SELECTIVE BACKPROPAGATION FOR EFFICIENT FEW-SHOT CLASS-INCREMENTAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Few-Shot Class-Incremental Learning (FSCIL) requires models to continuously learn new classes from limited samples while retaining prior knowledge, under strict constraints on compute and memory. Existing approaches lie along a difficult trade-off: simple fine-tuning is computationally efficient but suffers from catastrophic forgetting, replay-based methods mitigate forgetting at the cost of substantial compute and memory, and exemplar-free methods often reduce forgetting by freezing most of the backbone, improving efficiency at the expense of adaptability. We propose Selective Backpropagation (SBP), a deterministic parameter budgeting framework that bridges this gap. SBP restricts gradient updates to a pre-allocated subset of network parameters, freezing past knowledge and preserving unbiased capacity for future learning, enabling rapid adaptation without costly mask optimization. We show that SBP achieves strong performance across standard FSCIL benchmarks while requiring training time close to that of naïve fine-tuning and substantially lower training time than prior SOTA methods. Crucially, our experiments expose a limitation of standard FSCIL evaluation: performance on short, distribution-consistent benchmarks does not necessarily predict behavior under distribution shift or over substantially longer learning horizons. We therefore evaluate FSCIL methods in cross-domain settings and over an 80-session ImageNet-1K stream. SBP remains strong across these regimes while maintaining low training cost, providing a favorable stability–plasticity–efficiency trade-off.

1 INTRODUCTION

Modern deep learning systems are increasingly deployed in streaming settings where data evolves over time, new categories appear, and models must be updated repeatedly, known as continual learning (CL) Parisi et al. (2019); Delange et al. (2021); Mai et al. (2022); Ramasesh et al. (2020). Within CL, common settings include Task-Incremental Learning (TIL) Kirkpatrick et al. (2017); Rusu et al. (2016), where task labels are known at test time, and Class-Incremental Learning (CIL) Yan et al. (2021); Hou et al. (2019), where task labels are unavailable at test time. Much of this literature assumes abundant labeled data when new classes arrive; an assumption that breaks in many real deployments where labeling is costly, and compute may be limited. We target this setting and study Few-Shot Class-Incremental Learning (FSCIL) Mazumder et al. (2021), where each session introduces a few novel classes with only a handful of labeled examples per class.

FSCIL is a challenging problem because it combines two competing requirements: the model must remain stable on previously learned classes, i.e. mitigate forgetting without replaying any old data, while being plastic enough to incorporate novel classes from extremely limited supervision without overfitting. This tension is amplified by an *accuracy–efficiency* trade-off: many methods boost accuracy by sacrificing efficiency (and vice-versa). Importantly, this dilemma cannot be assessed reliably only through accuracy on short, distribution-consistent classes. A method may preserve prior accuracy by freezing a representation that remains adequate when all classes come from the same dataset, but fail when later sessions require genuine representational adaptation. Similarly, performance over a small number of increments may not predict whether interference accumulates catastrophically over a much longer horizon. We therefore argue that standard FSCIL benchmarks are necessary for comparability but insufficient for evaluating continual learning quality: evalua-

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.

tion should jointly test retention, adaptation under distribution shift, behavior over longer learning horizons, and computational cost.

From a computational perspective, FSCIL methods can be broadly placed along a spectrum. At one extreme, standard fine-tuning (FT) represents the lower bound on compute: it is fast, simple, and fully adaptive, but catastrophically forgets previously learned classes. At the other extreme, replay-based or heavily regularized FSCIL methods mitigate forgetting by reintroducing stored data or auxiliary optimization stages, often incurring substantial memory and compute overhead Yoon et al. (2020); Chi et al. (2022); Zhou et al. (2022b). Between these extremes lie exemplar-free methods that reduce forgetting by freezing most of the backbone or restricting updates Zhou et al. (2022a); Peng et al. (2022), typically improving efficiency but sacrificing plasticity. A promising compromise is to control interference through capacity allocation: subnetwork and parameter-isolation methods restrict learning to subsets of parameters, preserving prior knowledge while still permitting adaptation Mallya & Lazebnik (2018); Mallya et al. (2018). However, most of these methods are developed for TIL, implicitly relying on task identity at test time, unavailable in FSCIL Tao et al. (2020); Kang et al. (2024). Furthermore, these methods incur additional costs to search for important network parameters and learn importance masks. An ideal FSCIL method would combine the speed and adaptability of fine-tuning with the stability of more complex continual learning approaches, without requiring replay, learned masks, or expensive meta-training. Despite extensive prior work, this regime remains largely unexplored: existing high-performing FSCIL methods almost universally trade efficiency or adaptability for stability.

In this work, we propose a novel FSCIL framework that balances stability and efficiency under strict exemplar-free constraints. We introduce selective backpropagation (SBP) which uses a three-way budgeting scheme that partitions the network parameters into: (i) *frozen* parameters that preserve previously learned knowledge, (ii) *assigned* parameters for learning the current task, and (iii) *free* parameters reserved for future tasks, randomly reinitialized every epoch. Crucially, SBP pre-allocates future capacity without letting current training implicitly shape it: free parameters are unallocated and receive no gradient updates, and are also reinitialized at the beginning of each epoch, so later sessions start from unbiased capacity and earlier tasks do not depend on parameters needed later. Unlike prior parameter masking methods, SBP's deterministic budget schedule eliminates mask-learning overhead and keeps per-session updates consistently cheap. Additionally, we couple SBP with prototype-based weight imprinting and classification, providing a stable and geometry-consistent initialization for novel-class weights without heavy optimization. Across standard FSCIL benchmarks, SBP achieves state-of-the-art performance on CIFAR-100 and CUB-200 while remaining competitive on MiniImageNet at substantially lower training cost than several competing baselines. Additionally, experiments on cross-domain and long-horizon FSCIL benchmarks show that methods relying on frozen backbones or rigid representations fail in settings with a stronger distribution shift and over long-horizon tasks. In contrast, SBP remains highly adaptive and outperforms competing approaches with improved training efficiency, demonstrating an exceptionally strong stability–plasticity–efficiency balance suitable for realistic FSCIL scenarios.

2 METHODOLOGY

Problem Statement: FSCIL considers a combination of few-shot learning, where supervision is extremely scarce, and CL, where models must update repeatedly under strict resource budgets. The training procedure is organized into a base session and a sequence of incremental sessions. In the base session, the model is trained on a large set of classes $\mathcal{C}_0$ with abundant labeled data. Each incremental session $t \in \{1, \ldots, T_N\}$ then introduces a small number of novel classes $\mathcal{C}_t$, with only a few K labeled samples $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^{K}$ per class. The label space is disjoint across sessions $\mathcal{C}_i \cap \mathcal{C}_j = \varnothing, \forall i \neq j$. During each session, only data from that session is accessible, and models must adapt under deployment-level constraints with limited training time and restricted hardware resources while mitigating forgetting of past knowledge.

2.1 SELECTIVE BACKPROPAGATION

In FSCIL, and CL in general, the standard practice of fine-tuning a shared backbone is inherently limited by the greedy nature of backpropagation (BP). Without constraints and experience replay, BP is computationally efficient but utilizes the entire network's plasticity to minimize the current loss,

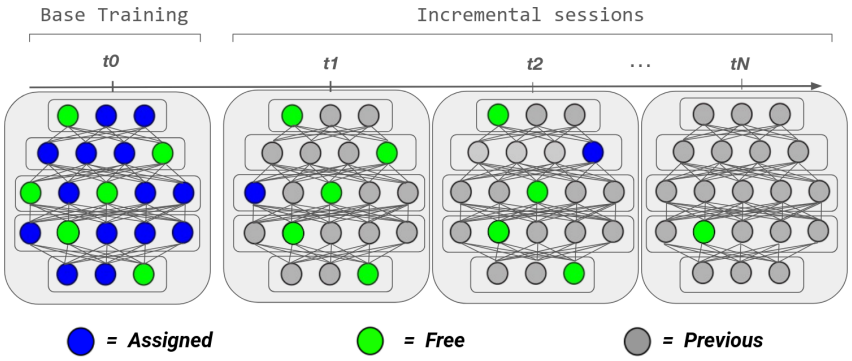


Figure 1: Selective Backpropagation (SBP) overview: parameter partitioning and masked gradient flow across incremental sessions.

leading to forgetting of prior knowledge and overfitting on current data. To address this, we propose Selective Backpropagation (SBP), an alternative to standard BP that can partially update the network to mitigate forgetting without additional compute overload. The key idea of SBP is preemptive and random parameter assignment for each new task in FSCIL, freezing parameters assigned for past tasks, and random reinitialization of free parameters kept for future tasks. This allows the model to acquire novel features while keeping most of the network unchanged, without spending any extra compute on post-training analysis for finding important parameters for previous tasks.

Figure 1 depicts an overview of the allocation mechanism in SBP during FSCIL. Let $\boldsymbol{\theta} = \{(\mathbf{W}_l, \mathbf{b}_l)\}_{l=1}^{L}$ denote the parameters of an L-layer neural network. At each incremental session, parameters in each layer l are partitioned into three mutually exclusive subsets,

$$\boldsymbol{\theta}^l = \boldsymbol{\theta}^l_{\text{assigned}} \cup \boldsymbol{\theta}^l_{\text{prev}} \cup \boldsymbol{\theta}^l_{\text{free}}. \tag{1}$$

Where, $\boldsymbol{\theta}^l_{\text{assigned}}$ denotes parameters optimized for the current task t, $\boldsymbol{\theta}^l_{\text{prev}}$ denotes parameters learned in previous tasks and kept frozen at increment t, and $\boldsymbol{\theta}^l_{\text{free}}$ denotes unallocated parameters reserved for future tasks. Note that parameter assignment is irreversible: at each session, the new assigned subset is sampled only from the remaining free parameters. Parameters assigned in previous sessions permanently enter θ_{prev} and are never eligible for reassignment or further optimization. Thus, capacity follows the one-way transition $\theta_{\text{free}} \rightarrow \theta_{\text{assigned}} \rightarrow \theta_{\text{prev}}$ across sessions.

During training, gradients are allowed to flow only through $\boldsymbol{\theta}^l_{\text{assigned}}$, whereas gradients corresponding to $\boldsymbol{\theta}^l_{\text{prev}}$ and $\boldsymbol{\theta}^l_{\text{free}}$ are explicitly blocked. The blocking mechanism is implemented through a set of binary masks, $\mathbf{M}^l_{\text{assigned}}$, $\mathbf{M}^l_{\text{prev}}$, and $\mathbf{M}^l_{\text{free}}$, which correspond to each parameter partition and are enforced using a backward hook on the gradient computation. No parameter mask is applied to the forward pass at inference; the prototype classifier only restricts distance computation to the currently active feature dimensions, as described in Appendix A.1. Thus, making it suitable for all CL settings (e.g., TIL, CIL, FSCIL). The parameter update for layer l at training step k is therefore given by:

$$\boldsymbol{\theta}^l_{k+1} \leftarrow \boldsymbol{\theta}^l_k - \eta \left(\nabla_{\boldsymbol{\theta}^l} \mathcal{L}_t \odot \mathbf{M}^l_{\text{assigned}} \right), \tag{2}$$

where η denotes the learning rate, $\mathcal{L}_t$ is the task-specific loss, and $\odot$ denotes the Hadamard (element-wise) product. SBP assigns a subset of network capacity ($\theta^l_{\text{assigned}}$) at each session using an allocation ratio β_t. For FSCIL, we allocate a larger fraction in the base session β_0 than in later increments to leverage the abundant base data, and use a constant schedule $\beta_t = \rho$ for $t \geq 1$, where $\rho \in (0, 1)$ (a hyperparameter). By restricting gradient updates to small assigned partitions (e.g. $\sim$1% of the network per incremental session), SBP naturally acts as a strong structural regularizer that restricts the model from overparameterizing and overfitting on the sparse K-shot data. For more balanced CL settings, the allocation can instead be scaled with task size (e.g., number of classes). Freezing $\boldsymbol{\theta}^l_{\text{prev}}$ during training prevents forgetting of past knowledge by preserving previously learned representations in the network. $\boldsymbol{\theta}^l_{\text{prev}}$ also allows transfer of prior knowledge for efficient learning of new tasks, without catastrophic forgetting.

2.2 FREE PARAMETER REINITIALIZATION

Unlike prior masking methods that are only suitable for TIL, a key novelty of SBP that makes it suitable for CIL and FSCIL is the random reinitialization mechanism of the free parameters $\boldsymbol{\theta}^l_{\text{free}}$ in each training epoch. This ensures that current tasks never become dependent on these parameters, and network performance for current tasks remains robust even when these parameters are updated for future tasks. A natural concern is whether free parameters, which are neither frozen nor explicitly masked at inference time, may inject noise into the learned representation. In SBP, free parameters are reinitialized at the beginning of every training epoch and receive no gradient updates. As a result, they are inherently non-stationary during optimization: the model never observes a consistent mapping through the free subspace across updates. This prevents the network from encoding task-relevant structure in these parameters and forces the optimizer to learn solutions that use previously allocated and currently assigned capacity while remaining invariant to the free partition. Consequently, although inference is deterministic and no forward masking is applied, the learned representation does not rely on free parameters in practice. Appendix H provides further theoretical reasoning for free parameter reinitialization, and Section 3.4 and Appendix I empirically demonstrate its effectiveness.

2.3 PROTOTYPE-BASED CLASSIFICATION

Given the limited number of samples during incremental rounds in FSCIL, we initialize classifier weights for new classes using weight imprinting Qi et al. (2018), which provides a deterministic initialization by directly aligning weights with each class centroid, computed as a mean of the feature vectors of the training set in each increment. Prototypes are frozen after each increment and are not updated in later tasks. Frozen prototypes, however, can become misaligned with a changing feature representation in the backbone De Lange & Tuytelaars (2021); Gomez-Villa et al. (2024). While SBP's parameter masking limits such changes, incremental optimization can still modify the unallocated portion of the representation, causing stored prototypes and current embeddings to differ along plastic dimensions. To reduce this mismatch, we use a partial-distance formulation that restricts prototype comparisons to feature dimensions allocated by SBP up to the current session. As new capacity is assigned, the active classification subspace expands accordingly, while dimensions reserved for future adaptation remain excluded. Additional details are provided in Appendix A.1.

During inference, we classify each sample using a nearest-class-mean classifier over the stored prototypes. Given a sample embedding $f_\theta(x)$ and class prototype p_c, we define the class logit as the negative distance between them:

$$z_c(x) = -d\left(f_\theta(x), p_c\right),$$

where $d(\cdot,\cdot)$ denotes the partial-distance function described in Appendix A.1. The prediction probability is then obtained by applying a softmax over the logits of all observed classes:

$$p(y = c \mid x) = \frac{\exp(z_c(x))}{\sum_{c'} \exp(z_{c'}(x))}.$$

3 EXPERIMENTS

We first evaluate our method on three well-established benchmark datasets for FSCIL: CIFAR-100 Krizhevsky et al. (2009), MiniImageNet Vinyals et al. (2016) and the Caltech–UCSD Birds-200-2011 (CUB-200) Wah et al. (2011) (in Appendix C). Dataset details are included in Appendix A. For CIFAR-100 and MiniImageNet, we follow the standard FSCIL protocol established in Tao et al. (2020). Specifically, the base session trains on 60 classes with full supervision. The remaining 40 classes are then learned incrementally across 8 sessions, each introducing 5 new classes under a 5-way 5-shot setting. Overall, the model learns all classes over 9 sequential sessions without revisiting old data. We report per-session mean-class accuracy and average incremental accuracy to measure performance, and per-session training time to measure computational efficiency. We also evaluate cross-domain transferability on MiniImageNet and CUB-200 datasets and long-horizon scalability on ImageNet-1K over 80 incremental sessions.

We compare against a broad set of recent FSCIL baselines (details in Appendix A) on all datasets. All baselines are reproduced under the same experimental settings and with the ResNet-18 back-

Table 1: CIFAR-100 FSCIL (5-way 5-shot) classification accuracy (%) with ResNet-18 using the TOPIC Cheraghian et al. (2021) class split. Results are over five runs with different random seeds. Best (per column) is in **bold**, second best is underlined. For efficiency, lower is better.

Method	Sessions (Accuracy %)										Efficiency	
	1	2	3	4	5	6	7	8	9	Avg	Time (m)	vs Ours
Entropy-Reg Liu et al. (2022)	74.37	71.31	68.93	66.61	64.43	62.30	60.29	58.40	56.60	64.80	486	34.71×
ALICE Peng et al. (2022)	79.00	70.50	67.10	63.40	61.20	59.20	58.10	56.30	54.10	63.21	196	14.00×
FACT Zhou et al. (2022a)	80.13	75.09	70.51	66.43	62.55	59.33	56.87	54.42	51.95	64.14	40	2.86×
WaRP Kim et al. (2023)	80.31	75.86	71.87	67.58	64.39	61.34	59.15	57.10	54.74	65.82	20	1.43×
NC-FSCIL Yang et al. (2023)	79.57	59.23	61.03	61.76	58.79	55.28	53.74	51.96	49.43	58.98	85	6.07×
OrCo Ahmed et al. (2024)	79.83	72.79	69.71	65.09	61.16	57.59	54.39	51.82	49.38	62.42	28	2.00×
Comp Zou et al. (2024)	79.17	69.11	63.61	59.01	55.22	52.09	50.01	48.27	45.67	58.02	89	6.36×
Approx Wang et al. (2024)	81.00	73.35	69.34	65.40	62.31	59.15	56.59	54.59	52.47	63.80	296	21.14×
ADBS Li et al. (2025a)	80.53	74.34	69.03	64.43	60.40	56.85	53.69	50.86	48.32	62.05	27	1.93×
SoftNet Kang et al. (2024)	79.97	75.75	71.76	67.36	64.09	60.91	59.07	56.94	54.76	65.62	67	4.79×
No Replay	81.42	51.40	23.04	14.68	2.86	1.92	1.34	1.05	1.00	19.86	11	0.79×
Partial Replay	81.05	72.83	66.76	61.19	56.91	52.20	49.88	46.61	44.99	59.16	11	0.79×
Full Replay	81.42	76.46	71.47	66.56	62.06	58.85	55.59	53.13	50.50	64.00	16	1.14×
Full Replay (Proj.)	84.00	78.50	73.30	68.80	64.60	61.20	58.20	55.40	53.00	66.33	11	0.79×
Ours	**82.70**	**78.40**	**74.40**	**70.90**	**67.50**	**64.90**	**62.00**	**59.90**	**57.90**	**68.73**	**14**	**1.00×**
± Std	0.20	0.40	0.40	0.40	0.40	0.50	0.40	0.30	0.30	0.37	–	–

bone as our method for a fair comparison. Results with additional backbone architectures for SBP are reported in Appendix F. We report accuracy as the mean over five independent runs using random seeds for all experiments. Complete details on implementation, hyperparameters, datasets, and baselines are provided in Appendix A.

3.1 Experiments on CIFAR-100 and MiniImageNet

Table 1 reports the per-increment accuracy, average incremental accuracy, and total training time of FSCIL methods, including ours, on the CIFAR-100 dataset. SBP attains 57.90% accuracy (1.30% higher than the second best) after the final session and 68.73% average incremental accuracy (2.91% higher than the second best), achieving state-of-the-art performance across all increments.

For efficiency, SBP trains in 14 minutes total across all sessions. Among the compared FSCIL methods, the next fastest is WaRP at 20 minutes, requiring 1.43× the training time of SBP. We note that the second-best method differs depending on the metric considered. For example, Entropy-Reg is the next best method in terms of final-session accuracy, but it requires 34.71× the training time of SBP. In contrast, WaRP is the second-best FSCIL method in terms of training time, but achieves 3.16% lower final-session accuracy.

To further demonstrate the effectiveness of our approach in balancing efficiency and performance, we also report results for fine-tuning (No Replay), Partial Replay (storing 5 exemplars per base and incremental class), and Full Replay baselines. As expected, naive fine-tuning without replay suffers from catastrophic forgetting, setting a lower bound on accuracy. However, it provides a strong efficiency baseline, requiring only 11 minutes of total training time. Storing a limited memory buffer of 5 samples per class partially mitigates this degradation, achieving a final accuracy of 44.99% and an average incremental accuracy of 59.16%, while requiring the same 11 minutes of training. FSCIL methods generally mitigate forgetting more effectively than these simple baselines, but often at the expense of substantially greater training time. In contrast, our exemplar-free method outperforms Partial Replay by 12.91% in final accuracy and 9.57% in average incremental accuracy, while requiring only 14 minutes of total training time. This shows that SBP achieves a favorable operating regime, providing substantially stronger stability and plasticity with only a small increase in training cost over simple replay.

Additionally, while Full Replay is treated as a strong reference baseline because it retains all historical training data, our results in Table 1 show that access to all previous samples does not necessarily yield the strongest performance in the FSCIL setting. Full Replay requires 16 minutes of total train-

Table 2: MiniImageNet FSCIL (5-way 5-shot) classification accuracy (%) and training efficiency with ResNet-18. Results are over five runs with different random seeds. Best (per column) is in **bold**, second best is underlined. For efficiency, lower is better.

Method	Sessions (Accuracy %)										Efficiency	
	1	2	3	4	5	6	7	8	9	Avg	Time (m)	vs Ours
Entropy-Reg Liu et al. (2022)	72.49	66.93	62.21	58.24	54.74	51.59	48.86	46.42	44.43	56.21	573	19.10×
ALICE Peng et al. (2022)	80.60	70.60	67.40	64.50	62.50	60.00	57.80	56.80	55.70	63.99	314	10.47×
FACT Zhou et al. (2022a)	70.45	64.73	60.58	57.44	54.32	51.60	49.00	47.00	45.46	55.62	41	1.37×
WaRP Kim et al. (2023)	68.63	64.43	60.44	57.82	55.42	52.64	50.02	48.19	46.89	56.05	**27**	0.90×
NC-FSCIL Yang et al. (2023)	76.45	68.12	63.50	59.28	57.10	54.22	51.80	49.65	47.90	58.67	883	29.43×
OrCo Ahmed et al. (2024)	**82.30**	72.60	68.94	65.57	62.70	60.11	58.08	56.58	54.96	64.65	157	5.23×
Comp Zou et al. (2024)	77.38	67.31	62.61	58.69	56.17	53.09	50.43	48.09	46.70	57.83	290	9.67×
Approx Wang et al. (2024)	38.27	24.69	22.93	21.83	20.68	19.64	18.64	18.14	17.58	22.49	91	3.03×
ADBS Li et al. (2025a)	81.65	**75.43**	**71.43**	67.99	**65.53**	61.97	**59.04**	**56.97**	**55.74**	**66.19**	191	6.37×
SoftNet Kang et al. (2024)	68.22	62.97	58.90	55.80	53.26	50.91	48.56	46.77	44.98	54.49	79	2.63×
Ours	78.10	73.80	70.20	**68.50**	65.40	**62.70**	58.20	56.30	54.10	65.26	30	1.00×
± Std	0.10	0.20	0.20	0.40	0.40	0.40	0.40	0.60	0.90	0.40	–	–

ing time and achieves a final-session accuracy of 50.50% and an average incremental accuracy of 64.00%. We also evaluate *Full Replay (Proj.)*, where the backbone is frozen after the base session and only a two-layer projection head is fine-tuned using full replay. This stronger baseline reaches 66.33% average and 53.00% final accuracy, but SBP still performs better without storing past data. Several FSCIL methods require substantially more training time while yielding only modest improvements over Full Replay, and some perform worse. In contrast, SBP outperforms Full Replay by 7.40% in final-session accuracy and 4.73% in average incremental accuracy, while also reducing total training time from 16 to 14 minutes. These results highlight the importance of evaluating FSCIL methods through an explicit stability–efficiency trade-off: if simple baselines already provide competitive performance at low computational cost, new FSCIL methods should justify their additional complexity through clear improvements in accuracy, efficiency, or both.

Table 2 shows the accuracy and training-time comparison of our method and FSCIL baselines on MiniImageNet. Our method remains competitive on this more challenging benchmark and achieves strong accuracy numbers in most sessions. SBP achieves 54.10% final-session accuracy and 65.26% average incremental accuracy, remaining highly competitive compared to ALICE at 55.70% final-session accuracy and OrCo at 64.65% average incremental accuracy. While WaRP requires less training time (27 minutes) than SBP (30 minutes), it achieves approximately 10% lower final and average incremental accuracy. Together with the CIFAR-100 results, these findings indicate that SBP provides a favorable stability–plasticity–efficiency trade-off. Finally, SBP maintains consistently strong performance across datasets and sessions, whereas the second-best method varies across benchmarks and metrics. This suggests that competing approaches may be more sensitive to the dataset or evaluation setting, while SBP provides more consistent performance across the evaluated FSCIL scenarios. Together, the CIFAR-100 and MiniImageNet results show that SBP provides a favorable accuracy–efficiency trade-off under the standard FSCIL protocol. However, these benchmarks contain only eight incremental sessions and draw base and novel classes from the same underlying distribution. They therefore provide limited evidence about whether a method remains plastic under distribution shift or stable over much longer learning sequences. We evaluate on these two settings next.

3.2 Cross-Domain FSCIL

Chen et al. (2019) showed that most few-shot learning methods fail in scenarios where base and novel classes are drawn from slightly different datasets. Therefore, we adapt the MiniImageNet-CUB cross-domain scenario reported in Chen et al. (2019) to FSCIL to investigate our method and FSCIL baselines on this challenging setup. In this setting, the base session is trained on 60 classes on MiniImageNet, and incremental sessions introduce novel classes from CUB-200 in the form of 20-way 5-shot tasks. All methods use the ResNet-18 backbone trained from scratch, and are trained for 100 epochs for the base session, and 25 epochs for the incremental sessions. We further validate cross-domain robustness on MiniImageNet→Caltech-101, with results in Appendix D.

Table 3: Cross-Domain FSCIL (MiniImageNet → CUB-200) classification accuracy (%) with an initial session of 60 base classes and 10 incremental sessions of 20-way 5-shot using a ResNet-18 backbone. Best (per column) is in **bold**, second best is underlined.

Method	Sessions (Accuracy %)											Avg (%)
	1	2	3	4	5	6	7	8	9	10	11	
ALICE Peng et al. (2022)	65.56	56.05	51.73	47.68	44.96	41.81	39.37	37.19	35.26	33.45	31.88	44.09
FACT Zhou et al. (2022a)	**79.20**	62.46	51.83	44.39	39.07	35.06	31.38	28.70	26.13	24.41	22.57	40.47
WaRP Kim et al. (2023)	69.41	64.13	59.33	54.81	51.37	48.23	45.54	43.06	41.09	38.39	36.12	50.13
NC-FSCIL Yang et al. (2023)	64.35	54.27	50.17	44.15	39.47	35.64	32.46	29.82	27.37	25.47	23.81	38.82
OrCo Ahmed et al. (2024)	58.05	43.95	39.89	34.38	30.35	29.62	28.33	26.50	24.25	23.24	22.07	32.78
Comp Zou et al. (2024)	59.97	55.48	50.98	47.56	44.37	41.52	39.05	36.74	34.73	32.88	31.19	43.13
SoftNet Kang et al. (2024)	67.63	60.98	54.94	50.46	46.03	42.74	39.90	37.25	34.88	32.79	31.08	45.33
Ours	77.33	**68.47**	**62.92**	**58.15**	**54.25**	**50.84**	**47.85**	**45.13**	**42.74**	**40.47**	**38.57**	**53.34**

Table 3 reports the results of the cross-domain FSCIL experiment. SBP achieves the best accuracy for all incremental sessions, with a final accuracy of 38.57% and an average accuracy of 53.34%, outperforming the next-best method, WaRP, by 2.45% and 3.21%, respectively. We also observe a substantial degradation for FSCIL methods that freeze the backbone after the base session, as their fixed representations cannot adapt when incremental classes come from a distribution different from the base classes. This highlights a key limitation of many efficient FSCIL methods: they reduce computation by sacrificing adaptability, making them sensitive to domain shifts. In realistic CL settings, where future data may differ semantically or statistically from the base distribution, such rigidity is undesirable. SBP avoids this failure mode by maintaining controlled plasticity across sessions, allowing the backbone to adapt while limiting catastrophic forgetting. More broadly, these results illustrate that strong retention on a distribution-consistent FSCIL benchmark does not necessarily imply an effective stability–plasticity trade-off. Restricting representation changes can preserve past performance while also limiting the model's ability to learn shifted future classes.

3.3 Long-Horizon FSCIL

Standard FSCIL benchmarks such as CIFAR-100 and MiniImageNet contain only eight incremental sessions, making it difficult to assess whether a method remains stable over substantially longer learning sequences. We therefore construct a long-horizon ImageNet-1K benchmark with 600 base classes followed by 400 novel classes introduced over 80 incremental sessions, each using a 5-way 5-shot protocol. All methods use a ResNet-18 backbone and are evaluated over all classes observed up to each session.

Table 4 shows that differences between methods become substantially more pronounced as the learning horizon increases. WaRP remains competitive during the early sequence and slightly outperforms SBP at Sessions 20 and 30, but its accuracy subsequently declines from 51.40% at Session 30 to 12.90% at Session 80. In contrast, SBP retains 39.08% accuracy after all 80 incremental sessions. ADBS and OrCo similarly reach only 14.50% and 18.40% at the final session.

These results show that performance over short FSCIL sequences does not necessarily predict long-horizon behavior. In particular, methods that appear competitive over the first few increments can accumulate substantial degradation as the sequence grows. This reinforces the need to evaluate FSCIL methods not only on short, distribution-consistent benchmarks, but also under longer learning horizons that more directly stress the stability–plasticity trade-off.

3.4 Ablation Study

To analyze the contribution of each component in our method, we conduct an ablation study on CIFAR-100 using the same experimental setting as in Table 1. Results are reported in Table 5. Our results show that SBP is the primary driver of performance. Removing SBP (*w/o SBP*) leaves base-session performance intact (84.10%) but causes immediate collapse once incremental learning begins, dropping to 13.90% in Session 2 and 4.30% by the final session. Average incremental accuracy falls from 68.73% to 15.46%, indicating severe interference from unrestricted parameter updates rather than insufficient base-session learning.

Table 4: Long-horizon FSCIL on ImageNet-1K. We report classification accuracy (%) at selected sessions throughout the incremental sequence. Best (per column) is in **bold**, second best is underlined.

Method	Sessions (Accuracy %)										
	0	**1**	**5**	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**
ADBS Li et al. (2025a)	61.50	57.10	47.20	39.20	27.40	22.20	18.80	17.20	16.00	14.90	14.50
OrCo Ahmed et al. (2024)	27.00	24.80	24.90	25.10	24.30	23.20	21.90	21.00	20.00	19.30	18.40
WaRP Kim et al. (2023)	63.20	61.90	60.40	58.60	**54.90**	**51.40**	47.80	42.10	33.30	20.00	12.90
Ours	**72.45**	**70.09**	**64.99**	**61.85**	54.26	50.96	**47.93**	**45.19**	**43.01**	**40.83**	**39.08**

Table 5: Ablation study on CIFAR-100 FSCIL, 5-way 5-shot task. We report per-session classification accuracy (%), average incremental accuracy, and the change in average accuracy relative to the full model (Δ). Best (per column) is in **bold**, second best is underlined.

	Session Accuracy (%)									Metrics	
Ablation	1	2	3	4	5	6	7	8	9	Avg (%)	Δ
Full	82.70	**78.40**	**74.40**	**70.90**	**67.50**	**64.90**	**62.00**	**59.90**	**57.90**	**68.73**	—
SBP Only	82.85	76.25	70.81	66.20	61.90	58.20	54.36	51.47	48.85	63.43	−5.30
w/o Imprinting	82.70	75.75	69.99	65.01	60.70	57.18	53.76	50.77	48.24	62.63	−6.10
w/o SBP	**84.10**	13.90	10.00	5.60	7.00	6.20	4.00	4.00	4.30	15.46	−53.27
w/o Free-Init	82.60	1.88	1.66	1.69	1.85	1.36	1.56	1.43	1.56	10.62	−58.11

In contrast, *SBP Only* (without a prototype-based classifier) retains most of the full model's performance, achieving 63.43% average incremental accuracy and 48.85% final-session accuracy, only 5.30 and 9.05 percentage points below the full model. Its gradual degradation indicates that selective parameter isolation is the main contributor to continual retention, while the remaining components mainly improve adaptation. Free-neuron re-initialization is also essential. Without it (*w/o Free-Init*), accuracy falls from 82.57% in the base session to 1.88% in Session 2 and 1.56% in the final session, yielding only 10.62% average accuracy. This suggests that unreinitialized free parameters can become implicitly co-adapted to the current session, so reassigning them later disrupts previously learned features. We further analyze this failure mode with t-SNE visualizations in Appendix I.

In line with prior few-shot learning research, prototype imprinting is a useful complementary component for FSCIL. Removing imprinting (*w/o Imprinting*) reduces average incremental accuracy from 68.73% to 62.63% and final-session accuracy from 57.90% to 48.24%. Unlike the collapses observed when SBP or free-neuron re-initialization is removed, performance without imprinting declines gradually across sessions. This suggests that imprinting does not primarily prevent catastrophic forgetting, but it is important in FSCIL, where each novel class is represented by only five training examples and reliable initialization of its classifier is difficult to obtain through gradient-based optimization alone. Additional ablations over backbone architectures and base-session budgets are provided in Appendices F and G.

4 Related Work

Few-Shot Class-Incremental Learning (FSCIL) lies at the intersection of few-shot learning and class-incremental learning (CIL), and inherits challenges from both. In CIL, models must learn a growing set of classes without access to task identity at test time, making them particularly vulnerable to catastrophic forgetting Rebuffi et al. (2017). FSCIL further amplifies this difficulty by introducing only a handful of labeled examples per incremental class Tao et al. (2020), which increases both overfitting on new classes and interference with previously learned ones Zhao et al. (2021). As a result, FSCIL methods must carefully balance stability, plasticity, and computational efficiency under strict data and memory constraints.

A broad class of FSCIL methods mitigates forgetting using replay, meta-learning, or heavily structured objectives. Meta-learning approaches aim to learn update rules or initializations that facilitate rapid adaptation to new classes Chi et al. (2022); Zhou et al. (2022a), while others improve base–incremental compatibility via pseudo-features, or data-free replay mechanisms Cheraghian et al. (2021); Liu et al. (2022). Related work also emphasizes prototype separability and alignment, using prototype evolution, feature regularization, or geometry-aware losses to control representation drift Tao et al. (2020); Hersche et al. (2022); Zhang et al. (2021); Zhu et al. (2021). While these approaches achieve strong accuracy in controlled benchmarks, they often require additional training stages, careful tuning of multiple loss terms, or replay-style refinement. As a result, their computational cost scales poorly, and performance can degrade when assumptions about task distributions or prototype structure are violated. To improve efficiency, several FSCIL methods freeze most or all of the backbone after the base session and restrict learning to lightweight classifier or prototype updates Zhou et al. (2022a); Yang et al. (2023). This strategy substantially reduces training cost and limits forgetting in the few-shot regime. However, it also implicitly assumes that features learned during base training remain sufficient for all future tasks, an assumption that breaks down in cross-domain FSCIL, resulting in severe performance degradation.

A promising line of work addresses forgetting by explicitly allocating network capacity to tasks through subnetworks, masking, or parameter isolation Mallya & Lazebnik (2018); Serra et al. (2018). Early methods demonstrate strong performance in the task-incremental learning (TIL) setting, where task identity is available at test time, but they are unsuitable for FSCIL. More recent work adapts these ideas to FSCIL, for example, via sparse or "winning ticket" subnetworks Mazumder et al. (2021); Kang et al. (2024). However, these methods require learning importance masks through additional optimization, or incur nontrivial overhead due to per-task search, pruning, or retraining. Moreover, aggressive isolation can fragment capacity and limit forward transfer to later tasks. These limitations motivate FSCIL-compatible capacity allocation strategies that control interference deterministically, without task inference or expensive mask learning.

In this direction, several recent works have explored using prompting mechanisms on large pre-trained foundation models (e.g., Vision Transformers) for FSCIL Li et al. (2025b); Wang et al. (2024). These approaches depart from the strict, learning-from-scratch, FSCIL setting studied in this paper. They rely heavily on massive external pre-training datasets, making it difficult to disentangle incremental learning from prior knowledge embedded in the model. This dependence introduces leakage from pre-training and obscures the model's true ability to acquire, retain, and adapt knowledge incrementally. Furthermore, because such models are typically frozen or only weakly adapted, they often exhibit limited robustness under domain shift and do not support genuinely continual representation learning. As a result, prompting-based methods are best viewed as a complementary direction rather than a solution to exemplar-free FSCIL under realistic deployment constraints.

5 Conclusion

This paper introduced *Selective Backpropagation* (SBP), an efficient parameter-budgeting framework for exemplar-free few-shot class-incremental learning. SBP addresses a central tension in FSCIL: standard fine-tuning is computationally efficient and adaptive but prone to catastrophic forgetting, whereas many strong FSCIL methods preserve stability through costly replay, rigid backbone freezing, or limited plasticity. By preemptively allocating capacity, freezing parameters associated with past knowledge, and reinitializing future capacity at each epoch, SBP preserves stability while avoiding implicit dependence on unused parameters. When combined with prototype-based classification, SBP achieves state-of-the-art performance across multiple FSCIL benchmarks while substantially reducing training cost relative to competing methods. Across standard benchmarks, cross-domain shifts, and an 80-session ImageNet-1K stream, our results show that strong performance on short, distribution-consistent FSCIL benchmarks alone is insufficient to characterize robust behavior over time. Methods that effectively preserve a fixed representation can remain competitive when future classes resemble the base distribution, yet degrade when actual adaptation or long-term interference becomes important. These findings motivate evaluating FSCIL together alongside stability, plasticity, and computational efficiency, with explicit evaluations for distribution shift and learning horizon. SBP performs strongly across these regimes, suggesting that controlled representational plasticity is an important component of practical FSCIL systems.

REFERENCES

Noor Ahmed, Anna Kukleva, and Bernt Schiele. Orco: Towards better generalization via orthogonality and contrast for few-shot class-incremental learning. In *41st IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2024.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

Ali Cheraghian, Shafin Rahman, Pengfei Fang, Soumava Kumar Roy, Lars Petersson, and Mehrtash Harandi. Semantic-aware knowledge distillation for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2534–2543, 2021.

Zhixiang Chi, Li Gu, Huan Liu, Yang Wang, Yuanhao Yu, and Jin Tang. Metafscil: A meta-learning approach for few-shot class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14166–14175, 2022.

Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8250–8259, 2021.

Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. ISSN 1939-3539. doi: 10.1109/tpami.2021.3057446. URL http://dx.doi.org/10.1109/TPAMI.2021.3057446.

Alex Gomez-Villa, Dipam Goswami, Kai Wang, Andrew D Bagdanov, Bartlomiej Twardowski, and Joost Van De Weijer. Exemplar-free continual representation learning via learnable drift compensation. In *European Conference on Computer Vision*, pp. 473–490. Springer, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Michael Hersche, Geethan Karunaratne, Giovanni Cherubini, Luca Benini, Abu Sebastian, and Abbas Rahimi. Constrained few-shot class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9057–9067, 2022.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Haeyong Kang, Jaehong Yoon, Sung Ju Hwang, and Chang D Yoo. Continual learning: Forget-free winning subnetworks for video representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):2140–2156, 2024.

Do-Yeon Kim, Dong-Jun Han, Jun Seo, and Jaekyun Moon. Warping the space: Weight space rotation for class-incremental few-shot learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=kPLzOfPfA2l.

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. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Linhao Li, Yongzhang Tan, Siyuan Yang, Hao Cheng, Yongfeng Dong, and Liang Yang. Adaptive decision boundary for few-shot class-incremental learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18359–18367, 2025a.

Shuo Li, Fang Liu, Licheng Jiao, Lingling Li, Puhua Chen, Xu Liu, and Wenping Ma. Prompt-based concept learning for few-shot class-incremental learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(5):4991–5005, 2025b.

Huan Liu, Li Gu, Zhixiang Chi, Yang Wang, Yuanhao Yu, Jun Chen, and Jin Tang. Few-shot class-incremental learning via entropy-regularized data-free replay. In *European Conference on Computer Vision*, pp. 146–162. Springer, 2022.

Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 67–82, 2018.

Pratik Mazumder, Pravendra Singh, and Piyush Rai. Few-shot lifelong learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 2337–2345, 2021.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

Can Peng, Kun Zhao, Tianren Wang, Meng Li, and Brian C Lovell. Few-shot class-incremental learning from an open-set perspective. In *European Conference on Computer Vision*, pp. 382–397. Springer, 2022.

Hang Qi, Matthew Brown, and David G. Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Vinay V Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. *arXiv preprint arXiv:2007.07400*, 2020.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning*, pp. 4548–4557. PMLR, 2018.

Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Xuan Wang, Zhong Ji, Xiyao Liu, Yanwei Pang, and Jungong Han. On the approximation risk of few-shot class-incremental learning. In *European Conference on Computer Vision*, pp. 162–178. Springer, 2024.

Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3014–3023, 2021.

Yibo Yang, Haobo Yuan, Xiangtai Li, Zhouchen Lin, Philip Torr, and Dacheng Tao. Neural collapse inspired feature-classifier alignment for few-shot class-incremental learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=y5W8tpojhtJ.

Sung Whan Yoon, Do-Yeon Kim, Jun Seo, and Jaekyun Moon. Xtarnet: Learning to extract task-adaptive representation for incremental few-shot learning. In *International conference on machine learning*, pp. 10852–10860. PMLR, 2020.

Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12455–12464, 2021.

Hanbin Zhao, Yongjian Fu, Mintong Kang, Qi Tian, Fei Wu, and Xi Li. Mgsvf: Multi-grained slow versus fast framework for few-shot class-incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(3):1576–1588, 2021.

Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9046–9056, 2022a.

Da-Wei Zhou, Han-Jia Ye, Liang Ma, Di Xie, Shiliang Pu, and De-Chuan Zhan. Few-shot class-incremental learning by sampling multi-phase tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12816–12831, 2022b.

Kai Zhu, Yang Cao, Wei Zhai, Jie Cheng, and Zheng-Jun Zha. Self-promoted prototype refinement for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6801–6810, 2021.

Yixiong Zou, Shanghang Zhang, Haichen Zhou, Yuhua Li, and Ruixuan Li. Compositional few-shot class-incremental learning. *arXiv preprint arXiv:2405.17022*, 2024.

A Experimental Setup

Implementation Details. We utilize a ResNet-18 He et al. (2016) backbone as the feature extractor f_θ for all models, including ours. We use an SGD optimizer with momentum 0.9 for all training sessions. During the base session, we train for 130 epochs on CIFAR-100 and 200 epochs on MiniImageNet. The learning rate is initialized to $\eta = 0.1$ and decayed using a cosine annealing schedule. After base training, similar to Kang et al. (2024), all batch normalization statistics are frozen to prevent representation drift. During incremental sessions, we fine-tune only the assigned parameters for 40 iterations on CIFAR-100 and 120 iterations on MiniImageNet using a reduced learning rate of $\eta = 0.0025$. We set an initial budget slice of $\beta = 0.85$, where β controls the proportion of parameters in each layer selected from the total capacity and assigned to the current session. Specifically, this means 85% of the parameters in each layer are assigned to the base classes. The remaining 15% capacity is distributed across incremental sessions, allocating approximately $\sim 1.8\%$ of the parameters per layer to each new task. However, SBP does not rely on this knowledge, and remains robust against a wide range of parameter budgets (Appendix G). All experiments on CIFAR-100 and MiniImageNet are run on a single NVIDIA RTX A4500 GPU. The ablation study was conducted on an NVIDIA A100 Tensor Core GPU, utilizing a 20GB Multi-Instance GPU (MIG) partition. We report the mean and standard deviation over five independent runs using fixed global random seeds for CIFAR-100, MiniImageNet, and the CUB-200-2011 dataset experiments. The global random seed also determines the random SBP parameter allocation. Consequently, each independent run uses a different realization of the session-wise allocation masks, so the reported run-to-run variation also captures sensitivity to random parameter assignment. We will publicly release our code after the paper is accepted.

Datasets. We evaluate our method on three well-established benchmark datasets for FSCIL: CIFAR-100 Krizhevsky et al. (2009), MiniImageNet Vinyals et al. (2016), and the Caltech–UCSD Birds-200-2011 (CUB-200) Wah et al. (2011). CIFAR-100 consists of 60,000 RGB images of size 32×32 across 100 classes, with 50,000 training images and 10,000 test images. MiniImageNet is a subset of ImageNet that contains 100 classes. Each class has 600 images, for a total of 60,000 images, all resized to 84×84. CUB-200 consists of 200 bird species and 11,788 images. Following standard practice Kang et al. (2024), all images are resized to 224×224. For CIFAR-100 and MiniImageNet, we follow the standard FSCIL protocol established in Tao et al. (2020). Specifically, the base session trains a ResNet-18 model from scratch on 60 classes with full supervision. The remaining 40 classes are then learned incrementally across 8 sessions, each introducing 5 new classes under a 5-way 5-shot setting. Overall, the model learns all classes over 9 sequential sessions without revisiting old data. For CUB-200, a ResNet-18 backbone pre-trained on ImageNet-1K is finetuned for 60 epochs on the first 100 classes using all available samples, and the remaining 100 classes are introduced over 10 incremental sessions trained for 120 incremental iterations, each containing 10 novel classes under a 10-way 5-shot setting.

Baselines. We compare our method against a strong set of representative FSCIL baselines. On CIFAR-100 and MiniImageNet, we evaluate methods such as Entropy-Reg Liu et al. (2022), ALICE Peng et al. (2022), FACT Zhou et al. (2022a), NC-FSCIL Yang et al. (2023), OrCo Ahmed et al. (2024), Comp Zou et al. (2024), Approx Wang et al. (2024), ADBS Li et al. (2025a), SoftNet Kang et al. (2024), and WaRP Kim et al. (2023). We reproduced and evaluated all baselines to ensure a fair comparison. On CUB-200-2011, we compare against some of the previously mentioned baselines including ALICE Peng et al. (2022), FACT Zhou et al. (2022a), WaRP Kim et al. (2023), OrCo Ahmed et al. (2024), Comp Zou et al. (2024), and NC-FSCIL Yang et al. (2023). To ensure a fair comparison, we evaluated all models on the same settings and backbone used for SBP. Additionally, we fixed a few errors, such as the use of the test set as the validation set, for some methods to ensure a fair comparison.

A.1 Partial Distance Classification

Conventional distance-based classifiers operate over the full feature embedding space $\mathbb{R}^D$, causing prototype comparisons to include dimensions associated with capacity that has not yet been allocated for learning. Under SBP, these dimensions may contain unstable activations because their corresponding parameters are reserved for future incremental sessions. We therefore restrict prototype comparisons to the feature dimensions that have been allocated up to the current session. As SBP progressively assigns new capacity, the active classification subspace grows accordingly.

Active Subspace Projection. For the convolutional backbones used in our standard experiments, the output channels of the final convolution correspond directly to feature dimensions after global pooling. Let $\mathbf{M}_{\text{assigned}}^{(j)}$ denote the feature dimensions allocated at session j. At session t, we define the active feature mask as

$$\mathbf{M}_{\text{active}}^{(t)} = \bigvee_{j=0}^{t} \mathbf{M}_{\text{assigned}}^{(j)},$$

where $\vee$ denotes the element-wise logical OR. The corresponding projection is

$$\Phi_t(\mathbf{v}) = \mathbf{v} \odot \mathbf{M}_{\text{active}}^{(t)}.$$

Thus, dimensions reserved for future sessions are excluded from classification, while newly allocated dimensions become available to the classifier as incremental learning proceeds.

Prototype Imprinting and Inference. For each new class c, we use standard weight imprinting and store its prototype as the centroid of the full support features at the session t_c in which the class is introduced:

$$p_c = \frac{1}{|D_c|} \sum_{x \in D_c} f_{\theta_{t_c}}(x).$$

The prototype is then frozen. At inference during session t, distances are computed over the feature dimensions allocated up to the current session:

$$d_c^{(t)}(x) = d\Big(M_{\text{active}}^{(t)} \odot f_{\theta_t}(x), M_{\text{active}}^{(t)} \odot p_c\Big).$$

Table 6: CIFAR-100 FSCIL (5-way 5-shot) per-session runtime (minutes) with ResNet-18 using the TOPIC class split. The "Backbone Update" column indicates whether the backbone is tuned during incremental sessions or frozen after base training. Session 1 reports the base session (60 classes) runtime. Lower runtime is better; best and second-best values in each runtime column are shown in **bold** and underlined, respectively. Results are averaged over five runs with different random seeds.

Method	**Backbone**	**Sessions (time in minutes)**									**Total**
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**(min)**
Entropy-Reg Liu et al. (2022)	Tuned	56	52	52	53	53	54	55	55	56	486
ALICE Peng et al. (2022)	Frozen	188	1	1	1	1	1	1	1	1	196
FACT Zhou et al. (2022a)	Frozen	39	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	40
WaRP Kim et al. (2023)	Tuned	19	0.08	0.05	0.05	0.05	0.05	0.05	0.05	0.06	20
NC-FSCIL Yang et al. (2023)	Frozen	84	0.03	0.04	0.06	0.06	0.07	0.08	0.08	0.08	85
OrCo Ahmed et al. (2024)	Frozen	27	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	28
Comp Zou et al. (2024)	Frozen	88	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	89
Approx Wang et al. (2024)	Frozen	295	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	296
ADBS Li et al. (2025a)	Frozen	27	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	27
SoftNet Kang et al. (2024)	Tuned	66	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	67
Ours	Tuned	**14**	0.05	0.06	0.06	0.06	0.06	0.05	0.04	0.05	**14**

This projection excludes dimensions that remain unallocated at session t, preventing currently free capacity from directly affecting classification.

B Incremental Training Time Comparison with SOTA

We provide a breakdown of the training time per session for our method compared to current state-of-the-art (SOTA) methods. As aforementioned, the practicality of FSCIL is often limited by the computational cost of training in resource-constrained environments. Table 6 reports the per-session runtimes on CIFAR-100 (5-way 5-shot) over nine sessions. For a fair comparison, all methods used the same backbone (ResNet-18), and were run on the same NVIDIA A4500 GPU. Our method achieves the fastest base-session training, completing Session 1 in approximately 14 minutes. This is $1.36\times$ faster than the closest competitor, WaRP, and almost $5\times$ faster than SoftNet. In contrast, several competing approaches require over one hour of base training, with ALICE and Approx exceeding 100 minutes. During incremental sessions (Sessions 2–9), our method consistently maintains low latency, requiring approximately 0.04–0.06 minutes per session. Methods that freeze the backbone after the base session (FACT, NC-FSCIL), generally achieve slightly better per-session training time (0.03 minutes) in most sessions. However, note that this comes at the cost of lower accuracy in incremental sessions, particularly in cases when the incremental sessions contain data that might be different from data used in the base session. Additionally, these methods often require substantially longer base training to reach peak performance. SoftNet achieves similar per-session training time as our method after the base session, as both methods update only a small number of parameters in the backbone. Other methods, such as ALICE and WARP, incur significantly larger incremental overheads, ALICE incurs approximately 1 minute per incremental session, whereas WaRP requires approximately 0.05–0.08 minutes, on top of their high base session training times. These results indicate that our method achieves a favorable balance between base-session training time and incremental update cost while achieving the best accuracy compared to SOTA baselines.

C Results on CUB-200-2011

We report additional results on the Caltech–UCSD Birds-200-2011 (CUB-200) dataset, comprising 200 bird species and 11,788 images. All methods follow the same training settings and the backbone as reported in Appendix A. Table 7 reports the results on the CUB-200-2011 benchmark under the 10-way 5-shot FSCIL setting. SBP achieves the best accuracy across most sessions, with a final-session accuracy of 62.51% and an average accuracy of 71.28%. This corresponds to a 3.40%

Table 7: CUB-200-2011 FSCIL (10-way 5-shot) classification accuracy (%) with ResNet-18. Best (per column) is in **bold**, second best is underlined. Results are averaged over five runs with different random seeds.

Method	Sessions (Accuracy %)											Avg (%)
	1	2	3	4	5	6	7	8	9	10	11	
ALICE Peng et al. (2022)	67.63	58.42	42.51	37.58	55.33	36.24	55.90	49.83	48.47	**65.97**	58.00	52.35
FACT Zhou et al. (2022a)	65.02	62.23	58.92	55.14	54.41	51.97	51.50	49.02	49.25	48.53	47.85	53.99
WaRP Kim et al. (2023)	77.65	74.80	71.60	67.44	64.92	61.88	60.94	58.75	55.86	53.66	49.91	63.40
NC-FSCIL Yang et al. (2023)	78.42	71.08	67.41	63.49	59.91	57.61	55.29	54.39	51.23	49.13	48.02	59.63
OrCo Ahmed et al. (2024)	73.01	65.70	63.64	61.94	60.45	59.22	58.41	57.59	55.84	56.44	56.11	60.76
Comp Zou et al. (2024)	78.39	74.51	71.17	67.41	65.02	63.01	62.38	61.39	59.74	59.74	59.11	65.62
Ours	**82.25**	**78.82**	**76.76**	**73.91**	**72.11**	**69.98**	**69.12**	**67.96**	**66.34**	64.27	**62.51**	**71.28**

Table 8: Architecture study of our method on CIFAR-100 FSCIL (5-way 5-shot). We report classification accuracy (%) using different backbone networks, including CNNx4, CNNx8, CNNx16, ResNet-18, ResNet-50, and ResNeXt-29. Best (per column) is in **bold**, second best is underlined.

Backbone	Sessions (Accuracy %)									Avg (%)
	1	2	3	4	5	6	7	8	9	
Without pretraining										
CNNx4	61.92	58.75	56.13	53.72	50.93	48.62	45.62	43.95	42.03	51.30
CNNx8	65.30	61.91	59.14	56.52	54.10	51.78	49.21	47.58	45.89	54.60
CNNx16	68.05	64.43	61.76	58.93	56.13	53.82	51.18	49.44	47.50	56.80
ResNet-18	82.50	77.22	72.69	70.55	67.76	64.40	62.76	60.31	58.52	68.52
ResNet-50	**86.83**	**82.06**	77.76	73.51	69.45	65.94	62.47	60.26	57.84	70.68
ResNeXt-29	85.30	81.55	**78.23**	**74.73**	**71.50**	**68.75**	**65.53**	**63.12**	**61.11**	**72.20**

improvement in final accuracy over the strongest competing method, Comp, and a 5.66% improvement in average accuracy over Comp. These results further confirm that SBP generalizes beyond CIFAR-100 and MiniImageNet, providing robust performance across diverse FSCIL settings.

D Cross-Domain Mini-ImageNet to Caltech-101

To further evaluate the robustness of the cross-domain results presented in Section 3.2, we conduct a second experiment using a different cross-domain dataset pair. Specifically, the base session is trained on 60 classes from Mini-ImageNet, while each subsequent session introduces 10 novel classes from Caltech-101 using only five examples per class. This setting is particularly challenging because the model must not only learn new classes from limited supervision, but also adapt to a different domain from that observed during base training. Results are reported in Table 9.

The results show that SBP remains robust under this larger distribution shift. SBP achieves the highest average accuracy over the complete sequence, reaching 64.87%, compared with 62.64% for the next-best method, ALICE. SBP also achieves the highest final-session accuracy of 58.77%, exceeding WaRP, the second-best method at the final session, by 4.38%.

The early sessions further illustrate the difficulty of the domain transition. ALICE initially performs slightly better than our method, achieving 71.60% and 68.90% at Sessions 1 and 2, compared with 70.34% and 67.88% for SBP. However, SBP overtakes ALICE by Session 3 and remains the best-performing method for every subsequent session. By Session 10, the gap in performance has grown to 5.77% over ALICE. This pattern suggests that the benefit of SBP becomes more pronounced as additional classes from the target domain are introduced and interference accumulates across incremental updates.

Table 9: Cross-Domain FSCIL (MiniImageNet → Caltech-101) classification accuracy (%) with a ResNet-18 backbone. Best (per column) is in **bold**, second best is underlined.

Method	Sessions (Accuracy %)											Avg (%)
	0	1	2	3	4	5	6	7	8	9	10	
ALICE Peng et al. (2022)	76.50	**71.60**	**68.90**	66.40	63.70	61.70	59.50	57.60	55.80	54.30	53.00	62.64
OrCo Ahmed et al. (2024)	76.50	63.60	61.40	59.20	58.00	57.00	55.40	54.50	53.10	51.80	50.30	58.25
ADBS Li et al. (2025a)	**82.10**	68.30	61.60	50.70	44.20	41.60	37.80	33.50	30.20	28.90	26.60	45.95
WaRP Kim et al. (2023)	69.41	64.41	63.15	61.86	60.06	58.70	57.68	57.22	56.11	55.19	54.39	59.83
Ours	77.33	70.34	67.88	**66.70**	**65.98**	**64.27**	**62.86**	**60.47**	**59.76**	**59.21**	**58.77**	**64.87**

Table 10: Long-horizon FSCIL on ImageNet-1K. We report classification accuracy (%) at selected sessions throughout the incremental sequence. Best (per column) is in **bold**, second best is underlined.

Method	Sessions (Accuracy %)										
	0	1	5	10	20	30	40	50	60	70	80
ADBS Li et al. (2025a)	61.50	57.10	47.20	39.20	27.40	22.20	18.80	17.20	16.00	14.90	14.50
OrCo Ahmed et al. (2024)	27.00	24.80	24.90	25.10	24.30	23.20	21.90	21.00	20.00	19.30	18.40
WaRP Kim et al. (2023)	63.20	61.90	60.40	58.60	54.90	51.40	47.80	42.10	33.30	20.00	12.90
ViT (SBP)	**73.56**	**70.19**	**65.03**	**63.28**	**58.85**	**55.78**	**53.08**	**50.77**	**48.93**	**46.85**	**45.07**
Ours	72.45	70.09	64.99	61.85	54.26	50.96	47.93	45.19	43.01	40.83	39.08

The results also show that strong base-session performance alone is not sufficient for robust cross-domain FSCIL. ADBS obtains the highest base-session accuracy at 82.10%, but its performance decreases rapidly after the domain shift, reaching only 26.60% by Session 10. In contrast, SBP starts from a lower base accuracy of 77.33% but retains 58.77% after all incremental sessions. This represents a much smaller degradation across the sequence and highlights the importance of preserving previously learned representations while still allowing sufficient capacity to adapt to the target domain.

WaRP exhibits comparatively stable performance after the domain shift, declining from 64.41% at Session 1 to 54.39% at Session 10. WaRP also remains comparatively stable after the domain shift, but SBP maintains higher accuracy throughout the incremental sequence, with a 4.38% advantage at the final session.

Overall, these results reinforce the findings from Section 3.2, showing that SBP is not limited to settings where base and incremental classes originate from the same data distribution. SBP shows strong performance even when incremental classes are drawn from a different dataset and domain, where the model must balance retention of the source-domain representation with adaptation to novel target-domain classes from very limited data.

E ViT Long-Horizon Learning

For the long-horizon ImageNet-1K experiment, we pretrain the backbone using 600 classes for the base session and introduce the remaining 400 classes over 80 incremental sessions, each following a 5-way 5-shot protocol. The ResNet-18 backbone is trained from scratch using only the 600 base classes. We treat this base-training stage as the representation-learning stage for Session 0 and subsequently imprint the base-class prototypes without additional base-session optimization. During incremental learning, each session is trained for 75 iterations. We allocate 90% of the network capacity to the base session and distribute the remaining 10% uniformly across the 80 incremental sessions. Few-shot support examples are sampled independently for each session using a fixed random seed, and evaluation is performed on the ImageNet-1K validation set restricted to all classes observed up to the current session.

Table 11: Impact of base session capacity budget on CIFAR-100 FSCIL performance. We vary the fraction of network capacity allocated to the base session and report per-session classification accuracy (%) and average accuracy over all sessions. Best (per column) is in **bold**, second best is underlined.

Budget	Sessions (Accuracy %)									Avg (%)
	1	2	3	4	5	6	7	8	9	
50%	74.87	71.20	68.09	65.32	62.14	59.28	56.14	53.68	52.14	62.54
75%	81.53	76.98	73.26	69.55	66.09	63.69	60.21	57.84	55.87	67.22
80%	81.93	77.62	74.01	70.47	66.81	64.39	61.37	59.03	57.01	68.07
85%	82.65	77.98	74.31	70.89	67.35	64.36	60.77	58.47	57.84	68.29
90%	82.32	77.95	74.17	70.75	67.06	64.73	61.73	59.37	56.89	68.33
95%	**83.27**	**79.25**	**75.60**	**72.33**	**68.61**	**66.00**	**62.74**	**60.48**	**58.14**	**69.60**
99%	83.25	79.09	74.97	71.56	68.08	65.28	62.01	59.56	57.44	69.03

The ViT variant achieves the highest accuracy across the reported sequence and retains 45.07% accuracy after Session 80, compared with 39.08% for ResNet-18 SBP. Importantly, the ViT is trained from scratch exclusively on the same 600 ImageNet-1K base classes and does not use external pretraining. Its improvement therefore reflects the effect of a stronger backbone architecture rather than additional prior data. Together, the ResNet-18 and ViT results indicate that SBP's long-horizon behavior is not tied to a single architecture and can benefit from increased representational capacity.

F Architecture Study

We conduct an extensive architectural study on CIFAR-100 (same setup as in ablation study) to further validate the robustness and scalability of our method. Specifically, we test shallow convolutional networks (CNN-4, CNN-8, CNN-16) as well as residual networks (ResNet-18, ResNet-50) and ResNeXt-29 to demonstrate that SBP is not tied to a specific network topology and scales effectively with representational capacity. Table 8 summarizes the per-increment and average accuracy for each backbone. We observe a strong positive correlation between network capacity and FSCIL performance. As we move from CNNx4 (51.30% Avg.) to CNNx16 (56.80% Avg.), our method consistently improves in per-increment accuracies. The transition to ResNet-based backbones results in a significant performance leap, with ResNet-50 achieving an average accuracy of 70.68%, followed by ResNet-18 at 68.52%. ResNeXt-29 further improves the average accuracy to 72.20%, indicating that grouped-convolution architectures can provide additional capacity benefits under the same from-scratch FSCIL setting. Across all tested architectures, SBP maintains a gradual accuracy decline across sessions rather than abrupt collapse, suggesting that its parameter-budgeting mechanism remains effective across backbone families and capacity scales.

G Effect of Base Session Budgets

We investigate the sensitivity of SBP to the base-session capacity budget by varying the fraction of network capacity allocated to the base session while keeping the training schedule and all incremental-session hyperparameters fixed. Specifically, we train the base session for 130 epochs and then run the same 8 incremental sessions (5-way 5-shot) as in the main experiments. The goal is to quantify how the base session budget influences the representation that supports efficient incremental adaptation.

Table 11 reports the per-session classification accuracy for budgets of $\{50\%, 75\%, 80\%, 85\%, 90\%, 95\%, 99\%\}$. Overall, increasing the base session budget improves performance consistently across the incremental stream, with the largest gains appearing at the earliest increments. In particular, moving from 50% to 75% yields a substantial improvement in average accuracy (62.54% $\rightarrow$ 67.22%) and improves the final-session accuracy (52.14% $\rightarrow$ 55.87%, +3.73%), indicating that allocating more capacity to the base session strengthens the initial

representation. This is likely because assigning too many parameters to incremental sessions with only a few training samples leads to suboptimal results.

Beyond 75%, gains become more incremental: 80% and 90% improve the average accuracy to 68.07% and 68.33%, respectively, while 95% achieves the best overall average accuracy at 69.60%. Interestingly, 99% is very close at 69.03%, suggesting diminishing returns from further allocating additional capacity to the base session. We believe this is likely due to the model's backbone being largely frozen with a base budget of 99%. Incremental sessions therefore have limited capacity to reshape global features, and performance becomes dominated by the quality of the base-session representation and the geometry it induces for prototype-based classification.

Our results suggest that moderate base session budgets (e.g., 75%–90%) already recover most of the attainable performance while preserving lower base training cost, whereas near-full budgets (95%–99%) offer only marginal additional gains in average accuracy. In settings where the base session is too expensive to train or must be trained quickly, reducing the base budget to 75% can provide a strong accuracy–efficiency compromise, while still maintaining stable incremental behavior.

H Theoretical Motivation for Free Neuron Re-initialization in SBP

In Section 2.1, we introduce the epoch-wise re-initialization of free parameters ($\boldsymbol{\theta}_{\text{free}}$) as a core component of SBP. A critical challenge in parameter isolation is preventing unallocated capacity from becoming entangled with the current session's optimization. To formalize this, consider the network's representation at session t as a linear superposition of three subspaces:

$$f_t(x) = \mathbf{W}^{l,t}_{\text{prev}} x + \mathbf{W}^{l,t}_{\text{assigned}} x + \mathbf{W}^{l,t}_{\text{free}} x, \tag{3}$$

where $\mathbf{W}_{\text{prev}}$ and $\mathbf{W}_{\text{assigned}}$ are the weights allocated to past and current tasks, respectively, and $\mathbf{W}_{\text{free}}$ is the unallocated pool reserved for future sessions.

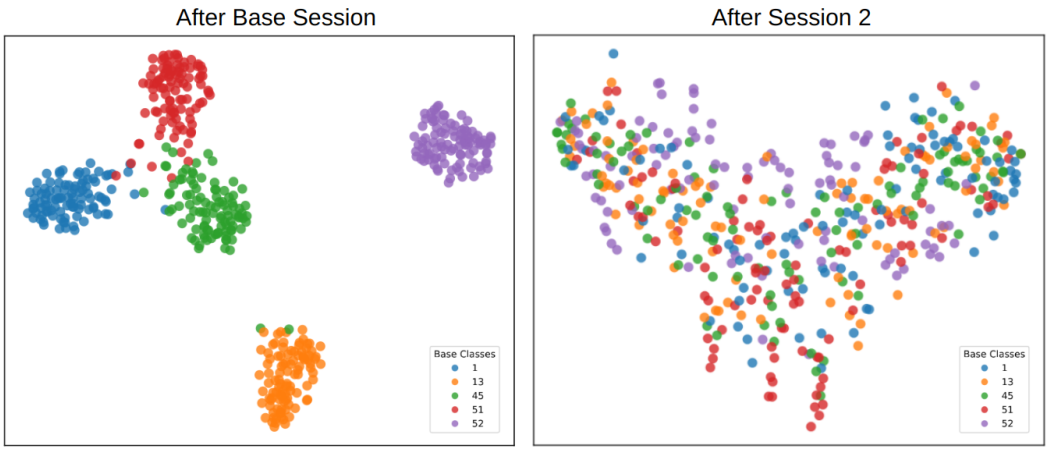
If the free parameters $\mathbf{W}_{\text{free}}$ remain static during the optimization of task t, the greedy nature of backpropagation will cause $\mathbf{W}_{\text{assigned}}$ to co-adapt to the specific noise realization present in the free block to minimize the task loss $\mathcal{L}_t$. This creates a latent coupling where the current task's performance becomes contingent on the state of future capacity. Consequently, when a subset of $\mathbf{W}_{\text{free}}$ is promoted to $\mathbf{W}_{\text{assigned}}$ in session $t+1$, it no longer follows a clean, unbiased distribution, but is instead biased by the training history of previous sessions.

To enforce strict temporal decoupling, SBP resets the free partition $\boldsymbol{\theta}^l_{\text{free}}$ at the start of every epoch to independent draws from a zero-mean distribution $\mathcal{D}$ (e.g., Kaiming Uniform). By treating the free parameters as a zero-mean random variable, we ensure that their expected contribution to the output vanishes over the course of optimization:

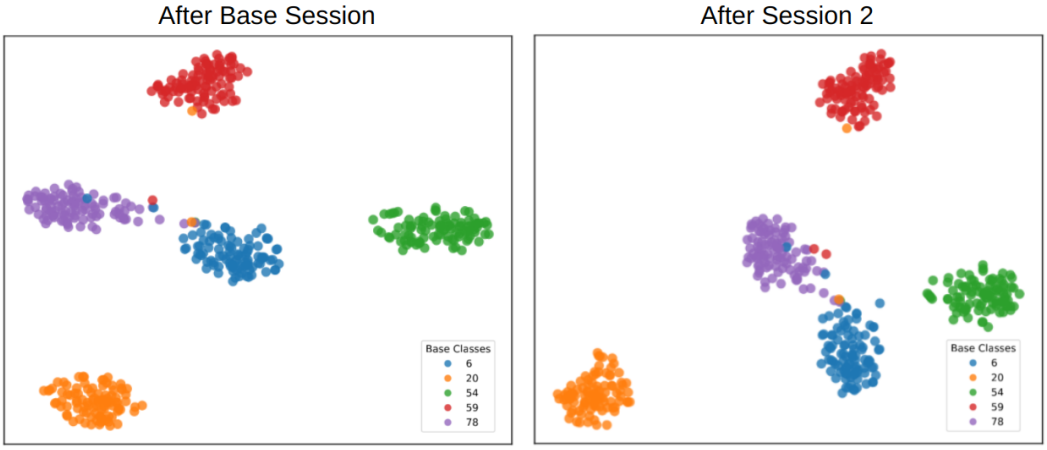
$$\mathbb{E}_{\mathbf{W}_{\text{free}} \sim \mathcal{D}} \left[\mathbf{W}^{l,t}_{\text{free}} x \right] = 0. \tag{4}$$

This formulation forces the optimizer to find a solution for $\mathbf{W}_{\text{assigned}}$ that is invariant to the values in the free subspace. This mechanism provides two key benefits: (i) the current task learns robust representations that do not consume unused capacity, and (ii) future tasks inherit an unbiased initialization that is independent of the previous training history and optimizer states.

Importantly, free-parameter reinitialization is not merely an implementation detail, but the mechanism that temporally decouples current learning from future capacity. Without reinitialization, unused parameters can remain fixed during training and become a stable, implicit part of the current task's solution, even though they receive no gradients. This creates hidden co-adaptation between θ_{assigned} and θ_{free}, so that when free parameters are later reassigned to future sessions, changing them can destroy representations learned earlier. By reinitializing θ_{free} at every epoch, SBP turns the free subspace into a non-stationary nuisance component. The optimizer is therefore forced to learn current-task representations that are invariant to this subspace and depend only on the assigned parameters. As a result, current tasks do not consume future capacity, and future tasks inherit parameters that remain statistically independent of previous-session optimization. This explains why SBP can use all parameters at inference time without explicit forward masking: the learned representation has been trained to ignore the free subspace. The severe collapse observed when removing



(a) *Without* free-parameter re-initialization.



(b) *With* epoch-wise free-parameter re-initialization.

Figure 2: Evolution of base-class features with and without SBP's free-parameter reinitialization. Each point represents a sample, with color indicating its class. Without reinitialization, base classes are initially separated after Session 1, but collapse after Session 2 once the implicitly co-adapted free parameters are reassigned and updated. With epoch-wise reinitialization, the feature clusters remain compact and well separated after Session 2, indicating that the learned representation is invariant to changes in the free subspace.

free-parameter reinitialization further confirms that this mechanism is essential for preventing latent co-adaptation and preserving reusable capacity across incremental sessions.

I Feature-Space Analysis of Free-Parameter Reinitialization

In our ablation studies (see Section 3.4), we observed that removing the epoch-wise re-initialization of free parameters (*w/o Free-Init*) results in an abrupt model failure, causing accuracy to plummet to $\sim$ 1.56% by the final session. Here, we provide a detailed diagnosis of this failure mode and visually demonstrate why SBP's temporal decoupling is essential.

To illustrate this, we visualize the feature embeddings of five base classes from the CIFAR-100 dataset, using t-SNE. As shown in Figure 2(a), without free-parameter re-initialization, the base classes form distinct, well-separated clusters after Session 1. However, immediately after Session 2, the feature space collapses; the previously distinct clusters scatter and overlap entirely, visually confirming catastrophic forgetting. Conversely, Figure 2(b) shows the feature space evolution when utilizing the full SBP method with epoch-wise re-initialization. Because the free parameters remain

non-stationary during the base session, the optimizer is strictly prevented from utilizing them to separate the base classes. Consequently, when these parameters are assigned and updated in Session 2, the base representations remain untouched. This explains why free-parameter reinitialization is critical for SBP: it prevents latent co-adaptation, preserves clean capacity for future sessions, and enables controlled plasticity without requiring explicit forward masking at inference time.