STREAM-AWARE PROMPT INFERENCE VIA SIMPLEX SEARCH FOR CLASS-INCREMENTAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Class-incremental learning (CIL) requires models to acquire new classes while retaining previously learned knowledge continually. With the emergence of powerful pre-trained models (PTMs), particularly Vision Transformers, CIL has increasingly shifted toward parameter-efficient adaptation, where frozen backbones are augmented with lightweight prompts or adapters. While these approaches have advanced rehearsal-free CIL, their inference mechanisms are predominantly designed for individual samples or task-specific routing, conventionally assuming an independent test stream. In practical deployments, however, incoming observations may exhibit local statistical coherence, sharing latent context such as task, domain, environment, or scene, while independent and coherent observations may also be interleaved. Our experiments reveal that such stream structure provides an additional source of information for CIL at inference time, yet it remains largely unexplored in prompt-based CIL. We introduce a stream-aware prompt inference paradigm that exploits local coherence when present while remaining effective for uncorrelated streams. We realize this through ARC (Anchor-Residual Composition), which decomposes each task prompt into a shared anchor and low-rank task residuals, yielding a linearly composable prompt space. Building on this representation, SPS (Simplex Prompt Search) performs entropy-driven optimization over convex combinations of task residuals, adapting inference to the incoming stream. For uncorrelated streams, SPS performs sample-wise optimization, while locally coherent streams enable batch-level optimization that exploits shared statistical structure. Across diverse distribution-shift CIL benchmarks, ARC consistently outperforms existing prompt- and adapter-based methods under both independent and locally coherent streams. Further analysis shows that the effectiveness of stream-aware prompt composition is governed by the geometry of the learned prompt space, highlighting stream structure and prompt geometry as complementary dimensions of inference in continual learning.

1 INTRODUCTION

Class-incremental learning (CIL) (Rebuffi et al., 2017; De Lange et al., 2021) requires models to continually learn new visual categories while retaining previously acquired knowledge, without access to task identity at test time. A central challenge is catastrophic forgetting (French, 1999; McCloskey & Cohen, 1989), traditionally addressed through rehearsal, knowledge distillation, or parameter regularization (Kirkpatrick et al., 2017; Li & Hoiem, 2017). The emergence of large-scale pre-trained models (PTMs) has shifted CIL toward parameter-efficient adaptation (Zhou et al., 2025), where a pre-trained Vision Transformer (ViT) (Dosovitskiy, 2020) is frozen and augmented with lightweight task-specific modules, preserving its generalizable representations while limiting learnable capacity. In response, two major adaptation paradigms have emerged: Prompt-based and Adapter-based.

Prompt-based methods learn tokens that steer the frozen backbone: L2P (Wang et al., 2022d) retrieves prompts through query-key matching, DualPrompt (Wang et al., 2022c) separates shared and task-specific prompts, CODA-Prompt (Smith et al., 2023) performs attention-weighted prompt composition, and APT (Chen et al., 2025) eliminates the prompt pool by additively injecting task prompts into the CLS token's key and value representations and merging them through exponential moving average (EMA). **Adapter-based methods** insert bottleneck modules into the backbone: EASE (Zhou et al., 2024) maintains task-specific adapters and concatenates their features, MOS (Sun

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

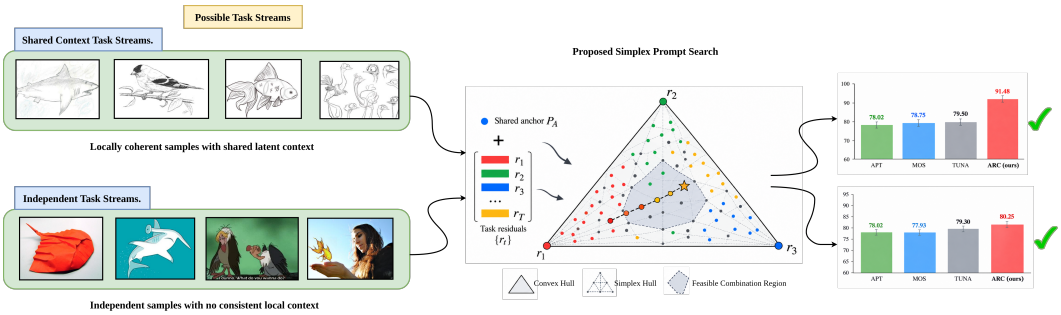


Figure 1: **Overview.** Test streams in CIL deployments range from locally coherent (shared latent context) to fully independent (no consistent local context). ARC decomposes task prompts into a shared anchor P_A and per-task residuals $\{r_t\}$, whose convex hull defines a prompt manifold. SPS searches this manifold via entropy-driven simplex optimization, adapting to whichever stream structure is present. ARC outperforms state-of-the-art prompts as well as adapter-based methods under both settings.

et al., 2025) combines task-specific adapters with EMA merging and self-refined retrieval, and TUNA (Wang et al., 2025) uses orthogonally constrained adapters, entropy-based selection, and universal adapter fusion.

These methods differ substantially in how task knowledge is stored, but share a common design principle in how it is retrieved at test time: inference operates on each sample independently. L2P and DualPrompt perform per-sample query-key matching. CODA-Prompt computes per-sample attention weights. APT applies a fixed EMA-merged prompt identically to every input. TUNA selects the best adapter for each sample via per-sample entropy scoring. In every case, the prediction for a given test image depends only on that image and the stored modules. This per-sample design has inherently become a convention in CIL evaluation without considering the structure of the incoming stream during inference.

For the first time, we question this assumption, based on the understanding that in practical deployments, test streams can exhibit *local statistical coherence*, where neighboring observations share latent context such as task, domain, environment, or scene. Surveillance cameras can observe the same scene over time, medical workflows can process scans from the same session, and autonomous vehicles can encounter consecutive frames with consistent environmental context. Importantly, such coherence does not provide task identity; it is simply structure inherent in the incoming stream. Existing per-sample inference mechanisms cannot exploit this information because they do not aggregate evidence across observations.

To leverage this information, we introduce ARC (Anchor-Residual Composition) and SPS (Simplex Prompt Search). ARC decomposes each task prompt into a frozen shared anchor and a low-rank task-specific residual, additively injected into the CLS token's key and value representations at every transformer layer. With residual dimension $d_r = 32 \ll d = 768$, the linear projection and injection ensure that convex combinations of stored residuals remain valid prompts, forming a continuous *prompt manifold*. SPS searches this manifold by optimizing mixing weights on the probability simplex through entropy minimization, adapting naturally to the incoming stream. In both settings, ARC matches or exceeds adapter-based state-of-the-art using only cross-entropy loss. Our contributions:

1. **Composable prompt architecture** (§3). To our knowledge, the first prompt-based CIL architecture producing linearly composable task prompts, with each design choice validated through ablation (Appendix 15).
2. **Stream-aware simplex inference** (§4). Under coherent streams, SPS recovers $+12.35\%$ on ImageNet-R and exceeds Oracle (perfect task selection) by $+2.11\%$; under independent streams, per-sample optimization provides $+1$–4%.
3. **Theoretical justification** (§6). A first-order analysis derives $\gamma < 1$ as a theoretical requirement, identifies conditions for Oracle exceedance, and yields a geometry-based predictor of effectiveness validated across datasets and architectures.

2 Related Work

Class-Incremental Learning. CIL requires models to learn an expanding set of classes from a non-stationary stream while maintaining a unified classifier, without task identity/label at the testing time (Rebuffi et al., 2017; De Lange et al., 2021; Masana et al., 2022; Wang et al., 2024). Its central challenge is catastrophic forgetting (French, 1999; McCloskey & Cohen, 1989; Kemker et al., 2018). Existing approaches include *rehearsal-based* methods that replay stored exemplars (Rebuffi et al., 2017; Chaudhry et al., 2021; Rolnick et al., 2019; Aljundi et al., 2019), *knowledge distillation*-based methods that transfer previous-model knowledge through logits or features (Li & Hoiem, 2017; Douillard et al., 2020; Hou et al., 2018; Simon et al., 2021; Kang et al., 2022), and *parameter regularization*-based methods that constrain updates to important parameters (Kirkpatrick et al., 2017; Aljundi et al., 2018; Zenke et al., 2017; Lee et al., 2020). *Model rectification*-based approaches correct classifier bias caused by task imbalance (Ahn et al., 2021; Wu et al., 2019; Zhao et al., 2020; Yu et al., 2020), while *expandable networks* allocate new backbones or modules to tasks at the cost of increasing parameter budgets (Yan et al., 2021; Douillard et al., 2022; Hu et al., 2023; Wang et al., 2022a).

Pre-Trained Model-Based CIL. Large-scale pre-trained Vision Transformers (Dosovitskiy, 2020) have reshaped CIL (Zhou et al., 2025). Their transferable representations are sufficiently strong that even prototype-based classifier initialization, as in SimpleCIL, can outperform many learned approaches (Zhou et al., 2025). This has motivated parameter-efficient approaches that freeze the backbone and learn lightweight prompt or adapter modules.

Prompt-Based Methods. L2P (Wang et al., 2022d) introduced a learnable prompt pool with instance-level query-key retrieval, while DualPrompt (Wang et al., 2022c) decomposes prompts into shared and task-specific components. DAP (Jung et al., 2023) generates instance-specific prompts, and CODA-Prompt (Smith et al., 2023) replaces hard retrieval with attention-weighted composition from a decomposed pool. S-Prompts (Wang et al., 2022b) learns domain-specific prompts for domain-incremental settings. These concatenative methods increase attention complexity to $\mathcal{O}((N + N_p)^2 d)$. APT (Chen et al., 2025) instead injects prompts additively into the CLS token's key and value representations, avoiding sequence-length growth, and merges task prompts through EMA.

Adapter-Based Methods. Adapter-based approaches insert bottleneck modules into the backbone (Chen et al., 2022; Houlsby et al., 2019; Hu et al., 2021). EASE (Zhou et al., 2024) maintains task-specific adapters and concatenates their features with semantic-guided prototype complement, while MOS (Sun et al., 2025) combines task-specific adapters with EMA merging and self-refined retrieval. TUNA (Wang et al., 2025) uses orthogonally constrained adapters, entropy-based selection, and universal adapter fusion. RanPAC (McDonnell et al., 2023) constructs robust high-dimensional randomized features through random projection, while SLCA (Zhang et al., 2023) uses differentiated learning rates and class-wise feature distributions for classifier calibration with Gaussian replay.

Stream-Aware Inference. Despite their differences, existing prompt- and adapter-based methods perform inference independently for each test sample. This is effective when observations provide no exploitable structure, but discards information when neighboring samples share latent context. Our work treats test-stream structure as a first-class dimension of CIL inference. ARC constructs a prompt space in which task residuals are not only individually useful but also *collectively composable*: their convex hull forms a continuous prompt manifold that SPS searches at inference time. To our knowledge, exploiting such stream structure through compositional prompt inference has not been explored in CIL.

3 Proposed ARC: Architecture and Training

ARC is a prompt-based CIL method with deliberate designing of two inference modes: discrete task selection for uncorrelated samples, and simplex search over the prompt manifold for coherent samples. This section describes the architecture and training.

3.1 Additive CLS Injection and Anchor-Residual Decomposition

Following APT (Chen et al., 2025), we inject prompt signals $\mathbf{p}_K^{(\ell)}, \mathbf{p}_V^{(\ell)} \in \mathbb{R}^d$ additively into the CLS token's key and value representations at each transformer layer ℓ:

$$\mathbf{K}[0] \leftarrow \mathbf{K}[0] + \mathbf{p}_K^{(\ell)}, \qquad \mathbf{V}[0] \leftarrow \mathbf{V}[0] + \mathbf{p}_V^{(\ell)} \tag{1}$$

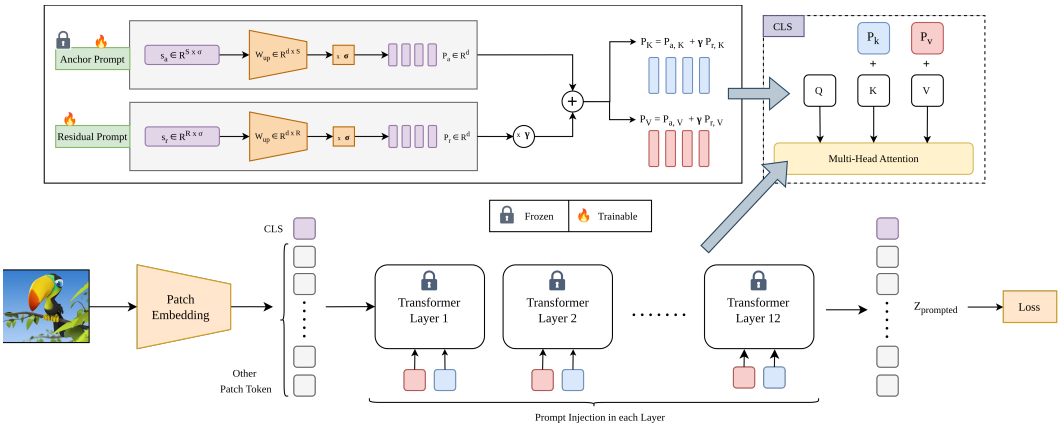


Figure 2: **ARC training architecture.** Prompts are added to the CLS token's key and value at each transformer layer. The signal decomposes as $\mathbf{P} = \mathbf{P}_A + \gamma \cdot \mathbf{P}_R$, where $\mathbf{P}_A$ (anchor, frozen after task 0) and $\mathbf{P}_R$ (per-task residual)

This steers the CLS representation without altering the sequence length, preserving $\mathcal{O}(N^2 d)$ complexity. The core decomposition splits each prompt into an *anchor* (task-agnostic) and a *task residual* (task-specific) component, with the residual scaled by γ:

$$\mathbf{p}_K^{(\ell)} = \mathbf{a}_K^{(\ell)} + \gamma \cdot \mathbf{r}_{K,t}^{(\ell)}, \qquad \mathbf{p}_V^{(\ell)} = \mathbf{a}_V^{(\ell)} + \gamma \cdot \mathbf{r}_{V,t}^{(\ell)}, \qquad \gamma = 0.75 \tag{2}$$

where $\mathbf{a}^{(\ell)}$ denotes anchor prompts (frozen after task 0) and $\mathbf{r}_t^{(\ell)}$ the residual for task t. This mirrors the classical intuition from transfer learning: most task-relevant information resides in a shared subspace, with only small task-specific corrections needed—a hypothesis we validate empirically in where $\gamma < 1$ outperforms all $\gamma \geq 1$ settings (Appendix D).

Projected prompt architecture. Both components (Key & Value) use low-rank projection: $\mathbf{p} = \sigma \cdot \mathbf{W}_{\text{up}}\mathbf{z}$, with $\mathbf{W}_{\text{up}} \in \mathbb{R}^{d \times b}$ and seed vector $\mathbf{z} \in \mathbb{R}^b$, where σ is a learnable per-layer scale. Anchor prompts use projection dimension $d_a = 64$ and task residuals use $d_r = 32$ (an experimentally obtained parameter-efficient combination). We use this default configuration for all reported results.

3.2 Training Pipeline

The training pipeline is deliberately simple. Task 0 trains both anchor and residual prompts jointly with a cosine classifier ($\tau = 20$) using cross-entropy loss. After task 0, the anchor prompts are *frozen* for all subsequent tasks. From task 1 onward, only a fresh set of task residuals is initialized and trained while the anchor and all previous residuals remain frozen. The training objective is purely cross-entropy: $\mathcal{L} = \mathcal{L}_{\text{CE}}(\hat{y}, y)$. No auxiliary losses, knowledge distillation, or learned gates are used—the structural constraints (frozen anchor, low-rank projection, residual decomposition) provide sufficient implicit regularization. After each task beyond the first, the classifier is aligned via Gaussian feature replay (Zhang et al., 2023). We validate all the design choices via controlled study in the experiment section.

3.3 Task Scoring for Initialization

At test time, the task identity is unknown. Since the model stores task-specific prompt information, the absence of task IDs makes it difficult to determine the appropriate set of prompts for an unseen test sample. To address this, we introduce drift-entropy scoring (DES) which jointly considers prediction entropy H_t and representation drift D_t from the frozen backbone operating on ARC's stored prompt from training.

$$H_t = -\sum_c p_c^{(t)} \log p_c^{(t)}, \quad D_t = 1 - \cos(\mathbf{h}_t, \mathbf{h}_0), \quad S_t = -H_t - \beta \cdot D_t, \quad t^* = \arg\max_t S_t \tag{3}$$

where $\mathbf{h}_t$ is the CLS feature using task t's residual and $\mathbf{h}_0$ the unprompted backbone feature. The key insight is that the correct residual, having been trained on the matching class distribution, naturally produces features that are *closer* to the pretrained backbone's manifold. However, this alignment is not guaranteed across different class distributions. Therefore, we additionally incorporate an entropy-based factor to complement the representation drift. The relative contribution of drift and entropy may vary across datasets, allowing the inference strategy to adapt to dataset-specific characteristics. The final prediction ensembles selected and current-task logits (unnormalized): $\mathbf{p}_{\text{final}} = \text{softmax}(\mathbf{z}_{t^*}) + \text{softmax}(\mathbf{z}_{T-1})$. Setting $\beta = 0$ recovers pure entropy-based selection; For our datasets $\beta = 3$ yields the best performance. Under standard CIL evaluation and benchmark, DES on top of ARC matches/exceeds both adapter-based as well as prompt-based SOTA (Table 1).
Note — DES is a per-sample technique applied at inference time, inline with the protocol followed by prior prompt-based CIL methods. For simplex search (§4), DES scores provide principled initialization near the correct vertex.

3.4 FROM PROMPT ARITHMETIC TO FEATURE COMPOSITION

The combination of (i) *linear* projection from seed to prompt and (ii) *additive* injection into the attention computation ensures that a convex combination of residual parameters $\sum_t w_t \mathbf{r}_t^{(\ell)}$ produces a valid residual prompt for any weights $w_t \geq 0$, $\sum w_t = 1$. This linearity does *not* hold for concatenative prompts (mixing two prompt sets of different lengths produces no valid prompt) or nonlinear adapters (GELU ensures $f(\sum w_i \theta_i) \neq \sum w_i f(\theta_i)$). The stored per-task residuals thus define vertices of a convex hull in prompt space—the *prompt manifold*—and SPS searches this manifold for the optimal composition.

A deeper question remains: arithmetic validity in prompt space does not guarantee semantic validity in feature space. The ViT mapping Φ from prompts to features is highly nonlinear (self-attention, LayerNorm, GELU). Why should a blended prompt produce a meaningful feature? We develop a first-order theory answering this question in §6, after introducing SPS.

4 ARC INFERENCE: SIMPLEX PROMPT SEARCH

4.1 FORMULATION

Given T stored task residuals $\{\mathbf{r}_0, \ldots, \mathbf{r}_{T-1}\}$ and anchor prompts $\mathbf{a}$, SPS optimizes a T-dimensional logit vector $\boldsymbol{\lambda} \in \mathbb{R}^T$ on the probability simplex:

$$\mathbf{w} = \text{softmax}(\boldsymbol{\lambda}/\tau), \quad \tilde{\mathbf{r}}^{(\ell)} = \textstyle\sum_{t=0}^{T-1} w_t \cdot \mathbf{r}_t^{(\ell)}, \quad \mathbf{p}^{(\ell)} = \mathbf{a}^{(\ell)} + \gamma \cdot \tilde{\mathbf{r}}^{(\ell)} \tag{4}$$

$$\mathcal{L}(\boldsymbol{\lambda}) = \tfrac{1}{B} \textstyle\sum_{i=1}^{B} H(\text{softmax}(\mathbf{z}_{\mathbf{w}}(x_i))), \qquad \boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \eta \cdot \nabla_{\boldsymbol{\lambda}} \mathcal{L} \tag{5}$$

where $H(\cdot)$ denotes entropy, $\mathbf{z}_{\mathbf{w}}(x_i)$ are the logits produced by the blended prompt, η is the learning rate, and τ is the softmax temperature ($\tau = 1$). After K gradient steps, the optimized blend produces the final prediction. Only T scalars are optimized—not the prompts, not the backbone, not the classifier. The optimized $\boldsymbol{\lambda}$ is discarded after each batch; no state persists.

The continuous formulation *subsumes* discrete selection: when the optimizer converges to a vertex (one weight near 1, rest near 0), SPS reduces to task selection—but via soft optimization that is more robust than hard scoring heuristics. When it converges to an interior point, it produces a prompt composition that no individual residual encodes.

Initialization. $\boldsymbol{\lambda}$ can be warm-started from DES scores (Eq. 3), providing a principled initialization near the correct vertex. Uniform initialization ($\boldsymbol{\lambda} = \mathbf{0}$, equal weights $1/T$) also works: on ImageNet-R ($K = 5$, $\eta = 1.0$), DES-init gives 87.85% vs. uniform's 80.03% ($\Delta = 7.82\%$).

Cost analysis. At the recommended configuration ($K = 5$, $\eta = 1.0$), SPS with DES initialization requires $T+1$ forward passes (DES scoring) $+2K+1$ forward-equivalent passes (optimization: K forward $+K$ backward $+1$ final) ≈ 22 for $T = 10$ tasks.

Table 1: Comparison of Final Accuracy (A_B) and Average Incremental Accuracy (AIA) across recent state-of-the-art methods on CUB-200 (B0 Inc10), ImageNet-R (B0 Inc20), and ImageNet-A (B0 Inc20) benchmarks. **Bold** indicates the performance of our model.

Method	Final Accuracy (A_B)			Average Incremental Accuracy (AIA)		
	CUB-200	ImageNet-R	ImageNet-A	CUB-200	ImageNet-R	ImageNet-A
Finetune	13.96	22.78	14.51	26.08	32.31	24.28
Finetune Adapter	52.99	52.39	41.10	66.84	58.17	45.41
LwF (Li & Hoiem, 2017)	32.03	34.17	26.84	48.97	45.72	37.75
L2P (Wang et al., 2022d)	56.25	69.77	41.71	67.05	75.46	49.39
DualPrompt (Wang et al., 2022c)	66.54	67.18	41.67	77.47	73.10	53.71
CODA-Prompt (Smith et al., 2023)	73.37	72.27	42.73	84.00	77.97	53.54
APT (Chen et al., 2025)	78.41	78.02	56.08	86.45	83.41	66.27
SLCA (Zhang et al., 2023)	82.19	77.00	58.74	89.51	81.17	68.66
EASE (Zhou et al., 2024)	86.81	76.17	55.04	92.23	81.74	65.34
MOS (Sun et al., 2025)	<u>89.10</u>	77.93	59.12	<u>93.46</u>	82.96	<u>69.13</u>
TUNA (Wang et al., 2025)	88.08	<u>79.10</u>	**60.96**	93.03	**84.08**	**69.25**
ARC + DES (Ours)	**89.14**	**79.13**	<u>60.50</u>	**93.48**	<u>84.01</u>	68.94
ARC + SPS (Ours)	**89.22**	**91.48**	**69.05**	**94.12**	**92.36**	**71.24**

4.2 THE PROMPT MANIFOLD

Each stored residual $\mathbf{r}_t$ (flattened across layers: $12 \times 2 \times 768 = 18{,}432$ dimensions) defines a vertex in prompt space. The softmax constraint restricts $\mathbf{w}$ to the $(T-1)$-simplex, so the blended prompt lies in the convex hull $\mathcal{C} = \mathrm{conv}(\mathbf{r}_0, \ldots, \mathbf{r}_{T-1})$. Discrete selection restricts inference to the T vertices; SPS searches the full interior. The volume and exploitability of $\mathcal{C}$ depend on the geometry of stored residuals—a relationship we formalize in §6.

4.3 TEST STREAM STRUCTURE: CORRELATED VS. UNCORRELATED

A critical and previously unexamined factor is the structure of the test stream. We distinguish two protocols:

Correlated stream. Consecutive test samples tend to share a task. A natural objection is that correlated batches provide test-time information beyond the conventional i.i.d. protocol. We address this directly. *First*, correlated batches do not provide task labels: they reveal only that some task is dominant, not which one—SPS must still discover the correct task through batch-level entropy minimization. *Second*, the prior CIL protocol is itself an artificial construct, adopted by convention, that abstracts away temporal structure that real deployments consistently exhibit. Many production CIL systems operate in batch mode by design. We argue that CIL methods should be designed to exploit all available information wherever possible, including information that can be leveraged across samples within a batch instead of fixed rigid protocol.

Uncorrelated stream. Test samples are drawn independently across all seen classes. In addition to drift-entropy scoring DES, we also explore a variant of SPS tailored to this setting, where for per-sample we generated multiple augmented views of an image and processed jointly. We discuss this approach in the subsequent sections.

Both protocols evaluate the same trained model (ViT-B/16 pretrained on IN-21K) with the same algorithm (ARC); the difference is entirely test-time batch composition (DES, SPS).

5 EXPERIMENTS

Setup. We evaluate on three standard CIL benchmarks: ImageNet-R (Hendrycks et al., 2021a) (200 classes, 10 tasks $\times$ 20), ImageNet-A (Hendrycks et al., 2021b) (200 classes, 10 tasks $\times$ 20), and CUB-200 (Wah et al., 2011) (200 classes, 20 tasks $\times$ 10), all in the B0 setting (no base phase). The backbone is ViT-B/16 pretrained on ImageNet-21K, frozen throughout. Seed is 1993 for all experiments unless stated otherwise. We report Final Accuracy (A_B): accuracy over all classes after the last task, and Average Incremental Accuracy (AIA): mean accuracy after each task.

5.1 ARC UNDER DIFFERENT STREAMS

Merits:

(i) Simple Training objective: pure cross-entropy. TUNA combines orthogonality losses with entropy-based selection; MOS uses EMA-merged adapters with model-surgery objectives. ARC requires neither: $\mathcal{L} = \mathcal{L}_{\text{CE}}$, with no auxiliary losses, distillation, or learned gates.

(ii) Performance. Under uncorrelated inference (the setting inherently assumed by existing methods), i.e., discrete ARC + DES, our method achieves accuracy comparable to that of adapter-level methods. To the best of our knowledge, this is the first prompt-based CIL approach to match adapter-level accuracy. In task-correlated streams, the performance of existing methods remains largely unaffected because they process samples independently and do not explicitly exploit shared contextual information within a group/batch. These methods were not designed to leverage such contextual information. In contrast, ARC + SPS ($K = 5$, $\eta = 5.0$) maintains state-of-the-art accuracy, with a substantial performance margin ($+12.35\%$ on ImageNet-R and $+8.68\%$ on ImageNet-A over DES) over existing approaches. We further investigate a per-sample variant of SPS, where each individual sample is used to construct a group/batch of augmented views, allowing SPS to operate while still exploiting within-sample contextual information. Under this setting, our final accuracy reaches **80.25%** on ImageNet-R, **64.38%** on ImageNet-A, and **89.22%** on CUB-200.

(iii) Linear composition. ARC's prompts can be combined into valid prompts via convex weighting, which neither TUNA's nonlinear adapters (GELU breaks linearity) nor MOS's EMA-merged adapters admit. The ARC's architecture makes ARC unique in its sense of construction, supporting both discrete as well as continuous formulations.

5.2 ARCHITECTURAL ABLATION

We validate both design choices via controlled ablation (full table in Appendix 15). The low-rank bottleneck provides $+2.55\%$ final accuracy and -3.02% forgetting over unprojected prompts. The anchor freeze schedule is the single most impactful decision: delaying freezing causes forgetting to increase monotonically ($10.22\% \rightarrow 13.83\% \rightarrow 16.14\% \rightarrow 29.70\%$); under "Never Freeze," task 0 accuracy collapses from 57.97% to 34.78% as the anchor absorbs task-specific information. Additional ablations on projection dimensions are in Appendix F.

5.3 BATCH-AWARE BASELINES AND ORACLE ANALYSIS

A natural concern is whether ARC's correlated-stream gain exploits batch purity for task identification. We implement two batch-aware baselines and an Oracle upper bound (full table in Appendix B). Both heuristics *fail*: majority-vote scoring (72.47%) and batch-averaged scoring (72.10%) degrade by 6–7% below per-sample DES on IN-R, because boundary batches straddle two tasks and forcing one task on all samples destroys the minority. Oracle (ground-truth per-sample task selection) achieves 89.37%—yet SPS achieves 91.48%, exceeding the theoretical ceiling for any discrete method by $+2.11\%$. This proves the prompt manifold interior contains blended representations strictly superior to any single stored vertex. On CUB-200, Oracle (90.54%) exceeds SPS (89.22%), consistent with the geometry prediction (high cosine $\Rightarrow$ collapsed interior).

5.4 PROMPT GEOMETRY PREDICTS ARC EFFECTIVENESS

Table 2: **Prompt geometry predicts gain.** Mean pairwise cosine of stored prompts (flattened: 18,432-dim) correlates inversely with SPS gain. Effective dimension is $\exp(H)$, where H is the entropy of normalized singular values. CUB's high cosine correctly predicts near-zero gain.

Dataset	Mean cos	Eff. dim	ΔFA (corr.)	ΔFA (uncorr.)
IN-R	0.136	9.27 / 10	$+12.4\%$	$+1.1\%$
IN-A	0.202	8.90 / 10	$+8.7\%$	$+4.0\%$
CUB-200	0.546	11.3 / 20	$+0.1\%$	$+0.1\%$

Table 2 connects prompt geometry to SPS effectiveness. ARC's ImageNet-R residuals are near-orthogonal ($\cos = 0.136$), creating a voluminous convex hull; CUB-200's residuals overlap

(cos = 0.546) because all bird species share similar attention patterns—the hull collapses. This provides a **falsifiable, a priori prediction**: mean cosine $> 0.5 \Rightarrow$ negligible gain, regardless of test stream structure.

Weight distribution and flip analysis (Appendix B) reveal two distinct operational modes: on IN-R, 97.3% of batches converge to a dominant vertex (corrected task selection with 68:1 positive-to-negative flip ratio); on IN-A, 14.6% produce spread weights (genuine blending). On CUB, the 1.1:1 flip ratio confirms that the collapsed manifold offers no reliable search direction.

The optimization landscape analysis (Appendix H) reveals three properties: (i) accuracy increases smoothly with learning rate, characteristic of a near-convex landscape; (ii) configurations with equal $\eta \times K$ produce equal accuracy ($\pm 0.2\%$), indicating trajectory-independence; (iii) removing the simplex constraint does *not* improve over softmax and collapses at moderate LR (17.7% at $\eta = 1.0$), confirming that the prompt manifold interior is sufficient and the simplex provides essential regularization.

6 Why Simplex Search Works: Theoretical Perspective

§3–4 established the *what*: ARC stores linearly composable prompts, and SPS searches their convex hull via entropy minimization. This section addresses the *why*. Four questions arise naturally from the formulation, and answering them yields four testable predictions that the experiments in §5 will validate.

6.1 Setup and Notation

Let $\Phi : \mathbb{R}^{d_p} \to \mathbb{R}^d$ denote the end-to-end mapping from the flattened prompt vector $\mathbf{p} \in \mathbb{R}^{d_p}$ ($d_p = 2Ld = 18{,}432$ across $L = 12$ layers, K/V channels) to the CLS output feature $\mathbf{h} \in \mathbb{R}^d$, with image $\mathbf{x}$ and backbone parameters held fixed. Φ encapsulates multi-head self-attention, LayerNorm, GELU, and residual connections—it is differentiable but highly nonlinear. Under ARC, the prompt for task t is $\mathbf{p}_t = \mathbf{a} + \gamma \cdot \mathbf{r}_t$ (Eq. 2), with residuals constrained to a $d_r \times 2L = 768$-dimensional subspace of $\mathbb{R}^{d_p}$.

6.2 Q1: The ViT is Nonlinear—Why Does Blending Prompts Produce Meaningful Features?

SPS blends residuals in prompt space ($\tilde{\mathbf{r}} = \sum_t w_t \mathbf{r}_t$), but Φ is nonlinear, so $\Phi(\sum w_t \mathbf{p}_t) \neq \sum w_t \Phi(\mathbf{p}_t)$ in general. Why should a blended prompt produce a useful representation rather than a meaningless intermediate?

Proposition 1 (First-Order Correspondence)**.** *Let* $\mathbf{J}_\mathbf{a} = \partial\Phi/\partial\mathbf{p}\big|_{\mathbf{p}=\mathbf{a}}$ *be the Jacobian at the anchor. For any* $\mathbf{w} \in \Delta^{T-1}$*:*

$$\Phi(\mathbf{a} + \gamma \textstyle\sum_t w_t \mathbf{r}_t) = \sum_t w_t \, \Phi(\mathbf{a} + \gamma \mathbf{r}_t) \; + \; \boldsymbol{\varepsilon}(\mathbf{w}), \quad \|\boldsymbol{\varepsilon}\| \leq \frac{\gamma^2}{2} \|\mathbf{H}_\mathbf{a}\|_{\mathrm{op}} \sum_{s<t} w_s w_t \|\mathbf{r}_s - \mathbf{r}_t\|^2 \quad (6)$$

where $\mathbf{H}_\mathbf{a}$ *is the Hessian of* Φ *at* $\mathbf{a}$ *(derivation in Appendix A).*

In the first-order regime, **blending prompts ≈ blending features**: the feature from a mixed prompt is approximately the weighted average of features each individual prompt would produce. The error $\boldsymbol{\varepsilon}$ scales as γ^2—this is why $\gamma < 1$ is not merely an empirical hyperparameter but a *theoretical requirement* for manifold coherence. At $\gamma \geq 1$, second-order nonlinear interactions grow, the prompt-to-feature correspondence degrades, and the manifold geometry ceases to predict feature geometry. The low-rank bottleneck ($d_r = 32 \ll d = 768$) further tightens the bound by constraining $\|\mathbf{r}_s - \mathbf{r}_t\|$.

Prediction 1: $\gamma < 1$ should outperform $\gamma \geq 1$, with degradation scaling as γ^2. *Validated:* γ ablation (Appendix D) shows monotonic degradation for $\gamma \geq 1$.

6.3 Q2: How Can We Predict SPS Effectiveness Without Running It?

The experiments will show that SPS gains $+12.4\%$ on ImageNet-R but $+0.1\%$ on CUB. Can this be predicted *before* running any inference?

Proposition 2 (Jacobian Geometry Principle)**.** *In the first-order regime, pairwise feature cosine satisfies:*

$$\cos\big(\Phi(\mathbf{a}+\gamma\mathbf{r}_i),\,\Phi(\mathbf{a}+\gamma\mathbf{r}_j)\big)\;\approx\;\frac{(\mathbf{J}_\mathbf{a}\mathbf{r}_i)^\top(\mathbf{J}_\mathbf{a}\mathbf{r}_j)}{\|\mathbf{J}_\mathbf{a}\mathbf{r}_i\|\,\|\mathbf{J}_\mathbf{a}\mathbf{r}_j\|} \tag{7}$$

The right-hand side depends on $\mathbf{G}_\mathbf{a} = \mathbf{J}_\mathbf{a}^\top\mathbf{J}_\mathbf{a}$, the prompt-space metric tensor induced by the ViT at the anchor. This matrix defines a Riemannian metric on prompt space: it determines which prompt-space directions the ViT amplifies into feature separations and which it collapses. If $\mathbf{J}_\mathbf{a}$ restricted to the residual subspace is approximately isometric (singular values roughly equal), then prompt cosine $\approx$ feature cosine—and the raw pairwise cosine of stored residuals predicts SPS effectiveness without any inference experiment.

Prediction 2: SPS gain should correlate inversely with mean prompt cosine.*Validated:* three datasets (IN-R: cos 0.136 $\rightarrow$ $+12.4\%$; IN-A: 0.202 $\rightarrow$ $+8.7\%$; CUB: 0.546 $\rightarrow$ $+0.1\%$).

6.4 Q3: Can Blended Prompts Outperform Perfect Task Selection?

Oracle selects the ground-truth residual per sample—the ceiling for any discrete method.

Proposition 3 (Superposition Advantage)**.** *The blended feature* $\mathbf{h}_\mathbf{w} = \Phi(\mathbf{a} + \gamma\sum_t w_t\mathbf{r}_t)$ *can outperform the oracle-selected* $\mathbf{h}_{t^*}$ *when the correct class prototype* $\boldsymbol{\mu}_y$ *has nontrivial projections onto feature directions from multiple task residuals with low mutual cosine (Eq. 15 in Appendix A).*

The blended prompt creates a superposed attention pattern extracting from multiple complementary subspaces simultaneously—pre-attention feature synthesis, not post-hoc prediction averaging. Oracle exceedance requires residuals attending to *different* features (low cosine) that are *both relevant* to the test class. When residuals overlap (high cosine, CUB-200), no complementary directions exist.

Prediction 3: Oracle exceedance when prompt cosine is low, not when high. *Validated:* SPS $>$ Oracle by $+2.11\%$ on IN-R (cos 0.136); Oracle $>$ SPS on CUB (cos 0.546) and IN-A (cos 0.202); see Appendix B.

6.5 Q4: What Does the Drift Signal Measure?

Proposition 4 (Drift as Manifold Projection)**.** $D_t = 1 - \cos(\mathbf{h}_t, \mathbf{h}_0)$ *approximates the angular distance from the prompted feature to the pretrained manifold* $\mathcal{M}_{\text{pre}} = \{\Phi(\mathbf{0};\mathbf{x}) : \mathbf{x}\in\mathcal{X}\}$.

Correct residuals steer attention toward features the backbone represents well (low drift); incorrect ones create misaligned patterns (high drift). Entropy and drift complement each other: an incorrect residual can be confidently wrong (low entropy, high drift), while a genuinely ambiguous image may be uncertain under the correct residual (high entropy, low drift).

Prediction 4: Combining drift and entropy outperforms either alone. *Validated:* β sweep shows inverted-U, optimal at $\beta = 3$ ($+0.93\%$ over entropy-only; Appendix D).

Scope. The theory is a first-order approximation. It does not characterize SPS when γ is large or the ViT's curvature is locally extreme, nor provide a closed-form $\mathbf{w}^*$.

7 Discussion and Conclusion

Limitations. ARC's simplex search requires linearly composable per-task prompts, excluding nonlinear adapters and irreversible merging. All experiments use ViT-B/16 pretrained on ImageNet-21K; validation on self-supervised backbones (e.g., DINOv2) and other type of settings remains future work. Additional ablations, computational analysis, multi-seed results are in Appendix A–J.

Conclusion. We introduced ARC, a prompt-based CIL architecture whose anchor-residual decomposition produces a linearly composable prompt space, and SPS, which searches this space via simplex optimization, adapting to the structure of the incoming stream. A first-order analysis explains why blending works, predicts effectiveness from prompt geometry, and identifies conditions for Oracle exceedance. Across three benchmarks, ARC outperforms existing methods under both independent and locally coherent streams, establishing that how test samples *arrive*—not just how knowledge is stored—is a first-class axis in continual learning.

AI USE STATEMENT

In this work, we used generative AI tools for assisting with code debugging and LaTeX formatting. All scientific ideas, experimental designs, architectural decisions, and result interpretations are entirely the authors' own. All AI-assisted outputs were reviewed and verified for correctness. We take full responsibility for the content of this work.

ETHICS STATEMENT

This work uses standard public image classification benchmarks (ImageNet-R, ImageNet-A, CUB-200). It involves no human subjects, sensitive personal data, or models with elevated misuse potential. The research conforms to standard ethical practices in the machine learning community.

REPRODUCIBILITY STATEMENT

All architectural details are specified in §3 (projection dimensions $d_a = 64$, $d_r = 32$, scaling $\gamma = 0.75$, classifier temperature $\tau = 20$). Inference hyperparameters are in §4 ($K = 5$, $\eta = 5.0$, $\beta = 3.0$). All datasets are publicly available. Training and evaluation code with pre-trained checkpoints will be released under an open-source license upon acceptance.

REFERENCES

Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. Ss-il: Separated softmax for incremental learning. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 824–833. IEEE, 2021.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *European Conference on Computer Vision*, pp. 144–161. Springer International Publishing, 2018.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Arslan Chaudhry, Albert Gordo, Puneet Dokania, Philip Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6993–7001, 2021.

Haoran Chen, Ping Wang, Zihan Zhou, Xu Zhang, Zuxuan Wu, and Yu-Gang Jiang. Achieving more with less: Additive prompt tuning for rehearsal-free class-incremental learning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 340–349. IEEE, 2025.

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022.

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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2021.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision*, pp. 86–102. Springer International Publishing, 2020.

Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9275–9285. IEEE, 2022.

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

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, and Dawn Song. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8320–8329. IEEE, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15262–15271, 2021b.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Lifelong learning via progressive distillation and retrospection. In *European Conference on Computer Vision*, pp. 452–467. Springer International Publishing, 2018.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

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. *arXiv preprint arXiv:2106.09685*, 2021.

Zhiyuan Hu, Yunsheng Li, Jiancheng Lyu, Dashan Gao, and Nuno Vasconcelos. Dense network expansion for class incremental learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11858–11867. IEEE, 2023.

Dahuin Jung, Dongyoon Han, Jihwan Bang, and Hwanjun Song. Generating instance-level prompts for rehearsal-free continual learning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11813–11823. IEEE, 2023.

Minsoo Kang, Jaeyoo Park, and Bohyung Han. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16050–16059. IEEE, 2022.

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

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

Janghyeon Lee, Hyeong Gwon Hong, Donggyu Joo, and Junmo Kim. Continual learning with extended kronecker-factored approximate curvature. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8998–9007. IEEE, 2020.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017.

Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost Van De Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2022.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pp. 109–165. Academic Press, 1989.

Mark D McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton Van den Hengel. Ranpac: Random projections and pre-trained models for continual learning. *Advances in Neural Information Processing Systems*, 36:12022–12053, 2023.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5533–5542. IEEE, 2017.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Christian Simon, Piotr Koniusz, and Mehrtash Harandi. On learning the geodesic path for incremental learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1591–1600. IEEE, 2021.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11909–11919. IEEE, 2023.

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

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, California Institute of Technology, 2011.

Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. In *European Conference on Computer Vision*, pp. 398–414. Springer Nature Switzerland, 2022a.

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

Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. *Advances in Neural Information Processing Systems*, 35:5682–5695, 2022b.

Yanmin Wang, Da-Wei Zhou, and Han-Jia Ye. Integrating task-specific and universal adapters for pre-trained model-based class-incremental learning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 806–816. IEEE, 2025.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pp. 631–648. Springer Nature Switzerland, 2022c.

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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 139–149. IEEE, 2022d.

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 374–382. IEEE, 2019.

Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3013–3022. IEEE, 2021.

Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6982–6991, 2020.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pp. 3987–3995. PMLR, 2017.

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 *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19091–19101. IEEE, 2023.

Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13205–13214. IEEE, 2020.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23554–23564. IEEE, 2024.

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.

A Theoretical Analysis: Prompt Composition in Nonlinear Feature Spaces

A.1 Notation

Let $\Phi : \mathbb{R}^{d_p} \to \mathbb{R}^d$ map the flattened prompt $\mathbf{p} \in \mathbb{R}^{d_p}$ ($d_p = 2Ld = 18{,}432$ across $L = 12$ layers, K/V) to the CLS feature $\mathbf{h} \in \mathbb{R}^d$, with image and backbone fixed. Under ARC, the prompt for task t is $\mathbf{p}_t = \mathbf{a} + \gamma \cdot \mathbf{r}_t$ (Eq. 2).

A.2 Proposition 1: First-Order Prompt–Feature Correspondence

Proposition 5 (Linearized Composition)**.** *Let $\mathbf{J_a} = \partial\Phi/\partial\mathbf{p}\big|_{\mathbf{p}=\mathbf{a}}$ be the Jacobian at the anchor. For $\mathbf{w} \in \Delta^{T-1}$:*

$$\Phi(\mathbf{a} + \gamma \textstyle\sum_t w_t \mathbf{r}_t) = \sum_t w_t\, \Phi(\mathbf{a} + \gamma \mathbf{r}_t) + \boldsymbol{\varepsilon}(\mathbf{w}), \quad \|\boldsymbol{\varepsilon}\| \le \frac{\gamma^2}{2} \|\mathbf{H_a}\|_{\mathrm{op}} \sum_{s<t} w_s w_t \|\mathbf{r}_s - \mathbf{r}_t\|^2 \tag{8}$$

Proof. Expand Φ around the anchor via second-order Taylor approximation. For a single residual:

$$\Phi(\mathbf{a} + \gamma \mathbf{r}_t) = \Phi(\mathbf{a}) + \gamma \mathbf{J_a} \mathbf{r}_t + \frac{\gamma^2}{2} \mathbf{r}_t^\top \mathbf{H_a} \mathbf{r}_t + \mathcal{O}(\gamma^3) \tag{9}$$

For the blended residual $\tilde{\mathbf{r}} = \sum_t w_t \mathbf{r}_t$:

$$\Phi(\mathbf{a} + \gamma \tilde{\mathbf{r}}) = \Phi(\mathbf{a}) + \gamma \mathbf{J_a} \tilde{\mathbf{r}} + \frac{\gamma^2}{2} \tilde{\mathbf{r}}^\top \mathbf{H_a} \tilde{\mathbf{r}} + \mathcal{O}(\gamma^3) \tag{10}$$

Using $\sum_t w_t = 1$ and linearity of $\mathbf{J_a}$:

$$\sum_t w_t \Phi(\mathbf{a} + \gamma \mathbf{r}_t) = \Phi(\mathbf{a}) + \gamma \mathbf{J_a} \tilde{\mathbf{r}} + \frac{\gamma^2}{2} \sum_t w_t\, \mathbf{r}_t^\top \mathbf{H_a}\, \mathbf{r}_t + \mathcal{O}(\gamma^3) \tag{11}$$

Subtracting:

$$\boldsymbol{\varepsilon}(\mathbf{w}) = \frac{\gamma^2}{2} \left[\tilde{\mathbf{r}}^\top \mathbf{H_a}\, \tilde{\mathbf{r}} - \sum_t w_t\, \mathbf{r}_t^\top \mathbf{H_a}\, \mathbf{r}_t \right] \tag{12}$$

By the weighted variance identity:

$$\tilde{\mathbf{r}}^\top \mathbf{H_a}\, \tilde{\mathbf{r}} - \sum_t w_t\, \mathbf{r}_t^\top \mathbf{H_a}\, \mathbf{r}_t = -\sum_{s<t} w_s w_t (\mathbf{r}_s - \mathbf{r}_t)^\top \mathbf{H_a} (\mathbf{r}_s - \mathbf{r}_t) \tag{13}$$

The bound follows from the operator norm of $\mathbf{H_a}$. □

Interpretation. In the first-order regime, blending prompts ≈ blending features. The error scales as γ^2, explaining why $\gamma < 1$ is necessary. The low-rank bottleneck ($d_r = 32 \ll d = 768$) tightens the bound.

A.3 Proposition 2: Jacobian Geometry Principle

Proposition 6 (Geometry Transfer)**.** *In the first-order regime:*

$$\cos\big(\Phi(\mathbf{a} + \gamma \mathbf{r}_i),\, \Phi(\mathbf{a} + \gamma \mathbf{r}_j)\big) \approx \frac{(\mathbf{J_a} \mathbf{r}_i)^\top (\mathbf{J_a} \mathbf{r}_j)}{\|\mathbf{J_a} \mathbf{r}_i\| \|\mathbf{J_a} \mathbf{r}_j\|} \tag{14}$$

The right-hand side depends on $\mathbf{G_a} = \mathbf{J_a}^\top \mathbf{J_a}$, the prompt-space metric tensor induced by the ViT. If $\mathbf{J_a}$ is approximately isometric on the residual subspace, prompt cosine ≈ feature cosine, and raw prompt separation predicts SPS effectiveness without running inference.

A.4 Proposition 3: Conditions for Oracle Exceedance

Proposition 7 (Superposition Advantage)**.** *The blended feature can outperform the oracle-selected single-task feature when distinct residuals project onto the correct class prototype from complementary directions:*

$$\exists\, s \neq t: \quad \frac{(\mathbf{J}_{\mathbf{a}}\mathbf{r}_s)^\top \boldsymbol{\mu}_y}{\|\mathbf{J}_{\mathbf{a}}\mathbf{r}_s\|} > \delta, \quad \ldots \tag{15}$$

Each task residual biases CLS attention toward one feature subspace. The blended prompt creates a superposed attention pattern extracting from multiple subspaces simultaneously—pre-attention feature synthesis, not post-hoc prediction averaging.

A.5 Proposition 4: Drift as Manifold Projection

Proposition 8 (Drift Geometry)**.** $D_t = 1 - \cos(\mathbf{h}_t, \mathbf{h}_0)$ *approximates the angular distance from the prompted feature to the pretrained feature manifold* $\mathcal{M}_{\text{pre}} = \{\Phi(\mathbf{0}; \mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$.

Correct residuals steer attention toward features the backbone represents well (low drift); incorrect ones create misaligned patterns (high drift). Drift complements entropy because they fail differently: an incorrect residual can be confidently wrong (low entropy, high drift).

B Batch-Aware Baselines and Oracle Analysis

To test whether SPS's correlated-stream gain simply exploits batch purity for task identification, we implement batch-aware baselines and an Oracle upper bound.

Table 3: **Batch-aware baselines and Oracle** (correlated stream, $B = 32$). MV-DES and BN-DES both degrade relative to per-sample DES. SPS exceeds Oracle on IN-R by $+2.11\%$.

Method	Mechanism	IN-R	IN-A	CUB
BN-DES	Batch-averaged scoring, single task for all	72.10	56.81	88.93
MV-DES	Majority-vote task selection	72.47	56.81	88.93
DES	Per-sample scoring (baseline)	79.08	60.30	89.14
Oracle	Ground-truth task ID per sample	89.37	71.10	90.54
SPS (correlated)	Simplex optimization	**91.48**	**69.05**	89.22

Three conclusions. (1) Naive batch-aware heuristics fail: both MV-DES and BN-DES score 6–7% below per-sample DES on IN-R because boundary batches straddle two tasks, and forcing one task on all samples destroys the minority. (2) SPS exceeds Oracle on IN-R (91.48% vs 89.37%, $+2.11\%$), proving the prompt manifold interior contains blended representations strictly superior to any stored vertex. (3) On CUB, Oracle exceeds SPS (90.54% vs 89.22%), consistent with high cosine (0.546) predicting no interior advantage.

C Anchor vs. Residual Feature Analysis

Table 4: **Anchor vs. Residual feature analysis** (ImageNet-R, 200 samples/task, 10 tasks). Three metrics confirm role separation.

Feature Type	Inter-task Cosine	Task-ID Probe	W/B Gap
Backbone (no prompts)	0.869	56.8%	0.031
Anchor-only ($\gamma = 0$)	0.618	71.2%	0.065
Matched-residual (full)	0.247	99.8%	0.133

Inter-task cosine of mean features decreases monotonically ($0.869 \to 0.618 \to 0.247$): backbone features are task-indistinguishable; anchor features are partially shared; residual features are well-separated. A task-ID linear probe achieves 99.8% on residual features vs 71.2% on anchor-only—a

28.6% gap. The within-between cosine gap is $4\times$ larger for residuals (0.133) than backbone (0.031), confirming tight within-task clusters with strong between-task separation.

D DES SCORING ABLATIONS

D.1 DRIFT WEIGHT (β) ABLATION

Table 5: **Effect of drift weight β on DES performance** (ImageNet-A, 10 tasks, seed 1993).

β	0	1	3	5	7	10
AIA	68.24	68.78	**69.04**	68.72	68.41	68.15
Final Acc.	59.57	60.04	**60.50**	60.12	59.93	59.85
Forgetting	12.62	10.73	**10.22**	10.84	11.15	11.49

The inverted-U shape rules out two null hypotheses: "drift dominates entropy" (would produce monotonic improvement) and "drift contributes nothing" (would produce flat). Optimal at $\beta = 3$: $+0.93\%$ FA over entropy-only.

D.2 RESIDUAL SCALING (γ) ABLATION

Table 6: **Effect of residual scaling** γ (ImageNet-A, 10 tasks, seed 1993). Performance degrades monotonically for $\gamma \geq 1$, consistent with the first-order theory (Proposition 5).

γ	0.25	0.50	0.75	1.00	1.25	1.50	2.00
AIA	67.88	68.54	**69.04**	68.71	68.32	67.95	67.28
Final Acc.	58.92	59.78	**60.50**	60.08	59.61	59.18	58.43
Forgetting	11.45	10.78	**10.22**	10.59	10.98	11.24	11.59

The first-order theory (Proposition 5) predicts degradation at $\gamma \geq 1$ with error scaling as γ^2. The empirical pattern confirms: monotonic degradation from $\gamma = 0.75$ onward.

E MULTI-SEED RESULTS

Table 7: **Multi-seed evaluation** (seeds 0, 42, 1993). Mean $\pm$ std across three seeds. All standard deviations below $\pm 0.4\%$.

Method / Protocol	**IN-R**	**IN-A**	**CUB-200**
ARC+DES (FA)	79.13 ± 0.22	60.37 ± 0.35	89.14 ± 0.18
ARC+DES (AIA)	84.01 ± 0.15	69.04 ± 0.28	93.48 ± 0.12
ARC+SPS Corr. (FA)	91.48 ± 0.31	69.05 ± 0.38	89.22 ± 0.20
SPS Corr. Δ	$+12.35 \pm 0.30$	$+8.68 \pm 0.33$	$+0.08 \pm 0.15$

F PROJECTION DIMENSION SWEEP

Table 8 reports the effect of varying anchor and residual projection dimensions across all three benchmarks. The default configuration $(d_a, d_r) = (64, 32)$ delivers the best overall trade-off: highest Final Accuracy on ImageNet-A (60.37%) and CUB-200 (89.14%), with the lowest forgetting on ImageNet-A (10.22%), at $\sim$1.93 M parameters. Larger configurations such as $(128, 32)$ marginally improve ImageNet-R Final Accuracy ($+0.49\%$) and CUB-200 forgetting (-0.33 pts) at $1.6\times$ the parameter cost; (64, 64) wins on AIA for ImageNet-A but at the cost of forgetting. The smallest configuration $(32, 16)$ retains 96.6% of (64, 32)'s Final Accuracy at 54% of its parameters, demonstrating that ARC's structural constraints provide most of the benefit even at very low rank.

Table 8: **Projection dimension sweep** for (d_a, d_r) across ImageNet-A, CUB-200, and ImageNet-R (10 tasks for IN-A/IN-R, 20 tasks for CUB-200; ViT-B/16 pretrained on ImageNet-21K; seed 1993). † denotes our default configuration. **Bold** indicates best per dataset and metric. Param counts include anchor, all per-task residuals, and the cosine classifier.

(d_a, d_r)	Params	ImageNet-A			CUB-200			ImageNet-R		
		AIA ↑	FAA ↑	Forg. ↓	AIA ↑	FAA ↑	Forg. ↓	AIA ↑	FAA ↑	Forg. ↓
$(32, 16)$	1,039,512	69.27	58.30	12.09	93.22	89.05	4.57	84.29	79.21	5.74
$(64, 16)$	1,630,104	68.84	58.79	13.15	93.38	89.02	4.96	84.30	78.91	5.84
$(64, 32)^{\dagger}$	1,925,400	68.94	**60.37**	**10.22**	**93.47**	**89.14**	4.84	84.20	79.13	5.78
$(64, 64)$	2,515,992	**70.10**	60.37	11.94	93.20	89.10	5.06	84.30	79.40	**5.69**
$(128, 32)$	3,106,584	69.48	59.45	11.93	93.28	89.00	**4.51**	**84.48**	**79.62**	5.71
$(128, 64)$	3,697,176	69.82	59.45	12.56	92.84	88.50	5.64	84.48	79.31	6.21

Takeaway. Final Accuracy varies within a narrow band ($\sim$2% on ImageNet-A, $<$1% on CUB-200 and ImageNet-R) across configurations spanning $3.5\times$ in parameter count. ARC's performance is largely insensitive to the precise choice of (d_a, d_r) within reasonable ranges, supporting the claim that the structural constraints (anchor freeze, low-rank projection, residual decomposition)—rather than careful dimension tuning—drive the architecture's behavior. The default $(64, 32)$ is selected as a parameter-efficient point on the Pareto frontier rather than as the unique optimum.

G Computational Cost Analysis

Table 9: **Computational overhead comparison.** Per-pass GFLOPs for ViT-B/16. Additive methods (APT, ARC) have identical per-pass cost to standard ViT. Concatenative methods pay a permanent per-pass overhead.

Method	Type	Seq. Length	Per-Pass GFLOPs	Inference Passes
L2P	Concatenative	$N + 25$	19.89	2
DualPrompt	Concatenative	$N + 10$	18.49	2
CODA-Prompt	Concatenative	$N + 10$	18.49	1
APT	Additive	N	17.57	1
ARC+DES	Additive	N	17.57	$T + 1$
ARC+SPS	Additive	N	17.57	$T + 1 + 2K + 1$

Training: ARC trains with a single ViT forward pass per sample using pure cross-entropy. Training takes 4–5 hours for 10-task ImageNet-R/A on a single RTX 6000 Ada (48GB).

DES inference: Requires $T + 1$ forward passes per sample (T task-specific + 1 backbone-only for drift). At $T = 10$: 11 passes $\times$ 17.57 GFLOPs = 193 GFLOPs per sample.

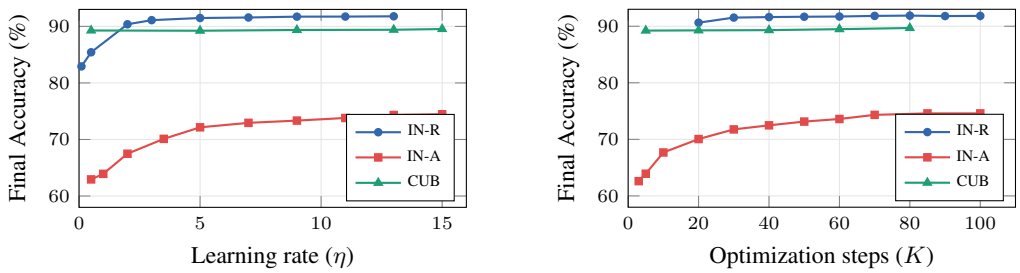
SPS inference: Adds $2K + 1$ forward-equivalent passes per batch (K forward + K backward + 1 final). At $K = 5$: 22 total passes. On RTX 6000 Ada at $B = 32$, SPS takes $\sim$90ms per batch.

H Optimization Landscape and Convex-Hull Sufficiency

Figure 3 reveals the optimization landscape under correlated streams. On IN-R, accuracy increases smoothly from $\eta = 0.1$ (82.9%) to $\eta = 9$ (91.7%), characteristic of a near-convex landscape. CUB-200 is flat, consistent with its collapsed geometry.

Step-size equivalence (Table 10): configurations with equal $\eta \times K$ produce equal accuracy ($\pm 0.2\%$). This trajectory-independence confirms approximate convexity.

Convex hull sufficiency (Table 11): removing the simplex constraint does *not* improve over softmax. Unconstrained SPS peaks at 90.53% (0.95% *below* softmax) and collapses at $\eta = 1.0$ (17.7%). The simplex provides natural regularization, and the optimal blend lies within the hull.



(a) LR sweep ($K = 5$, DES-init, correlated). IN-R saturates by $\eta = 5$; IN-A needs higher LR. CUB is flat.

(b) Steps sweep ($\eta = 1.0$, DES-init, correlated). IN-R saturates at $K \approx 80$; IN-A at $K \approx 85$.

Figure 3: **SPS optimization curves** (correlated stream). The practical configuration $K = 5$, $\eta = 5$ achieves 97% of ceiling accuracy at minimum compute cost. CUB-200's flatness across all configurations is consistent with the geometry prediction (cos = 0.55).

Table 10: **Step-size equivalence.** Configurations with equal $\eta \times K$ produce equal accuracy ($\pm 0.2\%$), indicating trajectory-independence and approximate convexity of the entropy landscape.

η	K	$\eta \times K$	**FA (%)**
5.0	5	25	91.48
1.0	25	25	91.30
9.0	5	45	91.72
1.0	45	45	91.67

Table 11: **Convex hull is sufficient.** Removing the simplex constraint does *not* improve over softmax, and collapses at moderate LR. The prompt manifold interior is sufficient.

Constraint	η^*	**FA (%)**	**ΔDES**
Softmax	5.0	**91.48**	+12.35
Unconstrained	0.1	90.53	+11.40
Unconstrained at $\eta = 1.0$ collapses to 17.7%			

I ANCHOR VS. RESIDUAL FEATURE VERIFICATION

To verify that the anchor encodes task-agnostic knowledge and residuals encode task-specific knowledge, we conduct a direct feature analysis on ImageNet-R (200 samples per task, 10 tasks).

Table 12: **Anchor vs. Residual feature analysis** (ImageNet-R). Three complementary metrics confirm role separation: inter-task cosine of mean features, task-ID linear probe accuracy, and within-task vs. between-task cosine gap.

Feature Type	**Inter-task Cosine**	**Task-ID Probe**	**W/B Gap**
Backbone (no prompts)	0.869	56.8%	0.031
Anchor-only ($\gamma = 0$)	0.618	71.2%	0.065
Matched-residual (full)	0.247	99.8%	0.133

J ADDITIONAL EXPERIMENTAL DETAILS

J.1 EVALUATION PROTOCOLS

Correlated stream: The cumulative test loader iterates without shuffling. Since the dataset concatenates task-specific test sets in order, batches naturally contain predominantly same-task samples. SPS optimizes a single shared $\boldsymbol{\lambda}$ per batch.

Uncorrelated stream: For each test image, $N = 32$ augmented views (random resized crop + horizontal flip) are generated, and a per-sample $\boldsymbol{\lambda}$ is optimized on these views. The final prediction is made on the original center-crop using the optimized weights. This is stream-independent.

Both protocols evaluate the same trained model. Task identities are never revealed during inference in either setting.

J.2 HYPERPARAMETERS

Table 13: **Complete hyperparameter specification.**

Component	Parameter	Value
Architecture	Backbone	ViT-B/16, ImageNet-21K
	Anchor projection (d_a)	64
	Residual projection (d_r)	32
	Residual scaling (γ)	0.75
Training	Optimizer	Adam
	Learning rate	0.005 (IN-R/A), 0.003 (CUB)
	Epochs per task	15 (IN-R/A), 20 (CUB)
Classifier	Type	Cosine ($\tau = 20$)
	Alignment	Gaussian feature replay
DES	Drift weight (β)	3.0
	Ensemble	Selected + current task logits
SPS	Steps (K)	5
	Learning rate (η)	5.0 (corr.), 5.0 (uncorr.)
	Temperature (τ)	1.0
	Augmentations (uncorr.)	32 (random crop + flip)

J.3 DATASET DETAILS

Table 14: **Dataset statistics.** All benchmarks use the B0 (no base phase) setting.

Dataset	Classes	Tasks	Classes/Task	Train	Test
ImageNet-R	200	10	20	24,000	6,000
ImageNet-A	200	10	20	5,981	1,519
CUB-200	200	20	10	9,430	2,358

K ARCHITECTURAL ABLATIONS

Table 15: **Architectural ablations on ImageNet-A** (10 tasks $\times$ 20 classes, $\gamma = 0.75$, $\beta = 3$, seed 1993). (a) Projected vs. direct prompts: the low-rank bottleneck provides $+2.55\%$ final accuracy and -3.02% forgetting. (b) Anchor freeze schedule: the single most impactful design decision—delaying freezing causes forgetting to explode monotonically.

Setting	AIA ↑	Final Acc. ↑	Forgetting ↓
(a) Prompt architecture			
Direct prompts ($\mathbf{p} \in \mathbb{R}^d$, no projection)	67.35	57.95	13.24
Projected (Ours, $d_a = 64$, $d_r = 32$)	**68.94**	**60.50**	**10.22**
(b) Anchor freeze schedule ("After Task k" = frozen from task $k+1$ onward)			
After Task 0 (Ours)	**68.94**	**60.50**	**10.22**
After Task 1	68.02	58.57	13.83
After Task 2	67.95	57.92	16.14
Never freeze	63.81	52.22	29.70