.; and Weijer, J. v. d. 2020. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6982–6991.
- Zhang, G.; Wang, L.; Kang, G.; Chen, L.; and Wei, Y. 2023. 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 (ICCV)*, 19148–19158.
- Zhang, G.; Wang, L.; Kang, G.; Chen, L.; and Wei, Y. 2024. SLCA++: Unleash the Power of Sequential Fine-tuning for Continual Learning with Pre-training. *arXiv preprint arXiv:2408.08295*.
- Zhao, L.; Zhang, X.; Yan, K.; Ding, S.; and Huang, W. 2024. SAFE: Slow and Fast Parameter-Efficient Tuning for Continual Learning with Pre-Trained Models. In *Advances in Neural Information Processing Systems*, volume 37, 113772–113796.
- Zheng, Z.; Ma, M.; Wang, K.; Qin, Z.; Yue, X.; and You, Y. 2023. Preventing Zero-Shot Transfer Degradation in Continual Learning of Vision-Language Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 19125–19136.
- Zhou, D.-W.; Zhang, Y.; Wang, Y.; Ning, J.; Ye, H.-J.; Zhan, D.-C.; and Liu, Z. 2025. Learning Without Forgetting for Vision-Language Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(6): 4489–4504.
- Zhuang, H.; He, R.; Tong, K.; Zeng, Z.; Chen, C.; and Lin, Z. 2024. DS-AL: A dual-stream analytic learning for exemplar-free class-incremental learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 17237–17244.

Appendix

Algorithm The pseudocode of SLDC methods is summarized in Algorithm 1.

Implementation details

LoRA settings. For a pre-trained weight matrix $W \in \mathbb{R}^{d_2 \times d_1}$, LoRA introduces a low-rank updating $\Delta W = BA$, where $A \in \mathbb{R}^{k \times d_1}$ and $B \in \mathbb{R}^{d_2 \times k}$, with $\text{rank } k \ll \min(d_1, d_2)$. During training, only the low-rank matrices A and B are updated, while the original weight matrix W remains frozen.

Following the settings from SLCA++ (Zhang et al. 2024), we leverage singular value decomposition (SVD) to enhance the initialization of LoRA adapters. Specifically, for a pre-trained weight matrix W , SVD decomposes it as $W = U\Sigma_s V^\top$, where $U \in \mathbb{R}^{d_2 \times d_2}$ and $V \in \mathbb{R}^{d_1 \times d_1}$ are the left and right singular vectors, respectively, and $\Sigma_s \in \mathbb{R}^{d_2 \times d_1}$ contains the singular values. Then, A is initialized by the top- k rows of $V^\top$, while B is initialized by zero values. By SVD-based initialization, it can be ensured that $BA = 0$ and the learning subspace of A is aligned with the principal directions of W at the start of training.

In all experiments, we let $k = 4$ and apply LoRA adapters to tune both the attention and MLP blocks. The parameter comparison between the full fine-tuning and the LoRA-based tuning are summarized in Table 3.

Table 3: Comparison of trainable parameters between the full fine-tuning and LoRA-based tuning

Parameter Type	Count	Percentage of Total
Total parameters	86,314,752	100.00%
LoRA parameters	516,096	0.60%

Backbone optimization. The training process employs 15 epochs for the CUB-200 and Cars-196 datasets, 5 epochs for CIFAR-100, and 10 epochs for ImageNet-R. All other hyperparameters remain consistent across the four datasets. For ViT backbones, an Adam optimizer is employed with an initial learning rate of 10^{-4} , which is reduced to $\frac{1}{3} \times 10^{-4}$ in the final epoch. The linear classifier uses a learning rate which is 10 times higher than that of the ViT backbones.

Optimization of weak-nonlinear and MLP transformations in α_2 -SLDC and MLPDC. To optimize the weak-nonlinear transformation $\psi(f)$ in α_2 -SLDC and the MLP in MLPDC, the following configurations are employed.

1. Both the weak-nonlinear and MLP transformations are optimized for 5,000 steps totally. The batch size is 32.
2. The Adam optimizer is employed with an initial learning rate of 10^{-3} , which is reduced to 5×10^{-4} at the last optimization step.
3. The models take normalized pre-optimization features $\tilde{F}_{y_t}^{t-1}$ as inputs and are trained to minimize the mean squared error (MSE) loss between the predictive values and the normalized post-optimization features $\tilde{F}_{y_t}^t$.

4. For the weak-nonlinear transformation $\psi(f)$, no weight decay is employed. In contrast, for MLP transformation, a weight decay of 10^{-6} is applied to mitigate overfitting.

Introduction to benchmark datasets

Below is a detailed introduction to the four benchmark datasets used for evaluating CIL performance. These datasets are widely adopted in the machine learning community for their diversity and ability to test various aspects of model performance, particularly in CIL and fine-grained classification tasks.

CIFAR-100 CIFAR-100 is a standard benchmark for image classification and CIL tasks. It comprises 100 classes of natural images, which are grouped into 20 superclasses (e.g., vehicles, animals, household items). The dataset contains 60,000 color images of size 32×32 pixels, with 500 training samples and 100 test samples per class, resulting in 50,000 training and 10,000 test images. In particular, we resize the resolution to 224×224 in our experiments.

ImageNet-R ImageNet-R (R for Renditions) includes 200 classes, featuring hard examples from ImageNet-21K and new images in diverse styles, such as cartoons, paintings, and sketches. The dataset consists of 30,000 images in total, with 24,000 training and 6,000 test images. The diversity in visual styles and the inclusion of difficult examples make ImageNet-R a rigorous benchmark for assessing generalization in incremental learning scenarios.

CUB-200 The CUB-200 dataset is tailored for fine-grained classification. It includes 200 distinct bird species. The dataset contains approximately 11,788 high-resolution images. Each class has nearly 60 images on average. The dataset is evenly divided into training and test sets. Each set includes roughly 30 images per class.

Cars-196 The Cars-196 dataset serves as another fine-grained classification benchmark. It comprises 196 distinct car types. The dataset includes 16,184 high-resolution images, which are split into 8,144 training images and 8,040 test images. It captures subtle differences in car designs, such as headlights, grilles, or body shapes across various angles and lighting conditions.

Introduction to implemented pre-trained ViT

This paper employs two pre-trained models, i.e., the MoCo-V3 ViT-B/16 and Sup21K ViT-B/16. Detailed descriptions of these models are provided below.

MoCo-V3 architecture. MoCo-V3 is a self-supervised learning framework which extends the momentum contrast (MoCo) approach to ViTs. The MoCo approach employs a query encoder and a momentum-updated key encoder, along with a contrastive loss, to learn robust visual representations without labeled data. In this study, the ViT-B/16 backbone pre-trained with MoCo-V3 on ImageNet-1K is employed.

Sup-21K architecture. The Sup-21K model refers to the ViT-B/16 architecture pre-trained in a fully supervised manner on the large-scale ImageNet-21K dataset. This model

Algorithm 1: Sequential Learning with Drift Compensation (SLDC) for Pre-trained ViT-based CIL

Require: Pre-trained ViT $\mathcal{F}_\theta$, Initial classifier $\mathcal{C}_\phi$, Training datasets $\{\mathcal{D}_t\}_{t=1}^T$ Hyperparameters: $\gamma_{\text{KD}}, \gamma_{\text{Norm}}, \gamma_{\alpha_1}, \gamma_{\alpha_2}$

- 1: Initialize Gaussian distribution set: $\mathcal{H}_0 \leftarrow \emptyset$. Initialize the cumulative set of observed classes: $\mathcal{C}_{t-1} = \emptyset$.
- 2: **for** each task $t = 1$ **to** T **do**
- 3: **// Phase 1: Sequential Model Adaptation**
- 4: Expand classifier $\mathcal{C}_\phi \leftarrow \text{Linear}(\mathbb{R}^d, |\mathcal{C}_{t-1} \cup \mathcal{Y}_t|)$
- 5: Update observed classes $\mathcal{C}_t \leftarrow \mathcal{C}_{t-1} \cup \mathcal{Y}_t$
- 6: Save backbone checkpoint $\theta_{t-1} \leftarrow \theta$
- 7: **for** each epoch = 1 **to** N_{epochs} **do**
- 8: **for** each batch $(x, y) \in \mathcal{D}_t$ **do**
- 9: Extract features: $f \leftarrow \mathcal{F}_\theta(x)$
- 10: Compute classification loss: $\mathcal{L}_{\text{CE}} \leftarrow \text{CE}(\mathcal{C}_\phi(f), y)$
- 11: $\mathcal{L} \leftarrow \mathcal{L}_{\text{CE}}$
- 12: **if** using distillation **then**
- 13: $f_{\text{prev}} \leftarrow \mathcal{F}_{\theta_{t-1}}(x)$, $\mathcal{L}_{\text{KD}} \leftarrow \gamma_{\text{KD}} \cdot \|f_{\text{prev}} - f\|_2^2$, $\mathcal{L}_{\text{Norm}} \leftarrow \gamma_{\text{Norm}} \cdot (\|f_{\text{prev}}\|_2 - \|f\|_2)^2$
- 14: $\mathcal{L} \leftarrow \mathcal{L} + \mathcal{L}_{\text{KD}} + \mathcal{L}_{\text{Norm}}$
- 15: **end if**
- 16: Update θ, ϕ via $\nabla_{\theta, \phi} \mathcal{L}$
- 17: **end for**
- 18: **end for**
- 19: **// Phase 2: Distribution Compensation**
- 20: Compute $F_{\mathcal{Y}_t}^{t-1} = [\mathcal{F}_{\theta_{t-1}}(x_{t,1}), \dots, \mathcal{F}_{\theta_{t-1}}(x_{t,n_t})] \in \mathbb{R}^{d \times n_t}$ and $F_{\mathcal{Y}_t}^t = [\mathcal{F}_{\theta_t}(x_{t,1}), \dots, \mathcal{F}_{\theta_t}(x_{t,n_t})] \in \mathbb{R}^{d \times n_t}$
- 21: Calculate $\tilde{F}^{t-1}$ and $\tilde{F}^t$ by normalizing the elements of $F_{\mathcal{Y}_t}^{t-1}$ and $F_{\mathcal{Y}_t}^t$ to unit vectors.
- 22: **if** use α_1 -SLDC **then**
- 23: Compute the linear transformation matrix: $\mathbf{A}_t \leftarrow \tilde{F}^t (\tilde{F}^{t-1})^\top (\tilde{F}^{t-1} (\tilde{F}^{t-1})^\top + \gamma_{\alpha_1} I)^{-1}$
- 24: Apply sample complexity-based re-weighting: $w \leftarrow \exp(-|\mathcal{D}_t|/d)$; $\mathbf{A}_t \leftarrow (1 - w)\mathbf{A}_t + wI$
- 25: **for** each $(\mu_c, \Sigma_c) \in \mathcal{H}_{t-1}$ **do**
- 26: $\mu_c \leftarrow \mathbf{A}_t \mu_c$
- 27: $\Sigma_c \leftarrow \mathbf{A}_t \Sigma_c \mathbf{A}_t^\top$
- 28: **end for**
- 29: **else if** use α_2 -SLDC **then**
- 30: Initialize $\mathbf{A} \leftarrow I_d, \psi \leftarrow \text{MLP}(d \rightarrow h \rightarrow d), (c_1, c_2) \leftarrow (0.9, 0.1)$
- 31: $\min_{\mathbf{A}, \psi, \alpha_1/2} \|c_1 \mathbf{A} \tilde{F}^{t-1} + c_2 \psi(\tilde{F}^{t-1}) - \tilde{F}^t\|_F^2 + \gamma_{\alpha_2} (c_1 - 1)^2$
- 32: Monte Carlo transformation:
- 33: **for** each $(\mu_c, \Sigma_c) \in \mathcal{H}_{t-1}$ **do**
- 34: Sample $\{f_c^{(i)}\}_{i=1}^N \sim \mathcal{N}(\mu_c, \Sigma_c)$
- 35: Transform $\tilde{f}_c^{(i)} \leftarrow \alpha_1 \mathbf{A} f_c^{(i)} + \alpha_2 \psi(f_c^{(i)})$
- 36: Re-estimate $\mu_c \leftarrow \text{mean}(\{\tilde{f}_c^{(i)}\})$, $\Sigma_c \leftarrow \text{cov}(\{\tilde{f}_c^{(i)}\})$
- 37: **end for**
- 38: **end if**
- 39: **// Phase 3: Classifier Refinement**
- 40: Update distribution set: $\mathcal{H}_t \leftarrow \mathcal{H}_{t-1} \cup \{\text{New Gaussians for } \mathcal{Y}_t\}$
- 41: Generate synthetic features: $\mathcal{F}_{\text{synth}} \leftarrow \bigcup_c \{f \sim \mathcal{N}(\mu_c, \Sigma_c)\}_{c \in \mathcal{C}_t}$
- 42: Refine classifier: $\phi \leftarrow \arg \min_{\phi} \mathbb{E}_{(f, c) \sim \mathcal{F}_{\text{synth}}} [-\log p_\phi(c|f)]$
- 43: **end for**

Table 4: Dataset descriptions

Dataset	Classes	Train Images	Test Images	Resolution
ImageNet-R	200	24,000	6,000	224×224
CUB200	200	5,994	5,794	224×224
CIFAR100	100	50,000	10,000	224×224
Cars196	196	8,054	8,131	224×224

benefits from rich semantic supervision across 21K categories, providing robust and transferable feature representations for downstream tasks.

More analytical results

Statement 1 (Solution to the regularized least-squares problem). *The regularized least-squares problem for estimating the linear operator $\mathbf{A}_t$ is formulated as*

$$\mathbf{A}_t = \arg \min_{\mathbf{A}} \left\| \mathbf{A} \tilde{F}_{\mathcal{Y}_t}^{t-1} - \tilde{F}_{\mathcal{Y}_t}^t \right\|_F^2 + \lambda \|\mathbf{A}\|_F^2,$$

where $\tilde{F}_{\mathcal{Y}_t}^{t-1} \in \mathbb{R}^{d \times n_t}$ and $\tilde{F}_{\mathcal{Y}_t}^t \in \mathbb{R}^{d \times n_t}$ are column-wise L_2 -normalized feature matrices, $\lambda = \gamma_{\alpha_1} > 0$ is the regularization coefficient, and $\|\cdot\|_F$ denotes the Frobenius norm. The analytical solution to this problem is given by

$$\mathbf{A}_t = \tilde{F}_{\mathcal{Y}_t}^t (\tilde{F}_{\mathcal{Y}_t}^{t-1})^\top \left(\tilde{F}_{\mathcal{Y}_t}^{t-1} (\tilde{F}_{\mathcal{Y}_t}^{t-1})^\top + \lambda I_d \right)^{-1},$$

where I_d is the $d \times d$ identity matrix.

Proof. Let $X = \tilde{F}_{\mathcal{Y}_t}^{t-1}$ and $Y = \tilde{F}_{\mathcal{Y}_t}^t$. The optimization objective is

$$\min_{\mathbf{A}} J(\mathbf{A}) = \|\mathbf{A}X - Y\|_F^2 + \lambda \|\mathbf{A}\|_F^2. \quad (17)$$

By expressing the Frobenius norms as traces, we obtain

$$\begin{aligned} \|\mathbf{A}X - Y\|_F^2 &= \text{tr}(X^\top \mathbf{A}^\top \mathbf{A} X) - 2 \text{tr}(X^\top \mathbf{A}^\top Y) \\ &\quad + \text{tr}(Y^\top Y), \end{aligned} \quad (18)$$

and

$$\lambda \|\mathbf{A}\|_F^2 = \lambda \text{tr}(\mathbf{A}^\top \mathbf{A}). \quad (19)$$

By combining (18) with (19), we obtain

$$\begin{aligned} J(\mathbf{A}) &= \text{tr}(X^\top \mathbf{A}^\top \mathbf{A} X) - 2 \text{tr}(X^\top \mathbf{A}^\top Y) \\ &\quad + \text{tr}(Y^\top Y) + \lambda \text{tr}(\mathbf{A}^\top \mathbf{A}). \end{aligned} \quad (20)$$

Taking the partial derivative of $J(\mathbf{A})$ with respect to $\mathbf{A}$ and setting it to zero, we have

$$\frac{\partial J}{\partial \mathbf{A}} = 2\mathbf{A}XX^\top - 2YX^\top + 2\lambda\mathbf{A} = 0, \quad (21)$$

which can be simplified to

$$\mathbf{A}(XX^\top + \lambda I_d) = YX^\top. \quad (22)$$

Since $XX^\top + \lambda I_d$ is invertible for $\lambda > 0$, the solution is obtained by

$$\mathbf{A} = YX^\top (XX^\top + \lambda I_d)^{-1}. \quad (23)$$

By substituting X and Y into the (23), we obtain the closed-form solution to $\mathbf{A}_t$. $\square$

Statement 2 (Linear transformation of a Gaussian distribution). *Let $\mathbf{x} \in \mathbb{R}^d$ be a random vector following a Gaussian distribution, $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. For any invertible linear transformation $\mathbf{y} = \mathbf{A}\mathbf{x}$, where $\mathbf{A} \in \mathbb{R}^{d \times d}$ is an invertible matrix, the random vector $\mathbf{y}$ follows a Gaussian distribution, $\mathbf{y} \sim \mathcal{N}(\mathbf{A}\boldsymbol{\mu}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top)$.*

Proof. The mean of $\mathbf{y}$ is given by

$$\mathbb{E}[\mathbf{y}] = \mathbb{E}[\mathbf{A}\mathbf{x}] = \mathbf{A}\mathbb{E}[\mathbf{x}] = \mathbf{A}\boldsymbol{\mu}. \quad (24)$$

The covariance of $\mathbf{y}$ is computed by

$$\begin{aligned} \text{Cov}(\mathbf{y}) &= \mathbb{E}[(\mathbf{y} - \mathbb{E}[\mathbf{y}])(\mathbf{y} - \mathbb{E}[\mathbf{y}])^\top] \\ &= \mathbb{E}[(\mathbf{A}\mathbf{x} - \mathbf{A}\boldsymbol{\mu})(\mathbf{A}\mathbf{x} - \mathbf{A}\boldsymbol{\mu})^\top] \\ &= \mathbb{E}[\mathbf{A}(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top \mathbf{A}^\top] \\ &= \mathbf{A}\mathbb{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top] \mathbf{A}^\top \\ &= \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top. \end{aligned} \quad (25)$$

Since $\mathbf{x}$ is Gaussian and $\mathbf{A}$ is invertible, $\mathbf{y} = \mathbf{A}\mathbf{x}$ is also Gaussian (as linear transformations preserve Gaussianity). Thus, $\mathbf{y} \sim \mathcal{N}(\mathbf{A}\boldsymbol{\mu}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top)$. $\square$

Statement 3 (The affinity of transition operator under the NTK limits). *Let $f_{\theta_0} : \mathcal{X} \rightarrow \mathbb{R}^d$ be a pre-trained neural network with pre-trained parameters θ_0 . During the fine-tuning, it is trained on a dataset $\{(x_i, y_i)\}_{i=1}^n$ where $y_i \in \mathbb{R}^d$. The loss function is the mean squared error*

$$\mathcal{L}(\theta) = \frac{1}{2} \sum_{i=1}^n \|f_{\theta}(x_i) - y_i\|^2. \quad (26)$$

Assuming that the network width $d \rightarrow \infty$, by the infinite NTK theory, the NTK

$$\Theta_{\theta_0}(x, x') = [\nabla_{\theta} f_{\theta_0}(x)]^\top [\nabla_{\theta} f_{\theta_0}(x')] \in \mathbb{R}^{d \times d} \quad (27)$$

is deterministic and constant during the fine-tuning. Suppose the learning rate η satisfies $\eta = O(1/\|\Theta_{\theta_0}\|_{\text{op}})$, where $\|\cdot\|_{\text{op}}$ is the operator norm of the NTK Gram matrix. After one gradient descent step, we have

$$\theta_1 = \theta_0 - \eta \nabla_{\theta} \mathcal{L}(\theta_0). \quad (28)$$

Specifically, the updated function satisfies the following affinity form

$$f_{\theta_1}(x) = A[f_{\theta_0}](x) + b(x), \quad \forall x \in \mathcal{X}, \quad (29)$$

where $A : \mathcal{F} \rightarrow \mathcal{F}$ is the linear operator

$$A[f](x) = f(x) - \eta \sum_{i=1}^n \Theta_{\theta_0}(x, x_i) f(x_i), \quad (30)$$

and $b(x) = \eta \sum_{i=1}^n \Theta_{\theta_0}(x, x_i) y_i$ is the input-dependent bias function.

Proof. The gradient of $\mathcal{L}$ at θ_0 is

$$\nabla_{\theta} \mathcal{L}(\theta_0) = \sum_{i=1}^n [\nabla_{\theta} f_{\theta_0}(x_i)] (f_{\theta_0}(x_i) - y_i). \quad (31)$$

Let $J_i = \nabla_{\theta} f_{\theta_0}(x_i) \in \mathbb{R}^{p \times d}$ be the Jacobian matrix and $r_i = f_{\theta_0}(x_i) - y_i \in \mathbb{R}^d$ be the residual terms. Then, we have

$$\nabla_{\theta} \mathcal{L}(\theta_0) = \sum_{i=1}^n J_i r_i. \quad (32)$$

The parameter updating is

$$\theta_1 - \theta_0 = -\eta \sum_{i=1}^n J_i r_i. \quad (33)$$

For any $x \in \mathcal{X}$, with the first-order Taylor expansion of $f(x)$, we have

$$f_{\theta_1}(x) - f_{\theta_0}(x) = J_x^\top (\theta_1 - \theta_0) + O(\|\theta_1 - \theta_0\|^2), \quad (34)$$

where $J_x = \nabla_\theta f_{\theta_0}(x)$. Substituting (32) into the (34), we obtain

$$f_{\theta_1}(x) - f_{\theta_0}(x) = -\eta \sum_{j=1}^n J_x^\top J_j r_j + O(\eta^2). \quad (35)$$

By the NTK definition $J_x^\top J_j = \Theta_{\theta_0}(x, x_j)$, (35) can be formulated by

$$f_{\theta_1}(x) - f_{\theta_0}(x) = -\eta \sum_{j=1}^n \Theta_{\theta_0}(x, x_j) r_j + O(\eta^2), \quad (36)$$

which can be further reformulated by

$$\begin{aligned} f_{\theta_1}(x) - f_{\theta_0}(x) &= -\eta \sum_{j=1}^n \Theta_{\theta_0}(x, x_j) f_{\theta_0}(x_j) \\ &\quad + \eta \sum_{j=1}^n \Theta_{\theta_0}(x, x_j) y_j + O(\eta^2). \end{aligned} \quad (37)$$

As $d \rightarrow \infty$, $O(\eta^2) \rightarrow 0$, we have

$$\begin{aligned} f_{\theta_1}(x) &= f_{\theta_0}(x) \\ &\quad - \eta \sum_{i=1}^n \Theta_{\theta_0}(x, x_i) f_{\theta_0}(x_i) + \eta \sum_{i=1}^n \Theta_{\theta_0}(x, x_i) y_i. \end{aligned} \quad (38)$$

This yields the affinity form $f_{\theta_1}(x) = A[f_{\theta_0}](x) + b(x)$. To demonstrate the linearity of A , for any $f, g \in \mathcal{F}$ and $\alpha, \beta \in \mathbb{R}$, we have

$$\begin{aligned} A[\alpha f + \beta g](x) &= (\alpha f(x) + \beta g(x)) - \eta \sum_{i=1}^n \Theta_{\theta_0}(x, x_i) (\alpha f(x_i) + \beta g(x_i)) \\ &= \alpha \left(f(x) - \eta \sum_{i=1}^n \Theta_{\theta_0}(x, x_i) f(x_i) \right) + \\ &\quad \beta \left(g(x) - \eta \sum_{i=1}^n \Theta_{\theta_0}(x, x_i) g(x_i) \right) \\ &= \alpha A[f](x) + \beta A[g](x). \end{aligned} \quad (39)$$

Thus, A is a linear operator on $\mathcal{F}$. $\square$

Remark 1. In particular, the statement 3 claims that the updated function after one gradient descent step takes the affine form $f_{\theta_1}(x) = A[f_{\theta_0}](x) + b(x)$. However, the operator A is not equivalent to multiplication by a real-valued matrix $P \in \mathbb{R}^{d \times d}$, i.e., $f_{\theta_1}(x) \neq P f_{\theta_0}(x) + b(x)$. The reason is as follows. The operator A is an integral-type operator

(specifically, a discrete sum approximating an integral) that depends on the entire training set. It maps a function f to a new function $A[f]$ by combining pointwise evaluation at x with a weighted sum of evaluations at all training points x_i . The weights $\Theta_{\theta_0}(x, x_i) \in \mathbb{R}^{d \times d}$ are matrix-valued and vary with both x and x_i . This makes A a global operator that cannot be reduced to a pointwise matrix multiplication. Nonetheless, restricting x to the training set allows an equivalent representation using a real-valued matrix P . Below, we state and prove this as a new theorem.

Statement 4. Consider the training set $S = \{x_1, \dots, x_n\}$. Define the following notations:

1. The vector of function values: $\mathbf{f}_\theta = \begin{bmatrix} f_\theta(x_1) \\ \vdots \\ f_\theta(x_n) \end{bmatrix} \in \mathbb{R}^{nd}$.
2. The vector of labels: $\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^{n \cdot d}$.
3. The NTK Gram matrix $\Theta \in \mathbb{R}^{(nd) \times (nd)}$, which is a block matrix where the (k, j) -th block is $\Theta_{\theta_0}(x_k, x_j) \in \mathbb{R}^{d \times d}$.

After one gradient descent step with learning rate η , the updated function values on the training points satisfy:

$$\mathbf{f}_{\theta_1} = P \mathbf{f}_{\theta_0} + \mathbf{b},$$

where $P = I - \eta \Theta$ is a real-valued matrix, $\mathbf{b} = \eta \Theta \mathbf{y}$, and I is the identity matrix with dimension nd .

Proof. From the proof of Statement 3, for any $x \in \mathcal{X}$, the first-order Taylor expansion of $f(x)$ in the infinite-width limit ($d \rightarrow \infty$) follows

$$\begin{aligned} f_{\theta_1}(x) &= f_{\theta_0}(x) - \eta \sum_{j=1}^n \Theta_{\theta_0}(x, x_j) f_{\theta_0}(x_j) \\ &\quad + \eta \sum_{j=1}^n \Theta_{\theta_0}(x, x_j) y_j. \end{aligned} \quad (40)$$

Hereafter, by evaluating it at a training point x_k (where $k \in \{1, \dots, n\}$), we get

$$\begin{aligned} f_{\theta_1}(x_k) &= f_{\theta_0}(x_k) - \eta \sum_{j=1}^n \Theta_{\theta_0}(x_k, x_j) f_{\theta_0}(x_j) \\ &\quad + \eta \sum_{j=1}^n \Theta_{\theta_0}(x_k, x_j) y_j. \end{aligned} \quad (41)$$

Define the vector $\mathbf{f}_{\theta_1}$ by stacking $f_{\theta_1}(x_k)$ for $k = 1, \dots, n$. Then, the k -th block of $\mathbf{f}_{\theta_1}$ is

$$[\mathbf{f}_{\theta_1}]_k = f_{\theta_1}(x_k) \quad (42)$$

$$\begin{aligned} &= [\mathbf{f}_{\theta_0}]_k - \eta \sum_{j=1}^n \Theta_{\theta_0}(x_k, x_j) [\mathbf{f}_{\theta_0}]_j \\ &\quad + \eta \sum_{j=1}^n \Theta_{\theta_0}(x_k, x_j) y_j. \end{aligned} \quad (43)$$

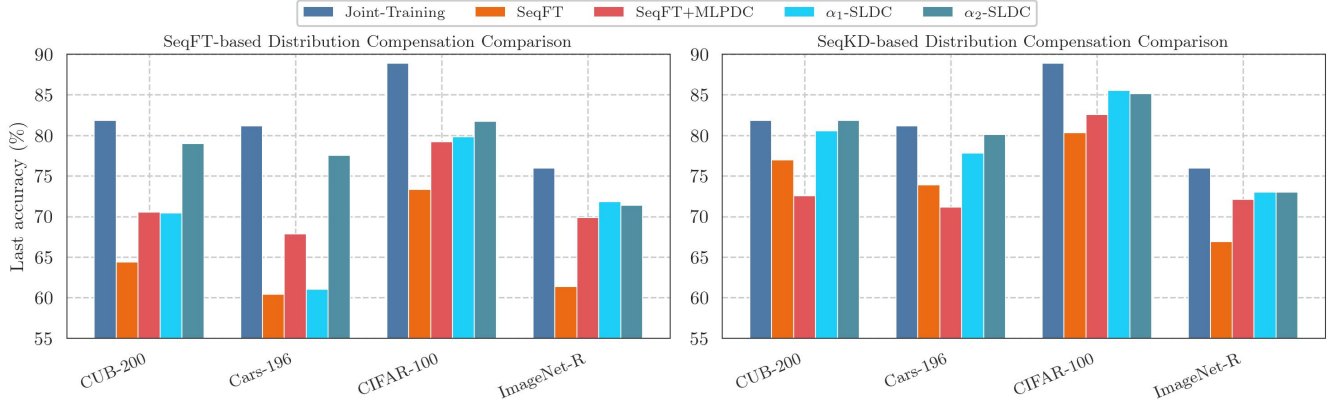
In matrix form, we get that the term $\sum_{j=1}^n \Theta_{\theta_0}(x_k, x_j)[\mathbf{f}_{\theta_0}]_j$ is the k th block of the matrix-vector product $\Theta \mathbf{f}_{\theta_0}$. Similarly, $\sum_{j=1}^n \Theta_{\theta_0}(x_k, x_j)y_j$ is the k th block of $\Theta \mathbf{y}$. Thus, the full vector updating $\mathbf{f}_{\theta_1} - \mathbf{f}_{\theta_0}$ satisfies

$$\mathbf{f}_{\theta_1} = \mathbf{f}_{\theta_0} - \eta \Theta \mathbf{f}_{\theta_0} + \eta \Theta \mathbf{y} = (I - \eta \Theta) \mathbf{f}_{\theta_0} + \eta \Theta \mathbf{y}. \quad (44)$$

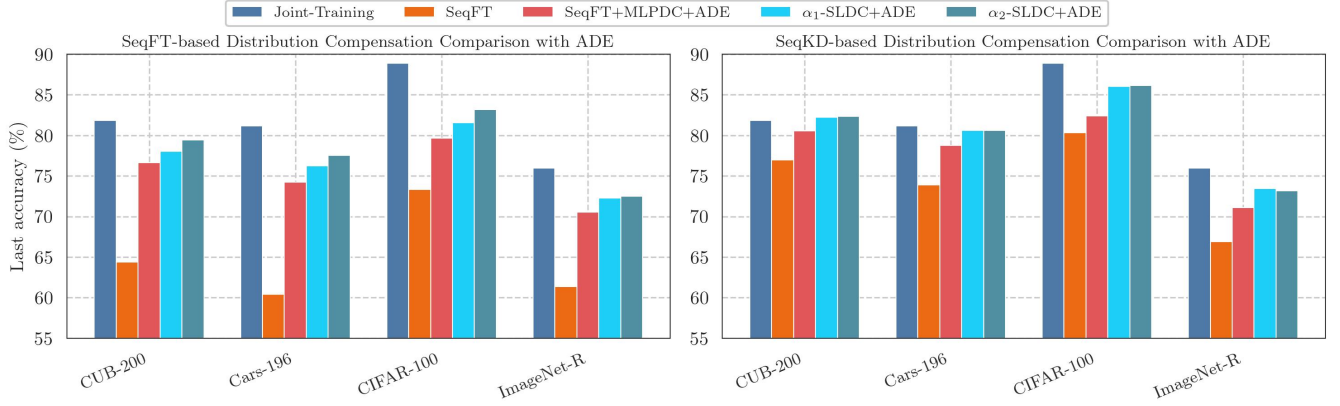
Let $P = I - \eta \Theta$ and $\mathbf{b} = \eta \Theta \mathbf{y}$, we obtain

$$\mathbf{f}_{\theta_1} = P \mathbf{f}_{\theta_0} + \mathbf{b}.$$

This is an affine transformation in $\mathbb{R}^{nd}$, which is parameterized by the real-valued matrix P and vector $\mathbf{b}$. $\square$

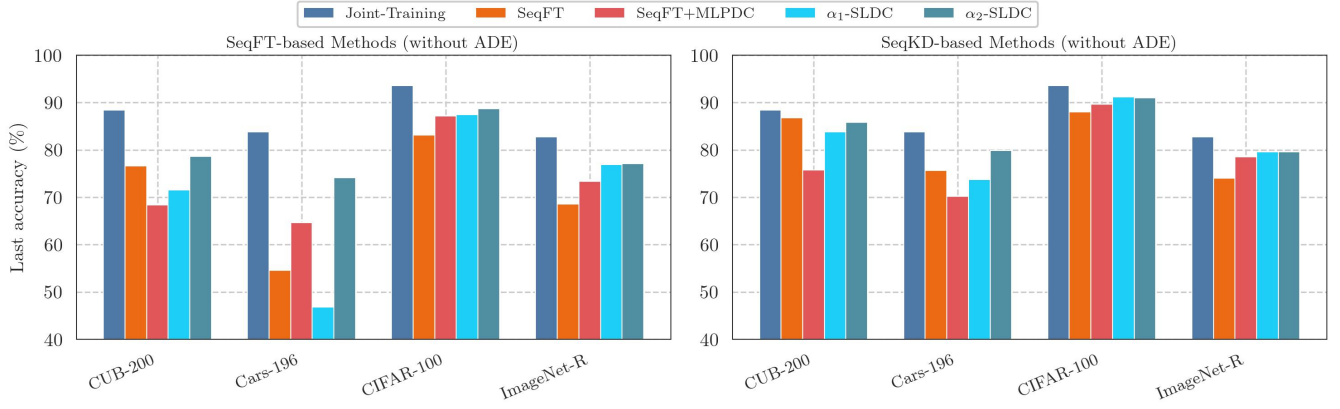


(a) Performance comparison on MoCo-V3 pre-trained ViT-B/16 without ADE

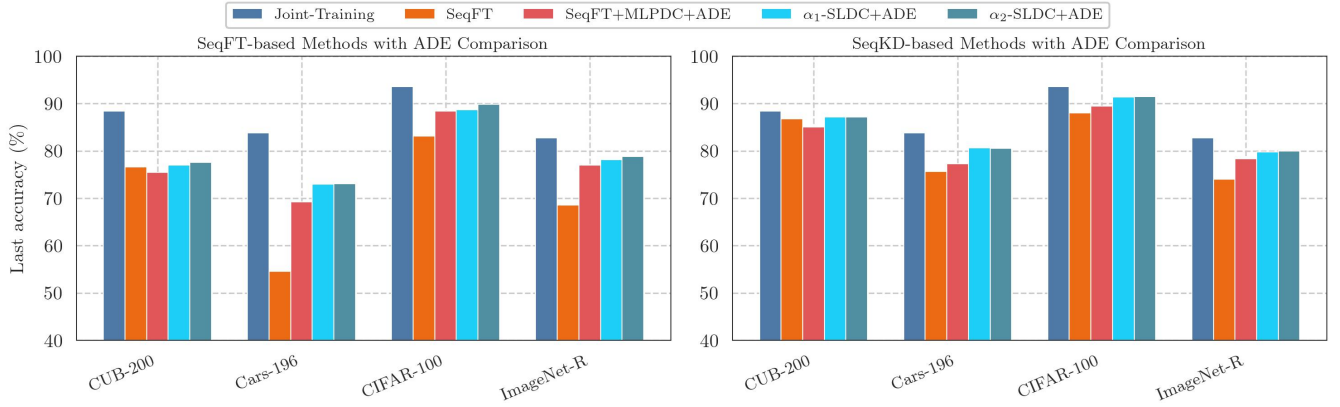


(b) Performance comparison on MoCo-V3 pre-trained ViT-B/16 with ADE

Figure 7: Comparative evaluation of CIL methods using self-supervised (MoCo-V3) pre-trained ViT-B/16 backbone. (a) Results without auxiliary data enrichment (ADE); (b) Results with ADE. (a) Performance comparison of SeqFT-based distribution compensation methods. (b) Performance comparison of SeqKD-based distribution compensation methods. Particularly, the results of joint-training serve as the performance upper bound for other methods.



(a) Performance comparison on ImageNet-21K pre-trained ViT-B/16 without ADE



(b) Performance comparison on ImageNet-21K pre-trained ViT-B/16 with ADE

Figure 8: Comparative evaluation of CIL methods using supervised (ImageNet-21K) pre-trained ViT-B/16 backbone. (a) Results without auxiliary data enrichment (ADE); (b) Results with ADE. (a) Performance comparison of SeqFT-based distribution compensation methods. (b) Performance comparison of SeqKD-based distribution compensation methods. Particularly, the results of joint-training serve as the performance upper bound for other methods.

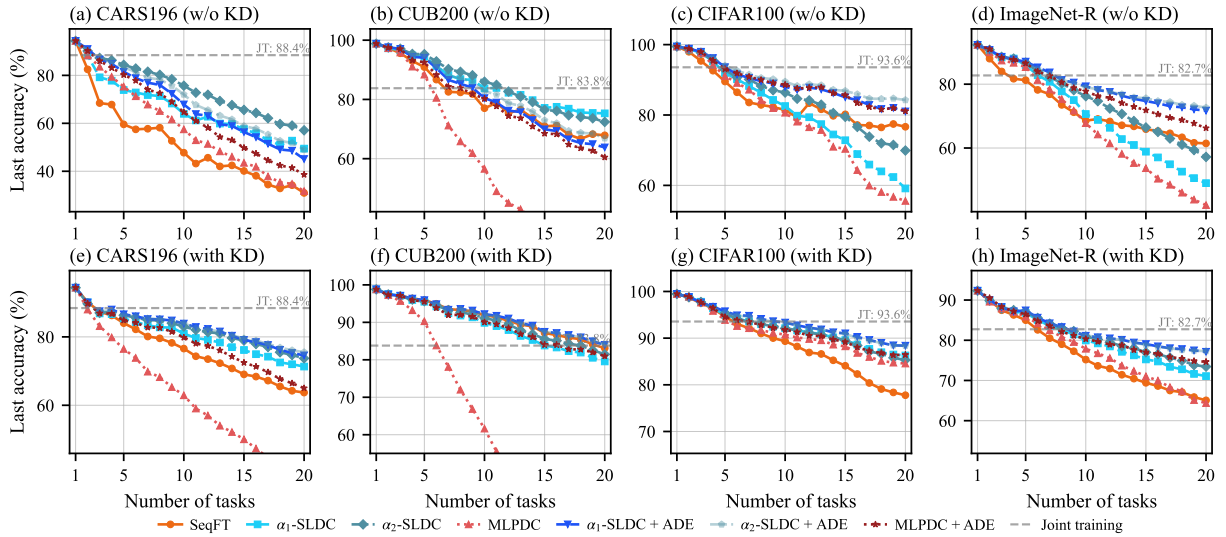


Figure 9: Performance comparison of SLDC methods on a 20-task sequence, demonstrating state-of-the-art results both with and without knowledge distillation.