This CVPR Findings paper is the Open Access version, provided by the Computer Vision Foundation. Except for this watermark, it is identical to the accepted version; the final published version of the proceedings is available on IEEE Xplore.

Bootstrap Your Own Classifier: Your Pretrained Vision Models are Secretly Strong Continual Learners

Yizheng Gong[1,2] Xiaoyang Wang[3] Siyue Yu[1] Waleed Al-Nuaimy[2] Jimin Xiao[1,*]

[1]Xi'an Jiaotong-Liverpool University [2]University of Liverpool [3]Shanghai Artificial Intelligence Laboratory

y.gong21@liverpool.ac.uk, wangxycc@outlook.com, siyue.yu02@xjtlu.edu.cn, wax@liverpool.ac.uk, jimin.xiao@xjtlu.edu.cn

Abstract

Continual learning requires models to sequentially acquire new knowledge while preserving previously learned information. Current approaches combine pretrained vision backbones with randomly initialized classifiers, creating an initialization asymmetry that limits performance. We propose Bootstrap Your Own Classifier (BYOC), which initializes classifiers using class prototypes extracted from the pretrained backbone, establishing a fully coherent initialization. This simple change dramatically stabilizes sequential fine-tuning. To address the gradual drift between the evolving backbone and previously learned classifiers, we employ exponential moving average updates for the backbone and calibrate historical classifiers based on current task adaptation patterns. Despite its simplicity, BYOC achieves state-of-the-art performance on CIFAR-100, ImageNet-R, ImageNet-A, CUB-200, and Stanford Cars, demonstrating that proper initialization from pretrained representations substantially outperforms complex parameter-efficient methods. Our work reveals that pretrained vision models are inherently strong continual learners when provided with consistent initialization across all components. Code is available at: https://github.com/jordangong/continual-learning.

1. Introduction

Imagine a student learning mathematics who suddenly forgets basic arithmetic upon studying calculus—this is the reality of catastrophic forgetting [12, 28] that plagues neural networks in continual learning [7, 44]. While humans seamlessly accumulate knowledge throughout their lives, artificial neural networks struggle to learn new tasks without destroying previously acquired capabilities. This fundamental limitation has persisted despite decades of research, blocking the path toward truly adaptive AI systems.

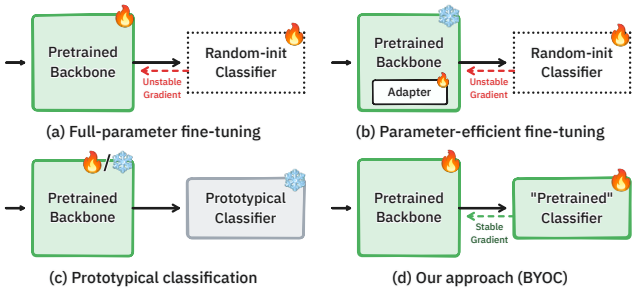


Figure 1. Comparison of continual learning paradigms revealing the initialization asymmetry problem. Dotted boxes indicate random initialization, while solid boxes show semantically initialized components. Red dashed arrows represent unstable gradient flow causing backbone disruption, while green dashed arrows indicate stable gradients. (a-b) Conventional approaches pair pretrained backbones with random classifiers, creating unstable optimization. (c) Prototypical classification avoids learning entirely. (d) BYOC achieves coherent initialization—both backbone and classifier are "pretrained", enabling stable joint optimization. Fire and snow symbols indicate trainable and frozen parameters.

The landscape changed dramatically with the advent of large-scale pretrained vision models [9]. Researchers discovered something remarkable: models pretrained on massive datasets exhibit unprecedented resistance to catastrophic forgetting [5, 33]. Yet paradoxically, even with these powerful pretrained backbones, continual learning performance remains far from satisfactory. Why do we still struggle when the foundation seems so solid?

The answer lies in an important but underexplored aspect of current methods. While researchers have focused on leveraging the stability of pretrained backbones, they typically pair them with randomly initialized classifiers—creating a fundamental asymmetry that may limit system performance (Fig. 1). This practice, carried over from from conventional transfer learning paradigms [22], overlooks a key distinction: in continual learning, we must sequentially adapt to multiple

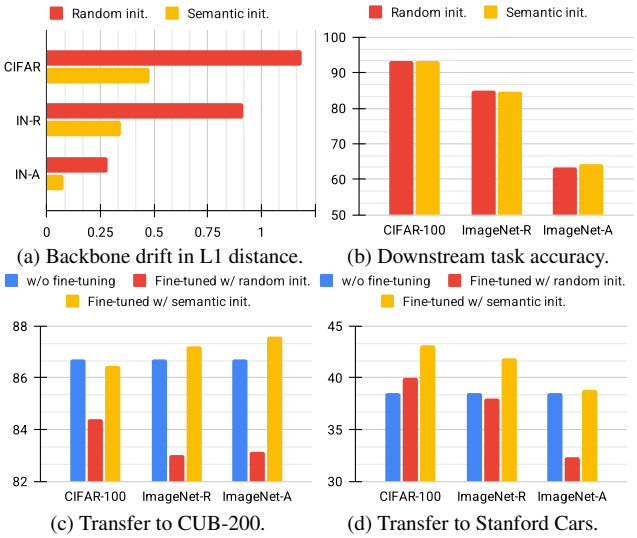
*Corresponding author.

tasks, not just one. Each random initialization introduces substantial perturbations to the carefully pretrained representations, requiring the backbone to compensate for arbitrary starting points rather than learning meaningful task-specific adaptations.

Consider what happens during the first gradient steps of training with a randomly initialized classifier (Fig. 1a). The dotted box and red dashed gradient arrow tell the story—the loss is dominated by the need to move classifier weights from their random positions to something sensible, propagating unstable gradients through the backbone and disrupting its pretrained structure. Researchers have attempted to shield the backbone through parameter-efficient fine-tuning methods [45, 46] (Fig. 1b), inserting adapters or prompts while freezing most parameters, yet the unstable gradients persist. Others have abandoned parametric classifiers entirely, using fixed prototypes extracted from frozen features [52] (Fig. 1c, shown in gray to indicate their immutable nature). But these solutions treat symptoms rather than the disease—they constrain the model's ability to adapt rather than addressing the root cause of instability.

What if we could eliminate this initialization asymmetry entirely? Our key insight is straightforward yet effective: *we can "pretrain" classifiers by initializing them with semantically meaningful weights derived from the backbone's own representations* (Fig. 1d). Notice how both components share the same green color, indicating coherent initialization, with stable green gradients flowing between them. By computing class prototypes [37]—the mean representations of training examples per class—we create classifier weights that already reside in the correct region of the representation space. This is not merely an engineering trick; our investigation reveals that this initialization significantly improves the optimization landscape.

Fig. 2 tells a compelling story through a carefully designed experiment. We take a ViT-B/16 backbone pretrained on ImageNet-21K and fine-tune it on three diverse target datasets (CIFAR-100, ImageNet-R, and ImageNet-A) using either random or semantic classifier initialization. The results are striking: random initialization causes 2.5-3.5× more backbone drift (Fig. 2a) while achieving similar target task performance (Fig. 2b). The true revelation comes from the zero-shot transfer of the fine-tuned backbone to held-out datasets (CUB-200 and Stanford Cars), where we extract class prototypes as classifier weights using the fine-tuned backbone to measure preservation of general representations (Fig. 2c and Fig. 2d). Random initialization consistently degrades transfer performance below the original pretrained model (w/o fine-tuning), indicating overfitting to the target task. In contrast, semantic initialization maintains or even improves transfer capabilities, suggesting the backbone learns broadly useful refinements rather than narrow task-specific adaptations. These findings reveal that proper initialization



(a) Backbone drift in L1 distance. (b) Downstream task accuracy.

(c) Transfer to CUB-200. (d) Transfer to Stanford Cars.

Figure 2. Empirical comparison of random versus semantic classifier initialization. We fine-tune a ViT-B/16 pretrained on ImageNet-21K on 3 target datasets: CIFAR-100, ImageNet-R, and ImageNet-A, comparing random-init vs. semantic-init classifiers. (a) Backbone drift: L1 distance between original and fine-tuned parameters ($\times 10^{-2}$). (b) Target task accuracy after fine-tuning. (c-d) Zero-shot transfer of the fine-tuned backbone to held-out datasets (CUB-200 and Stanford Cars) measures generalization, where we extracts class prototypes as classifier weights.

indeed changes what the model learns, not just the stability of learning.

Our method, Bootstrap Your Own Classifier (BYOC), turns continual learning into a different process. Instead of fighting against random initialization with architectural constraints [27, 45, 46] or learning rate tricks [50], we embrace full model adaptation from a coherent starting point. The pipeline is elegantly simple: extract prototypes using the pretrained backbone, initialize classifiers with these prototypes, then jointly fine-tune both components. This approach unlocks the full potential of pretrained models while keeping the stability that make continual learning possible.

But our innovation does not stop at initialization. Sequential learning introduces a subtle yet critical challenge: as the backbone evolves to accommodate new tasks, it drifts away from the representation space where previous classifiers were optimized. We address this through two synergistic mechanisms. In details, we 1) employ exponential moving average (EMA) [15] updates for the backbone, smoothing its trajectory through representation space, and 2) introduce a novel classifier calibration strategy that adjusts previous classifiers based on observed adaptation patterns in current task, keeping alignment without requiring access to historical data.

The results speak for themselves. Across challenging benchmarks spanning ImageNet variants [17, 18, 24] and fine-grained recognition [23, 41], BYOC achieves state-of-

the-art performance through what is arguably the simplest approach in recent continual learning literature. No architectural modifications, no carefully tuned hyperparameters, just principled initialization and thoughtful adaptation.

Our contributions are threefold:

- We reveal the impact of initialization asymmetry on continual learning stability, showing through systematic experimentation that prototype-based initialization creates a more favorable optimization landscape for sequential learning compared to random initialization.
- We introduce the first continual learning framework that fully embraces pretrained knowledge in both backbone and classifiers, showing that simple sequential fine-tuning is highly effective when started from the right point.
- We develop practical solutions for backbone-classifier alignment through EMA updates and similarity-based calibration, maintaining system coherence throughout the continual learning process.

2. Related Work

Traditional continual learning. Early continual learning research focused on training neural networks from scratch while mitigating catastrophic forgetting through three main paradigms. Regularization-based methods like LwF [26], EWC [21], SI [49], and MAS [1] constrain parameter updates to preserve previous knowledge. Replay-based approaches including iCaRL [34], A-GEM [4] BiC [47], and DER [2] maintain memory buffers of previous examples, with GDumb [31] demonstrating that simple replay can outperform sophisticated methods. Architecture-based methods like PNN [35], DEN [48], PodNet [10], and DyTox [11] modify network structure to accommodate new tasks. While foundational, these methods struggle to match the performance of pretrained model approaches.

PEFT-based continual learning. The emergence of powerful pretrained models has shifted continual learning toward parameter-efficient adaptation [13, 19, 20]. Prompt-based methods learn task-specific prompts while freezing the backbone, including L2P [46], DualPrompt [45], CODA-Prompt [36], HiDe-Prompt [43], and SDP [38]. Adapter-based approaches insert trainable modules, with LAE [14], EASE [51], InfLoRA [27], MOS [39], DIA [25], and SEMA [42] achieving strong performance through various mechanisms for managing task-specific parameters. However, recent analyses [40] suggest their success may stem primarily from restricting the parameter space rather than the proposed mechanisms, which motivates our full fine-tuning approach.

Full fine-tuning with pretrained models. Recent work demonstrates that pretrained models fundamentally alter continual learning dynamics [6, 30, 32]. Zhou et al. [52] introduced prototypical classifiers to continual learning through SimpleCIL (training-free) and APER (with fine-tuning), showing strong performance with minimal complexity. SLCA [50] shows that differential learning rates enable effective full fine-tuning, though it implicitly implements feature replay. RanPAC [29] bypasses forgetting through random projections. Our BYOC advances beyond these methods by demonstrating that full fine-tuning can be highly effective when combined with semantic initialization and explicit alignment strategies, addressing the initialization asymmetry that previous approaches overlooked and achieving superior performance without replay or parameter restrictions.

3. Method

We first formalize the continual learning problem setting, then detail our proposed Bootstrap Your Own Classifier.

3.1. Problem Formulation

Continual learning evaluates a model's ability to sequentially acquire knowledge from a stream of tasks without forgetting previously learned information. We focus on class-incremental continual learning, where the dataset $\mathcal{D}$ is partitioned into T tasks based on class membership. For task t, we denote $\mathcal{D}_t = \{(x, y) \mid (x, y) \in \mathcal{X}_t \times \mathcal{Y}_t\}$, where $\mathcal{X}_t$ and $\mathcal{Y}_t$ represent the input and label spaces. The key constraint is that label spaces are disjoint: $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$, meaning each class appears in exactly one task. The complete dataset satisfies $\mathcal{D} = \bigcup_{t=1}^{T} \mathcal{D}_t$ and $\mathcal{Y} = \bigcup_{t=1}^{T} \mathcal{Y}_t$.

Following recent paradigms in continual learning, we employ a pretrained backbone $f_{\theta_0} : \mathcal{X} \to \mathbb{R}^d$ with initial parameters θ_0, where d denotes the representation dimension. For each task t, we attach a task-specific classifier $g_{w_t} : \mathbb{R}^d \to \mathbb{R}^{|\mathcal{Y}_t|}$ with parameters w_t. The model sequentially learns tasks 1 through T, updating the backbone parameters from θ_0 to θ_T while acquiring classifier parameters $\{w_1, \ldots, w_T\}$. The objective is to minimize the expected loss across all observed tasks: $\mathcal{L} = \mathbb{E}_{t\in[1,T]}\mathbb{E}_{(x,y)\in\mathcal{D}_t}[\ell(g_{w_t}(f_{\theta_T}(x)), y)]$, where ℓ denotes the task-specific loss function.

3.2. Bootstrap Your Own Classifier

Building upon the empirical findings presented in the introduction, we present Bootstrap Your Own Classifier (BYOC), a continual learning framework that leverages semantically initialized classifiers to unlock the full potential of pretrained vision models. Our approach initializes task-specific classifiers using class prototypes extracted from the pretrained backbone, creating a fully coherent system that maintains stability throughout sequential learning. To address the inevitable drift between the evolving backbone and previously learned classifiers, we introduce two complementary mechanisms: exponential moving average updates for backbone parameters and a calibration strategy for historical classifiers.

Pipeline. Our continual learning pipeline operates through three distinct phases, as illustrated in Fig. 3:

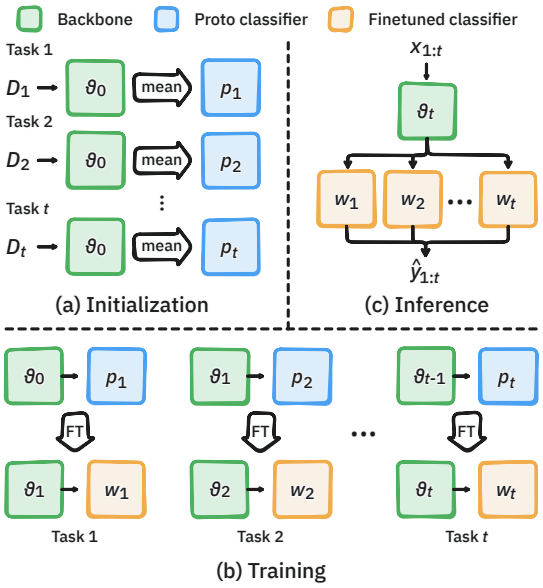


Figure 3. The Bootstrap Your Own Classifier (BYOC) framework. Our method consists of three key stages: (a) Initialization: Class prototypes are extracted using the original pretrained backbone θ_0 for each task on-demand, serving as semantically meaningful classifier initializations. (b) Training: Each task involves joint fine-tuning of the backbone (inherited from the previous task) and a prototype-initialized classifier, creating a sequence of specialized classifiers paired with an evolving shared backbone. (c) Inference: The current backbone processes inputs through all previously learned classifiers to generate predictions across the entire learned label space.

Initialization Phase Fig. 3a): When encountering task t, we first extract class prototypes from the training data $\mathcal{D}_t$ using the original pretrained backbone f_{θ_0}. For each class $c \in \mathcal{Y}_t$, we compute the prototype as:

$$p_t^{(c)} = \frac{1}{|\mathcal{D}_t^{(c)}|} \sum_{(x,y)\in\mathcal{D}_t^{(c)}} f_{\theta_0}(x), \tag{1}$$

where $\mathcal{D}_t^{(c)} = \{(x, y) \in \mathcal{D}_t \mid y = c\}$. Critically, we consistently use the original pretrained backbone θ_0 for prototype extraction across all tasks, rather than the current backbone θ_{t-1}. This design choice prevents the accumulation of representational drift, as empirical evidence shows that task-adapted backbones become increasingly specialized, leading to suboptimal prototype initialization for subsequent tasks. The extracted prototypes $\{p_t^{(c)}\}_{c\in\mathcal{Y}_t}$ form the initial weights for the task-t classifier.

Training Phase (Fig. 3b): For each task t, we jointly fine-tune the backbone parameters (initialized from θ_{t-1}) and the classifier weights (initialized from prototypes p_t). This joint optimization updates:

$$\theta_{t-1} \to \theta_t \quad \text{and} \quad p_t \to w_t, \tag{2}$$

through standard gradient descent on the task-specific loss ℓ_t. Unlike parameter-efficient approaches that freeze the backbone or restrict updates to small modules, our method allows full adaptation of both components, leveraging the stability provided by semantic initialization. Each task produces a specialized classifier g_{w_t} that remains fixed after training, while the backbone continues to evolve through later tasks.

Inference Phase (Fig. 3c): During evaluation, an input x is processed through the current backbone f_{θ_t} (where t denotes the current number of learned tasks) and all previously learned classifiers $\{g_{w_1}, \ldots, g_{w_t}\}$. The prediction is obtained by aggregating logits across all classifiers:

$$\hat{y} = \arg\max_{c\in\mathcal{Y}_{1:t}} \left[g_{w_1}(f_{\theta_t}(x)) \oplus \cdots \oplus g_{w_t}(f_{\theta_t}(x))\right], \tag{3}$$

where $\oplus$ denotes concatenation of logit vectors, and $\mathcal{Y}_{1:t} = \bigcup_{t=1}^{t} \mathcal{Y}_t$ represents the complete label space learned so far.

Backbone-classifier misalignment. A fundamental challenge in our framework emerges from the sequential nature of learning: as the backbone adapts to new tasks, it gradually drifts from the representation space where previous classifiers were optimized. As illustrated in Fig. 4a, when learning task 2, the backbone parameters evolve from θ_1 to θ_2 to minimize the task-2 loss. While θ_2 aligns well with its corresponding classifier w_2, it has drifted from the task-1 classifier w_1, which was optimized for θ_1. This misalignment accumulates across tasks, potentially degrading performance on earlier tasks despite the absence of catastrophic forgetting in the traditional sense.

Backbone Stabilization via EMA: To mitigate rapid representational drift, we employ exponential moving average (EMA) updates for the backbone parameters, inspired by self-supervised learning methods such as MoCo [16], BYOL [15] and DINO [3]. Before training on task t, we create a teacher copy θ'_{t-1} of the current backbone θ_{t-1}. During training, after each gradient update to the student backbone:

$$\theta_{t-1} \leftarrow \theta_{t-1} - \eta\nabla_{\theta_{t-1}}\ell_t, \tag{4}$$

we update the teacher using momentum:

$$\theta'_{t-1} \leftarrow \alpha\theta'_{t-1} + (1-\alpha)\theta_{t-1}, \tag{5}$$

where $\alpha \in [0, 1]$ is the momentum coefficient (typically set close to 1). At the end of task t, we use the teacher parameters θ'_t as the backbone for evaluation and subsequent task learning. This mechanism ensures that the backbone evolves smoothly while incorporating knowledge from all gradient updates, effectively ensemble the trajectory of optimization rather than using only the final checkpoint.

Classifier Calibration: While EMA reduces the rate of backbone drift, we further address the backbone-classifier misalignment by adapting previous classifiers to the evolved representation space. Our calibration strategy leverages a key

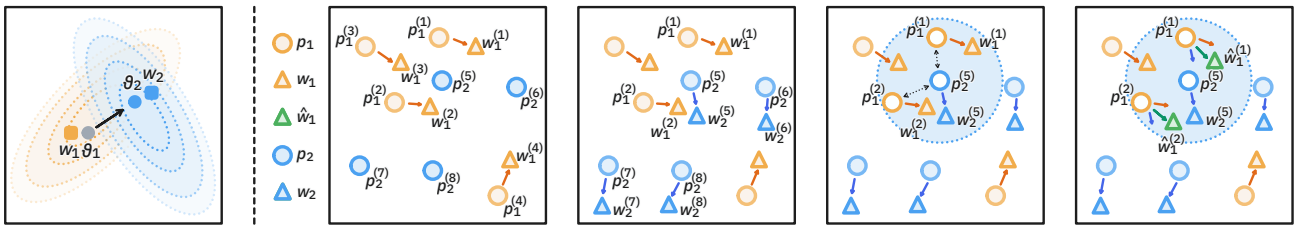


Figure 4. Illustration of backbone-classifier misalignment and our calibration strategy. (a) As the backbone adapts from θ_1 to θ_2 during task-2 learning, it drifts away from the task-1 classifier w_1, creating misalignment in the representation space. (b-e) Our classifier calibration process: (b) After learning task 1, we have prototypes $p_1^{(1)}$ through $p_1^{(4)}$ (yellow circles) and their corresponding fine-tuned weights $w_1^{(1)}$ through $w_1^{(4)}$ (yellow triangles), with transformation vectors showing the adaptation pattern. Task 2 prototypes $p_2^{(5)}$ through $p_2^{(8)}$ (blue circles) are initialized. (c) After task 2 training, the task 2 prototypes are transformed to fine-tuned weights $w_2^{(5)}$ through $w_2^{(8)}$ (blue triangles), while task 1 weights remain unchanged. (d) We compute similarities between all task 1 and task 2 prototypes—the dashed circle around $p_2^{(5)}$ is shown for illustration only; in practice, all task 2 prototypes contribute to each task 1 class calibration based on their similarities. (e) We apply weighted combinations of task 2 transformation vectors (from p_2 to w_2) to task 1 weights, yielding calibrated weights $\hat{w}_1^{(1)}$ through $\hat{w}_1^{(4)}$ (green triangles) that better align with the updated backbone θ_2.

design choice: since all prototypes across tasks are extracted using the original pretrained backbone θ_0, they exist in a unified representation space. This enables us to meaningfully compare and transfer adaptation patterns (from prototype to fine-tuned weights) across tasks. When the backbone evolves for a new task, we calibrate previous classifiers by applying weighted transformations from similar current-task classes, effectively propagating the learned adaptation to maintain alignment. Fig. 4b to Fig. 4e illustrate this calibration process.

After learning task t, we observe the transformation from prototypes p_t to fine-tuned weights w_t (Fig. 4b and Fig. 4c). For each class c in a previous task $s < t$, we calibrate its weights by aggregating transformations from all classes in task t, weighted by their similarity (Fig. 4d):

$$\hat{w}_s^{(c)} = w_s^{(c)} + \sum_{k \in \mathcal{Y}_t} \lambda_{ck} \cdot (w_t^{(k)} - p_t^{(k)}), \tag{6}$$

where λ_{ck} represents the similarity weight between class c from task s and class k from task t, computed as:

$$\lambda_{ck} = \frac{\exp(\text{sim}(p_s^{(c)}, p_t^{(k)})/\tau)}{\sum_{j \in \mathcal{Y}_t} \exp(\text{sim}(p_s^{(c)}, p_t^{(j)})/\tau)}. \tag{7}$$

Here, $\text{sim}(\cdot, \cdot)$ denotes cosine similarity and τ is a temperature parameter. This calibration mechanism (Fig. 4e) allows previous classifiers to adapt to the evolved representation space by learning from the adaptation patterns of similar classes in the current task, maintaining alignment without requiring access to previous task data.

The calibrated weights $\hat{w}_s$ replace the original fine-tuned weights w_s for all previous tasks $s < t$. This calibration process is executed after each task completion, ensuring that all historical classifiers remain aligned with the current backbone representation. Notably, since the calibration involves only arithmetic operations on existing weights without gradient computation, it can be performed offline or as a post-processing step, adding minimal computational overhead to the continual learning pipeline.

4. Experiments

This section details our experimental design, presents benchmark results, and provides an ablation analysis.

4.1. Experimental Setup

Datasets and evaluation. We evaluate on 5 standard benchmarks, three ImageNet variants: CIFAR-100 [24] (low-resolution 32×32 images), ImageNet-R [17] (artistic style renditions), and ImageNet-A [18] (natural adversarial examples) and two fine-grained datasets: CUB-200 [41] (200 bird species) and Stanford Cars [41] (196 car models). Following standard protocols, we partition each dataset into 10 sequential tasks, with additional 20-task splits for CIFAR-100 and ImageNet-R. We report last-task accuracy (*last*) and average accuracy across all tasks (*avg*) to capture both final performance and learning stability.

Baseline methods. We compare against three categories of continual learning approaches: 1) Parameter-Efficient Fine-Tuning (PEFT): Prompt-based methods including L2P [46], DualPrompt [45], and CODA-Prompt [36]; adapter-based approaches including LAE [14], EASE [51], InfLoRA [27], MOS [39], DIA [25], and SEMA [42]. 2) Full Fine-Tuning: Standard fine-tuning, APER w/ FT [52], Slow Learner, and SLCA [50]. 3) Training-Free: SimpleCIL [52], which uses fixed prototypes without parameter updates.

Implementation details. Following standard practice [46,

Table 1. Continual learning performance on CIFAR-100, ImageNet-R, and ImageNet-A across 10 and 20 task splits. We report last-task accuracy (*last*) and average accuracy across all tasks (*avg*). Results marked with an asterisk (∗) are from DIA, while dagger (†) indicates values from original papers. Best results are in **bold**, second-best results are underlined. Joint training performance represents the upper bound with simultaneous access to all data.

Method	CIFAR-100				ImageNet-R				ImageNet-A	
	10 tasks		*20 tasks*		*10 tasks*		*20 tasks*		*10 tasks*	
	last	*avg*	*last*	*avg*	*last*	*avg*	*last*	*avg*	*last*	*avg*
L2P* [46]	82.84	87.31	79.93	85.94	72.34	77.36	69.64	75.28	44.04	51.24
DualPrompt* [45]	83.44	88.76	81.15	87.87	69.10	74.28	66.61	72.45	53.19	61.47
CODA-P* [36]	87.13	91.75	81.96	89.11	73.30	78.47	69.96	75.34	52.08	63.92
LAE* [14]	85.33	89.96	83.69	87.86	72.39	79.07	69.86	77.38	47.18	58.15
EASE* [51]	87.76	92.35	85.80	91.51	76.17	81.73	65.23	77.45	55.04	65.34
InfLoRA* [27]	88.31	92.57	85.80	91.51	76.78	81.42	71.01	77.28	52.16	61.27
MOS† [39]	—	—	89.25	93.30	77.93	82.96	—	—	59.12	69.13
DIA* [25]	90.80	94.29	88.74	93.41	79.03	**85.61**	76.32	83.51	61.69	**71.58**
SEMA† [42]	87.84	92.23	—	—	74.82	81.39	69.60	77.84	51.35	62.50
Finetune	77.19	87.74	63.63	80.43	61.30	73.82	53.19	65.56	26.49	41.81
SimpleCIL [52]	80.07	84.87	80.07	85.26	52.78	57.89	52.78	58.66	47.59	56.89
APER w/ FT [52]	87.10	91.03	87.10	91.31	67.87	73.94	67.87	74.60	48.92	59.61
Slow Learner [50]	87.57	92.35	84.87	90.13	75.95	80.56	74.01	78.12	54.94	64.05
SLCA* [50]	91.26	94.09	90.08	93.85	77.00	81.17	74.63	79.92	60.63	70.04
BYOC (ours)	**91.45**	**94.34**	**90.33**	**94.17**	**80.47**	85.29	**78.39**	**83.54**	**62.82**	71.31
Joint	93.24	—	93.24	—	85.09	—	85.09	—	63.33	—

50, 52], we employ ViT-B/16 [9] pretrained on ImageNet-21K [8] as the backbone for all methods. We use SGD with momentum and train for 20 epochs per task with batch size 256 to ensure convergence. For BYOC, we set the learning rate to 0.005 without scheduling, while baseline methods use grid-searched optimal learning rates. Training augmentations include random resized crop to 224×224 (resize-only for CIFAR-100), random horizontal flip, and normalization with mean and standard deviation from pretraining protocols [9]. Evaluation uses center-cropped 224×224 images after resizing.

4.2. Benchmark Results

Tab. 1 and Tab. 2 show comprehensive comparisons across all benchmarks, with BYOC achieving state-of-the-art.

ImageNet variants (Tab. 1). BYOC achieves best performance on CIFAR-100 across all configurations, surpassing the previous best method SLCA by 0.19-0.25% on last-task accuracy while approaching the 93.24% joint training upper bound. On ImageNet-R, BYOC demonstrates substantial improvements with 80.47% (10-task) and 78.39% (20-task) last accuracy—outperforming SLCA by 3.47% and 3.76% respectively. The challenging ImageNet-A benchmark shows similar gains, with BYOC reaching 62.82% last accuracy, a 1.13% improvement over DIA. While DIA achieves marginally higher average accuracy on some configurations, this reflects PEFT methods' conservative learning; BYOC's superior last-task performance indicates better knowledge acquisition with competitive stability.

Table 2. Performance comparison on fine-grained recognition benchmarks CUB-200 and Stanford Cars with 10-task splits. Metrics include last-task accuracy (*last*) and average accuracy (*avg*). Notation follows Table 1.

Method	CUB-200		Stanford Cars	
	last	*avg*	*last*	*avg*
L2P*	79.62	85.76	38.18	51.79
DualPrompt*	81.10	88.23	40.14	56.74
EASE*	84.65	90.51	—	—
DIA*	86.73	**93.21**	—	—
Finetune	45.79	68.30	28.04	53.76
SimpleCIL	86.49	91.22	37.84	47.97
APER w/ FT	86.09	90.97	57.16	67.41
Slow Learner	75.49	84.69	58.43	70.99
SLCA†	84.71	90.94	67.73	76.93
BYOC (ours)	**87.14**	92.85	**68.82**	**77.89**
Joint	88.42	—	86.85	—

Fine-grained recognition (Tab. 2). On CUB-200, BYOC achieves 87.14% last accuracy, closely approaching the 88.42% joint ceiling and outperforming all baselines including SimpleCIL's competitive 86.49%. While SimpleCIL's

Table 3. Component-wise ablation of BYOC on CIFAR-100 and ImageNet-R (10 tasks). We evaluate the individual and combined contributions of semantic initialization (sem. init.), exponential moving average updates (EMA), and classifier calibration (calib.).

	Method	CIFAR-100		ImageNet-R	
		last	*avg*	*last*	*avg*
Baseline	Finetune	76.84	87.29	61.30	73.82
Ours	w/ sem. init.	89.92	93.23	79.05	83.79
	w/ EMA	80.57	89.72	67.12	74.85
	w/ sem. init. + EMA	90.93	94.04	79.69	84.53
	w/ sem. init. + calib.	90.42	93.69	79.33	84.19
	w/ all (a.k.a BYOC)	**91.45**	**94.34**	**80.47**	**85.29**

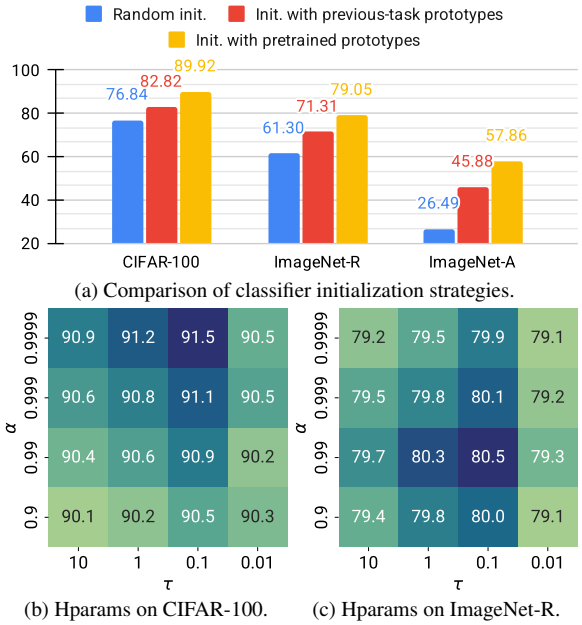
strong performance demonstrates the value of prototype representations, BYOC extends this insight through learnable prototype-initialized classifiers that adapt during training. Stanford Cars shows more dramatic improvements: BYOC achieves 68.82% last accuracy (1.09% over SLCA) and remarkably outperforms standard fine-tuning by 40.78%, highlighting how semantic initialization transforms continual learning dynamics on fine-grained tasks.

Notably, SLCA implements implicit feature replay by sampling from recorded class statistics, making it an exceptionally strong baseline. BYOC's consistent improvements without any replay mechanism underscore the effectiveness of semantic initialization and alignment strategies.

4.3. Ablation Studies

Tab. 3 decomposes BYOC's performance gains by evaluating individual components. Semantic initialization alone provides substantial improvements over standard fine-tuning, increasing last-task accuracy by 13.08% on CIFAR-100 and 17.75% on ImageNet-R, validating our core hypothesis that prototype-based initialization creates a more stable optimization landscape. The EMA mechanism in isolation yields modest gains (3.73% on CIFAR-100, 5.82% on ImageNet-R), confirming that backbone stabilization alone is insufficient without proper initialization. However, combining semantic initialization with EMA produces synergistic effects, achieving 90.93% on CIFAR-100—a 1.01% improvement over semantic initialization alone. Classifier calibration provides additional refinement when combined with semantic initialization, and the complete BYOC framework achieves optimal performance by addressing both initialization and alignment challenges, reaching 91.45% and 80.47% last-task accuracy on CIFAR-100 and ImageNet-R.

Initialization strategy analysis (Fig. 5a). Comparing three initialization (without EMA or classifier calibration) approaches reveals that using the original pretrained backbone for prototype extraction is crucial. This approach achieves 90.01% on CIFAR-100 compared to 82.82% with



(a) Comparison of classifier initialization strategies.

(b) Hparams on CIFAR-100.

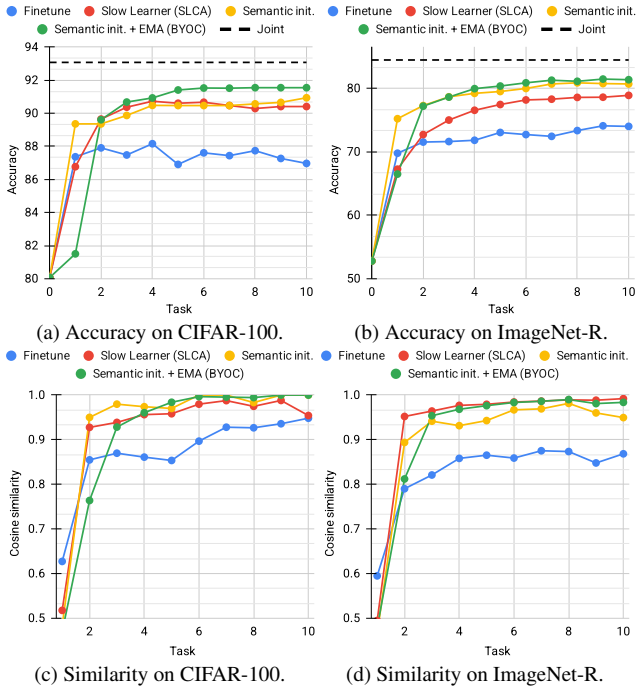
(c) Hparams on ImageNet-R.

Figure 5. Ablation studies on key design choices. (a) Comparison of classifier initialization strategies: random, previous-task backbone prototypes, and original pretrained backbone prototypes. (b-c) Sensitivity analysis of hyperparameters (Hparams) on CIFAR-100 and ImageNet-R.

previous-task prototypes and 76.84% with random initialization—improvements of 13.17%, 17.75%, and 31.37% over random initialization on CIFAR-100, ImageNet-R, and ImageNet-A, respectively. This confirms that maintaining consistency with the pretrained representation space, rather than adapting to potentially drifted task-specific representations, is essential for stable continual learning.

Hyperparameter robustness (Fig. 5b and Fig. 5c). BYOC demonstrates remarkable stability across hyperparameter settings. The temperature parameter τ and momentum coefficient α show consistent performance across wide ranges, with variations of only $\sim$1.4% between best and worst configurations. Optimal settings vary slightly by dataset ($\alpha = 0.9999$ for CIFAR-100 vs. $\alpha = 0.99$ for ImageNet-R), potentially reflecting different degrees of distribution shift from pretraining data.

Representation quality analysis (Fig. 6). We evaluate backbone representation quality through prototype-based probing (Fig. 6a and Fig. 6b), where class prototypes extracted as classifier weights assess the backbone's discriminative power on the entire dataset after each task. BYOC consistently achieves the highest probing accuracy, with particularly pronounced advantages on ImageNet-R. Representation stability analysis (Fig. 6c and Fig. 6d) reveals the stability-plasticity trade-off: both semantic initialization variants exhibit high stability while achieving superior accuracy compared to Slow Learner. On ImageNet-R, the stabilizing effect of EMA is



(a) Accuracy on CIFAR-100. (b) Accuracy on ImageNet-R.

(c) Similarity on CIFAR-100. (d) Similarity on ImageNet-R.

Figure 6. Representation quality analysis on CIFAR-100 and ImageNet-R. (a-b) Prototype-based probing accuracy on the entire dataset after each task, where prototypes are extracted as classifier weights to evaluate the backbone's discriminative power throughout continual learning. (c-d) Cosine similarity between representations from consecutive tasks measured on the test set, where higher values indicate greater stability.

evident as semantic init. + EMA maintains higher similarity than semantic initialization alone. Crucially, while Slow Learner achieves comparable stability to BYOC, its accuracy remains substantially lower, revealing that it sacrifices plasticity for stability. In contrast, BYOC achieves the optimal balance between stability and knowledge acquisition.

EMA application analysis (Tab. 4). We investigate whether EMA universally improves performance or specifically benefits backbone stabilization. Applying EMA to classifiers alone degrades performance relative to standard training (-2.12% drop on CIFAR-100), while applying it to both components similarly underperforms. This confirms that EMA's benefit is specifically tied to backbone stabilization rather than being a general performance enhancement technique. The optimal configuration—EMA on backbone only—aligns with our motivation of maintaining backbone stability while allowing classifiers to fully adapt to their specific tasks.

Interaction with Slow Learner (Tab. 5). Our final ablation examines whether BYOC benefits from differential learning rates between backbone and classifier (Slow Learner approach [50]). Across backbone learning rate multipliers spanning four orders of magnitude, performance variations remain within 0.3%, indicating that BYOC's stabilization mechanisms already achieve optimal balance between stability and plasticity. This robustness to learning rate ratios further simplifies hyperparameter selection and confirms that semantic initialization with proper alignment supersedes the need for careful learning rate tuning.

Table 4. Analysis of EMA application to different model components. Results show EMA specifically benefits backbone parameters, while applying it to classifiers or both components degrades performance, validating our targeted stabilization approach.

Backbone	Classifier	CIFAR-100		ImageNet-R	
		last	*avg*	*last*	*avg*
✗	✗	90.42	93.69	79.33	84.19
✗	✓	89.33	91.25	78.86	83.77
✓	✗	**91.45**	**94.34**	**80.47**	**85.29**
✓	✓	89.87	92.15	79.42	84.09

Table 5. Performance of BYOC combined with Slow Learner's differential learning rates. The learning rate multiplier controls the ratio between backbone and classifier learning rates. Minimal performance variation across four orders of magnitude confirms that BYOC's stabilization mechanisms eliminate the need for careful learning rate balancing.

Method	CIFAR-100		ImageNet-R	
	last	*avg*	*last*	*avg*
BYOC	91.45	94.34	80.47	85.29
BYOC + SL@0.001	91.22	94.12	80.24	85.20
BYOC + SL@0.01	91.50	94.32	80.45	85.34
BYOC + SL@0.1	91.46	94.35	80.51	85.31
BYOC + SL@10	91.42	94.40	80.49	85.28

5. Conclusion

We presented Bootstrap Your Own Classifier (BYOC), which achieves state-of-the-art continual learning performance through a simple insight: semantic initialization of classifiers from pretrained backbones dramatically improves sequential learning stability. Combined with EMA and classifier calibration, BYOC outperforms complex methods despite its simplicity. Our results reveal that pretrained vision models possess strong inherent continual learning capabilities, previously untapped due to random initialization practices. BYOC demonstrates that effective continual learning requires no architectural modifications—proper initialization and alignment suffice to unlock the full potential of pretrained representations.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (No. 62471405, 62331003, 62301451), Basic Research Program of Jiangsu (BK20241814), Suzhou Basic Research Program (SYG202316) and XJTLU REF-22-01-010.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, pages 139–154, 2018. 3

[2] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *NeurIPS*, 2020. 3

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 4

[4] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019. 3

[5] Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. In *EMNLP*, pages 7870–7881, 2020. 1

[6] Andrea Cossu, Antonio Carta, Lucia Passaro, Vincenzo Lomonaco, Tinne Tuytelaars, and Davide Bacciu. Continual pre-training mitigates forgetting in language and vision. *Neural Networks*, 179:106492, 2024. 3

[7] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *TPAMI*, 44(7):3366–3385, 2022. 1

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 6

[9] 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 *ICLR*, 2021. 1, 6

[10] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. PodNet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, pages 86–102, 2020. 3

[11] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. DyTox: Transformers for continual learning with dynamic token expansion. In *CVPR*, pages 9285–9295, 2022. 3

[12] Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999. 1

[13] Zihao Fu, Haoran Yang, Anthony Man-Cho So, Wai Lam, Lidong Bing, and Nigel Collier. On the effectiveness of parameter-efficient fine-tuning. In *AAAI*, pages 12799–12807, 2023. 3

[14] Qiankun Gao, Chen Zhao, Yifan Sun, Teng Xi, Gang Zhang, Bernard Ghanem, and Jian Zhang. A unified continual learning framework with general parameter-efficient tuning. In *ICCV*, pages 11483–11493, 2023. 3, 5, 6

[15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 33, 2020. 2, 4

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 4

[17] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, pages 8340–8349, 2021. 2, 5

[18] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, pages 15262–15271, 2021. 2, 5

[19] 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 *ICLR*, 2022. 3

[20] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, pages 709–727, 2022. 3

[21] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *PNAS*, 114(13):3521–3526, 2017. 3

[22] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *ICCV*, pages 2661–2671, 2019. 1

[23] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, pages 554–561, 2013. 2

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images., 2009. 2, 5

[25] Jiashuo Li, Shaokun Wang, Bo Qian, Yuhang He, Xing Wei, Qiang Wang, and Yihong Gong. Dynamic integration of task-specific adapters for class incremental learning. In *CVPR*, pages 30545–30555, 2025. 3, 5, 6

[26] Zhizhong Li and Derek Hoiem. Learning without forgetting. *TPMI*, 40(12):2935–2947, 2017. 3

[27] Yan-Shuo Liang and Wu-Jun Li. InfLoRA: Interference-free low-rank adaptation for continual learning. In *CVPR*, pages 23638–23647, 2024. 2, 3, 5, 6

[28] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, pages 109–165. 1989. 1

[29] Mark D McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton Van den Hengel. RanPAC: Random projections and pre-trained models for continual learning. In *NeurIPS*, 2023. 3

[30] Sanket Vaibhav Mehta, Darshan Patil, Sarath Chandar, and Emma Strubell. An empirical investigation of the role of pre-training in lifelong learning. *JMLR*, 24(214):1–50, 2023. 3

[31] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. GDumb: A simple approach that questions our progress in continual learning. In *ECCV*, pages 524–540, 2020. 3

[32] Vinay V. Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. In *ICLR*, 2021. 3

[33] Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In *ICLR*, 2022. 1

[34] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCarl: Incremental classifier and representation learning. In *CVPR*, pages 2001–2010, 2017. 3

[35] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. In *NeurIPS Deep Learning Symposium*, 2016. 3

[36] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. CODA-Prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *CVPR*, pages 11909–11919, 2023. 3, 5, 6

[37] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 2

[38] Siqi Song, Limin Yu, and Jimin Xiao. SDP: Spectral-Decomposed Prompting for Continual Learning. In *ACMMM*, 2025. 3

[39] 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 *AAAI*, pages 20699–20707, 2025. 3, 5, 6

[40] Lukas Thede, Karsten Roth, Olivier J Hénaff, Matthias Bethge, and Zeynep Akata. Reflecting on the state of rehearsal-free continual learning with pretrained models. In *CoLLAs*, pages 1076–1093, 2024. 3

[41] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. CUB-200: A dataset for fine-grained bird classification. Technical report, 2011. 2, 5

[42] Huiyi Wang, Haodong Lu, Lina Yao, and Dong Gong. Self-expansion of pre-trained models with mixture of adapters for continual learning. In *CVPR*, pages 10087–10098, 2025. 3, 5, 6

[43] 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 *NeurIPS*, 2023. 3

[44] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *TPAMI*, 46(8):5362–5383, 2024. 1

[45] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. DualPrompt: Complementary prompting for rehearsal-free continual learning. In *ECCV*, pages 631–648, 2022. 2, 3, 5, 6

[46] 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 *CVPR*, pages 139–149, 2022. 2, 3, 5, 6

[47] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, pages 374–382, 2019. 3

[48] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *ICLR*, 2017. 3

[49] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, pages 3987–3995, 2017. 3

[50] 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 *ICCV*, pages 19148–19158, 2023. 2, 3, 5, 6, 8

[51] Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *CVPR*, pages 23554–23564, 2024. 3, 5, 6

[52] Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *IJCV*, 133(3):1012–1032, 2025. 2, 3, 5, 6