

Critical Patch-Aware Sparse Prompting with Decoupled Training for Continual Learning on the Edge

Wonseon Lim¹ Jaesung Lee^{2*} Dae-Won Kim^{1*}

¹School of Computer Science and Engineering, Chung-Ang University

²Department of Artificial Intelligence, Chung-Ang University

{costor, curseor, dwkim}@cau.ac.kr

Abstract

Continual learning (CL) on edge devices requires not only high accuracy but also training-time efficiency to support on-device adaptation under strict memory and computational constraints. While prompt-based continual learning (PCL) is parameter-efficient and achieves competitive accuracy, prior work has focused mainly on accuracy or inference-time performance, often overlooking the memory and computational costs of on-device training. In this paper, we propose CPS-Prompt, a critical patch-aware sparse prompting framework that explicitly targets training-time memory usage and computational cost by integrating critical patch sampling (CPS) for task-aware token reduction and decoupled prompt and classifier training (DPCT) to reduce backpropagation overhead. Experiments on three public benchmarks and real edge hardware show that CPS-Prompt improves peak memory, training time, and energy efficiency by about $1.6\times$ over the balanced CODA-Prompt baseline, while maintaining accuracy within 2% of the state-of-the-art C-Prompt on average and remaining competitive with CODA-Prompt in accuracy. The code is available at <https://github.com/laymond1/cps-prompt>.

1. Introduction

Modern continual learning (CL) systems are increasingly expected to adapt directly on edge devices such as home robots, drones, or smartphones, where retraining in the cloud is infeasible or privacy-restricted [7, 18, 34]. In such scenarios, the model must efficiently update as new tasks arrive while operating under shared CPU–GPU memory and limited computational budgets [6, 26, 27]. Although most research has focused on reducing inference-time latency for edge deployment, on-device continual training remains an equally critical yet underexplored challenge. Excessive in-

*Corresponding authors.

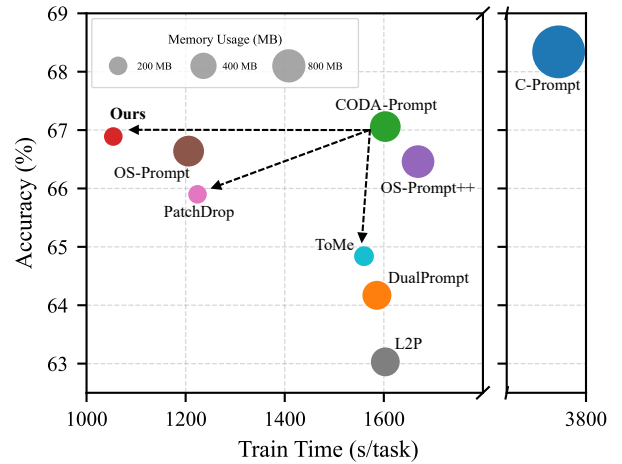


Figure 1. Comparison of accuracy and training-time efficiency on CIFAR-100 using PCL and token reduction methods, with efficiency metrics measured on a Jetson Orin Nano. Our method exhibits a more balanced accuracy–efficiency trade-off under edge-device constraints.

intermediate activations can easily exceed device memory capacity, leading to instability or training failures. Hence, improving training-time memory and compute efficiency is essential for sustainable continual adaptation on edge hardware.

Prompt-based continual learning (PCL) [22, 28, 29] offers an appealing solution by reusing a frozen vision transformer (ViT) backbone [3] and updating only lightweight prompt parameters for each task. PCL typically follows a two-stage feed-forward pipeline: a frozen query forward pass that produces task-relevant cues for prompt selection, followed by a prompt-injected forward pass for classification. This design achieves high accuracy with few trainable parameters while mitigating catastrophic forgetting in the pretrained backbone. However, existing PCL methods largely optimize for accuracy, with training-time re-

source use on constrained hardware remaining underexplored [5, 16]. Recent methods such as C-Prompt [5] enhance accuracy via prompt alignment, but this comes at the cost of substantial memory overhead, limiting deployment on memory-constrained edge devices. One notable exception is OS-Prompt [9], which collapses the two stages to reduce computation, but peak memory consumption during backward propagation remains high, thereby limiting scalability on memory-constrained devices.

A straightforward approach to reduce training-time memory in PCL is to adopt token-reduction techniques originally developed for ViTs. Existing token-reduction methods, such as Token Merging (ToMe) [1] and PatchDropout (PD) [14], reduce activation cost but discard task-relevant patches, thereby degrading accuracy when applied to PCL, as shown in Fig. 1. This motivates a task-aware sparsification approach for efficient edge training. To address this gap, we propose CPS-Prompt, a critical patch-aware sparse prompting framework that enables PCL to reduce memory usage and computational overhead while preserving accuracy on resource-constrained edge devices. This framework comprises two key modules designed to jointly address the aforementioned challenges: critical patch sampling (CPS) and decoupled prompt and classifier training (DPCT). CPS extracts task-specific signals from the final block of the frozen query forward pass to select critical patches before the prompt-injected forward pass, reducing stored activations and peak training memory. DPCT mitigates representation mismatch through a two-phase schedule: (1) joint optimization with sparse-patch inputs to learn task-adaptive features, and (2) classifier-only alignment with full-patch inputs while keeping the prompt frozen. This strategy reduces backpropagation overhead and shortens wall-clock training time.

Experimental results on three public benchmarks and real edge hardware show that CPS-Prompt improves peak memory, training time, and energy efficiency by about $1.6\times$ over CODA-Prompt. Meanwhile, CPS-Prompt maintains near state-of-the-art accuracy, with only a 2% average drop relative to C-Prompt. The main contributions of this work are as follows:

- We introduce CPS-Prompt, a PCL framework that improves training-time efficiency on resource-constrained edge devices by explicitly reducing both memory usage and computational cost with minimal accuracy loss.
- We design two complementary modules, CPS and DPCT, that jointly improve training-time efficiency in PCL by reducing memory usage and backpropagation overhead through task-aware patch selection and decoupled training.
- We validate CPS-Prompt on real edge hardware, the Jetson Orin Nano, confirming its robustness and effectiveness for realistic on-device continual learning scenarios.

2. Related Work

2.1. Prompt Continual Learning

CL primarily aims to mitigate catastrophic forgetting [15], a challenge rooted in the stability–plasticity dilemma [17], wherein a model must retain prior knowledge while acquiring new information. Traditional CL approaches include regularization-based methods [13, 33], architectural expansion [12, 31], and rehearsal-based strategies [2, 21]. However, these methods exhibit limited scalability: regularization-based approaches struggle to preserve discriminative representations across tasks [11], whereas expansion and rehearsal strategies incur growing memory or computational overhead as tasks increase [24].

PCL has emerged as an effective alternative by leveraging the strong transferability of pretrained ViTs to overcome these challenges [3]. By updating only lightweight prompts and keeping the backbone frozen, PCL reduces the number of learnable parameters while mitigating forgetting. L2P [29] introduces a prompt pool selected through a query mechanism, and DualPrompt [28] extends this strategy with both general and task-specific prompts. CODA-Prompt [22] improves end-to-end prompt optimization, whereas C-Prompt [5] leverages training–testing consistency to achieve state-of-the-art accuracy. However, these methods prioritize accuracy and often incur significant memory or computational overhead, limiting their deployability on resource-constrained edge devices. OS-Prompt [9] reduces computation by collapsing the two-stage design into a single stage, but peak memory usage during backpropagation remains high. In contrast, CPS-Prompt preserves the standard two-stage structure and introduces task-aware sparsity and decoupled optimization to improve both training-time memory and compute efficiency.

2.2. ViT Token Reduction

ViTs have achieved impressive performance across a wide range of visual tasks [3, 23], but their quadratic attention cost poses challenges for efficient deployment. Token-reduction methods address this by decreasing the number of tokens processed by self-attention. These methods fall into two categories: training-required approaches such as DynamicViT [20] and A-ViT [30], which use auxiliary modules to prune redundant tokens, and training-free approaches such as ToMe [1] and PatchDropout (PD) [14], which merge or drop tokens without additional training overhead. ToMe merges similar tokens across layers to reduce computational cost, whereas PD randomly drops input patches during training while keeping full tokens at inference. However, these methods operate in a task-agnostic manner, reducing tokens without considering their relevance to downstream representations. When combined with PCL, this often removes task-critical patches, leading to sig-

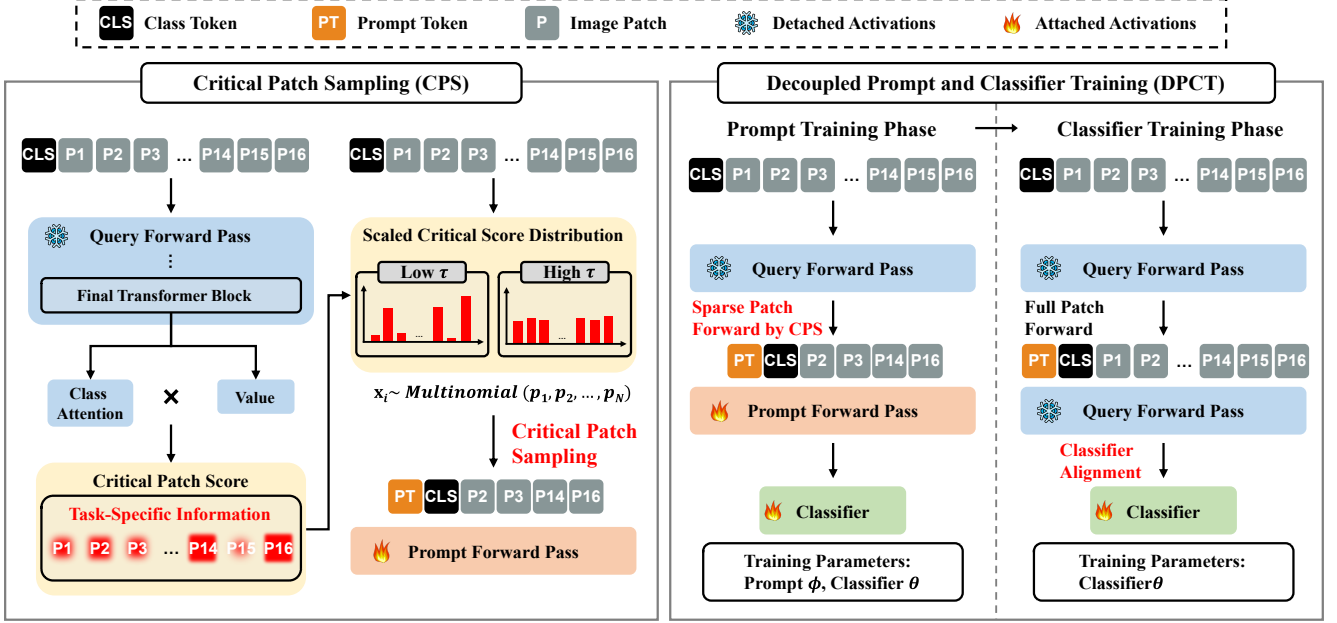


Figure 2. Overview of the CPS-Prompt framework. **Left:** CPS selects a small subset of task-relevant patches to preserve accuracy while reducing memory usage. **Right:** DPCT mitigates representation mismatch through decoupled training, where the prompt is optimized with sparse patches and the classifier with full patches.

nificant accuracy degradation under moderate-to-high sparsity. In contrast to prior token-reduction approaches, CPS-Prompt integrates token sparsity into PCL through a task-aware patch selection mechanism driven by attention and value activations from a frozen query encoder. This design improves both resource efficiency and representational stability under continual updates, enabling effective on-device continual learning under strict memory constraints.

3. Method

3.1. Preliminary

We consider a class-incremental learning setting in which a model is exposed to a sequence of tasks over time. Let the full dataset be denoted as $D = \{D_1, D_2, \dots, D_T\}$, where each D_i consists of image samples x and corresponding labels y , and T is the total number of tasks. Each task introduces a disjoint subset of classes, and data from previous tasks are not retained or revisited in later tasks. The objective is to learn each task sequentially while preserving knowledge of previously seen classes without access to past data.

PCL typically adopts a two-stage feed-forward architecture built on a shared pretrained backbone: a frozen query encoder f_q and a prompt-injected backbone f_p , where prompts are applied only in the second pass. This design leverages pretrained representations while enabling task-specific adaptation through learnable prompts. Formally, let

x denote an input image and ϕ denote the learnable prompt parameters. The frozen query encoder first produces a contextual representation $z_q = f_q(x)$, which is used to select appropriate prompts. The input is then forwarded to the prompt-injected backbone, yielding task-adaptive features $z = f_p(x; \phi)$. This two-stage formulation enables efficient reuse of pretrained knowledge while limiting the number of trainable parameters.

3.2. Critical Patch-aware Sparse Prompting

CPS-Prompt is a unified framework designed to improve training-time efficiency in PCL by integrating two modules: Critical Patch Sampling (CPS), which selects a compact set of informative patches using task-aware signals from the query encoder, and Decoupled Prompt and Classifier Training (DPCT), which mitigates representation mismatch and reduces computation. As illustrated in Fig. 2, CPS enables sparse and memory-efficient training, whereas DPCT improves both robustness and training efficiency through decoupled optimization.

3.2.1. Critical Patch Sampling

We introduce CPS, a lightweight module that selects a subset of task-relevant patch tokens to reduce memory usage and computational cost during prompt training. The key idea is to exploit the task-aware attention patterns of a frozen query encoder to estimate patch importance. In particular, we extract signals from the final transformer block, which captures the most task-relevant semantics as

Algorithm 1 Critical patch sampling

```

1: Input: image  $x$ , query encoder  $f_q$ , temperature  $\tau$ , patch reduction ratio  $r$ 
2: Let  $N$  be the number of patch tokens in  $x$ 
3: Extract  $A_{\text{cls},2:N+1}^L$  and  $\{V_j^L\}_{j=2}^{N+1}$  from the final layer  $L$  of  $f_q(x)$ 
4: Let  $\mathbf{a} = A_{\text{cls},2:N+1}^L$  and  $\boldsymbol{\nu} = [\|V_j^L\|_2]_{j=2}^{N+1}$ 
5: Compute  $\mathbf{s} = \mathbf{a} \odot \boldsymbol{\nu}$ 
6: Compute  $\mathbf{p} = \text{Softmax}(\mathbf{s}/\tau)$ 
7: Set the number of budget patches  $k = \lfloor (1-r) \cdot N \rfloor$ 
8: Sample indices  $\mathcal{I} \sim \text{Multinomial}(k; \mathbf{p})$  without replacement
9:  $\mathbf{X}_{\text{sampled}} \leftarrow [\mathbf{x}_{\text{cls}}] \cup \{\mathbf{x}_j \mid j \in \mathcal{I}_{\text{sampled}}\}$ 
10: return  $\mathbf{X}_{\text{sampled}}$ 

```

observed in prior PCL studies [22, 28]. Given an input image, the query encoder processes all tokens and the CPS module extracts the attention matrix $\mathbf{A}^L \in \mathbb{R}^{(N+1) \times (N+1)}$ and value matrix $\mathbf{V}^L \in \mathbb{R}^{(N+1) \times D}$ from the final transformer block L , where N is the number of patch tokens and D is the feature dimension. For multi-head attention, we sum the class-token-to-patch attentions across heads, and compute value norms over the full feature dimension D after head concatenation.

To identify patch tokens that most strongly contribute to the task-relevant class representation, we compute a critical score that reflects both attention relevance and feature strength for each patch token. Specifically, we use the class-to-patch attention weight in $\mathbf{A}^L$, where the class token is at index 1, and the L2 norm of its corresponding value vector:

$$s_j = A_{\text{cls},j}^L \cdot \|V_j^L\|_2, \quad j = 2, \dots, N+1 \quad (1)$$

This formulation captures both how strongly each patch influences the class representation (via attention) and how salient its features are (via the value norm). Following the token scoring in ATS [4], we adapt this scoring for training-free critical token selection using a frozen query encoder.

To control the trade-off between emphasizing highly task-relevant patches and introducing sampling diversity, we apply the temperature-scaled softmax to convert the critical scores into a sampling distribution:

$$p_j = \frac{\exp(s_j/\tau)}{\sum_{i=2}^{N+1} \exp(s_i/\tau)}, \quad j = 2, \dots, N+1 \quad (2)$$

Here, the temperature $\tau > 0$ controls the sharpness of the distribution; lower values produce a more peaked distribution that emphasizes the top-ranked patches based on task-specific cues from the query encoder, whereas higher values introduce greater sampling variability, which can help regularize training.

Algorithm 2 Decoupled prompt and classifier training

```

1: Input: dataset  $D$ , prompt-injected backbone  $f_p$ , epochs  $E$ , ratio  $\lambda$ 
2: Initialize: prompt parameters  $\phi$ , classifier parameters  $\theta$ 
3: for  $t = 1$  to  $\lfloor \lambda \cdot E \rfloor$  do  $\triangleright$  Prompt Training phase
4:    $\hat{y} \leftarrow f_p(\mathbf{X}_{\text{sampled}}; \theta, \phi)$   $\triangleright$  Sparse Patch Forward
5:    $\mathcal{L} \leftarrow \mathcal{L}_p(\hat{y}, y)$ 
6:    $\theta, \phi \leftarrow \text{Adam}(\nabla \mathcal{L}, \theta, \phi)$ 
7: end for
8: Freeze prompt parameters  $\phi$ 
9: for  $t = \lfloor \lambda \cdot E \rfloor + 1$  to  $E$  do  $\triangleright$  Classifier Training phase
10:   $\hat{y} \leftarrow f_p(\mathbf{X}_{\text{full}}; \theta, \phi)$   $\triangleright$  Full Patch Forward
11:   $\mathcal{L} \leftarrow \mathcal{L}_{\text{cls}}(\hat{y}, y)$ 
12:   $\theta \leftarrow \text{Adam}(\nabla \mathcal{L}, \theta)$ 
13: end for

```

Given a patch reduction ratio $r \in [0, 1]$, we compute the number of patch tokens to be retained as $k = \lfloor (1-r) \cdot N \rfloor$, and then sample k indices from the multinomial distribution parameterized by $\{p_j\}_{j=2}^{N+1}$:

$$\mathcal{I}_{\text{sampled}} \sim \text{Multinomial}(k; \{p_j\}_{j=2}^{N+1}) \quad (3)$$

We apply sampling without replacement to patch tokens after adding positional embeddings, ensuring k unique patches. The sampled patch tokens are then combined with the class token to form a sparse input sequence:

$$\mathbf{X}_{\text{sampled}} = [\mathbf{x}_{\text{cls}}] \cup \{\mathbf{x}_j \mid j \in \mathcal{I}_{\text{sampled}}\} \quad (4)$$

The sampled indices refer to the original token positions, with the class token at index 1 and patch tokens at indices 2 to $N+1$, preserving positional embeddings without renumbering. The patch indices are resampled at every mini-batch to promote stochastic exploration during training.

CPS significantly reduces memory and computational overhead during prompt training while preserving the semantic relevance of the input by forwarding only the most informative patches. In particular, because all critical scores are computed from a frozen backbone, the procedure is lightweight, training-free, and seamlessly integrated into existing PCL pipelines. A detailed description of the complete CPS procedure is provided in Algorithm 1.

3.2.2. Decoupled Prompt and Classifier Training

Although CPS enables memory-efficient training, it can lead to a representation mismatch between training and inference, particularly at high patch-reduction ratios. During training, the prompt-injected backbone is exposed only to a subset of patches, leading to feature representations that are misaligned with those derived from full inputs at inference time. To mitigate this, we propose DPCT, a decoupled

training strategy that separately optimizes the prompt and classifier to better align training with inference.

The DPCT comprises two sequential training phases. In the first phase, we jointly optimize the prompt parameters ϕ and classifier parameters θ using sparse patch inputs $\mathbf{X}_{\text{sampled}}$ selected by CPS. This phase focuses on learning task-adaptive representations from the reduced input and is trained using the standard cross-entropy loss:

$$\mathcal{L}_p = \mathcal{L}(f_p(\mathbf{X}_{\text{sampled}}; \theta, \phi), y) \quad (5)$$

In the second phase, we freeze the prompt parameters ϕ and fine-tune only the classifier θ using the full patch inputs $\mathbf{X}_{\text{full}}$. This step aligns the classifier with the representations it encounters at inference time, thereby mitigating the mismatch introduced by sparse training:

$$\mathcal{L}_{\text{cls}} = \mathcal{L}(f_p(\mathbf{X}_{\text{full}}; \theta, \phi), y), \quad (\text{with } \phi \text{ frozen}) \quad (6)$$

Given a fixed training budget of E epochs, we allocate $\lfloor \lambda \cdot E \rfloor$ epochs to prompt training, denoted as E_p , and assign the remaining $E - E_p$ epochs to classifier fine-tuning. This input-aware decoupling not only improves the alignment between training and inference but also reduces memory and computational overhead because gradients are not propagated through the prompt during the second phase. The complete DPCT procedure is summarized in Algorithm 2.

4. Experiments

4.1. Experiment Settings

Datasets and Metrics. We evaluate the proposed method on three widely used benchmarks for class incremental learning: CIFAR-100 [10], ImageNet-R [8], and CUB-200 [25]. Each dataset is partitioned into ten disjoint tasks. Following the standard protocol [21], we report the average accuracy (ACC_T) and forgetting (FGT_T) across all tasks. Additionally, we assess the training efficiency in terms of GPU peak memory usage [32] and per-task training time and energy consumption [19]. Details of the evaluation metrics are provided in the supplementary material.

Comparing methods. We include representative prior CL methods for comparison. The upper bound is obtained via joint training across all tasks, and SGD denotes naive fine-tuning. LwF [13] and ER [2] represent regularization- and rehearsal-based approaches, respectively, whereas L2P [29], DualPrompt [28], CODA-Prompt [22], C-Prompt [5], and OS-Prompt++/OS-Prompt [9] are prompt-based methods. In addition, we compare the proposed approach with existing ViT token-reduction techniques, including ToMe [1] and PD [14]. Details of ToMe merge ratios and PD schedules are provided in the supplementary material.

Implementation details. We use ViT-Tiny/16 as the default backbone for edge-device deployment. All models are initialized with weights pretrained on ImageNet-21K and finetuned on ImageNet-1K. Following CODA-Prompt [22], we use the same prompt length and number of components. We use the Adam optimizer with a batch size of 16 and train for 50 epochs on ImageNet-R and 20 epochs on the other datasets. The learning rate follows a cosine decay schedule, starting at 0.001. The phase ratio λ and temperature τ are set to (0.4, 0.1), (0.2, 0.1), and (0.6, 0.1) for CIFAR-100, ImageNet-R, and CUB-200, respectively, based on validation. For comparison with prior CL methods, we fix the patch reduction ratio to 0.4 across datasets, as it provides a robust accuracy–efficiency trade-off. All experiments are conducted using PyTorch on an RTX 4090 GPU, and the efficiency metrics are evaluated separately on a Jetson Orin Nano. The results are averaged over ten runs with different random seeds. Additional implementation details are provided in the supplementary material.

4.2. Comparison Results

Comparison with prior CL methods. We compare CPS-Prompt with representative CL methods on CIFAR-100, ImageNet-R, and CUB-200, with accuracy results summarized in Tab. 1 and efficiency results presented in Fig. 3. CPS-Prompt achieves competitive accuracy while significantly improving efficiency. While C-Prompt [5] achieves the highest average accuracy across datasets, it consumes about $4.3\times$ more memory, requires about $3.1\times$ longer training time, and uses about $3.3\times$ more energy than our method, making it impractical for memory-constrained edge devices. Compared to the balanced baseline CODA-Prompt [22], CPS-Prompt shows no statistically significant accuracy difference on CIFAR-100 and ImageNet-R while using about $1.6\times$ less memory, requiring about $1.5\times$ shorter training time, and consuming about $1.6\times$ less energy. Our method also outperforms the streamlined OS-Prompt, using about $1.6\times$ less memory, requiring about $1.1\times$ shorter training time, and consuming about $1.1\times$ less energy despite using a two-stage architecture. These results demonstrate that CPS-Prompt occupies a unique position in the accuracy–efficiency trade-off space, achieving near state-of-the-art accuracy, only 2% lower on average than C-Prompt, while providing substantial resource savings essential for edge deployment.

Comparison with token reduction methods.

Figure 4 and Figure 5 show comparisons of accuracy and training time with other token reduction methods under varying memory reduction ratios. As shown in Fig. 4, the proposed method consistently outperforms previous approaches across all datasets in terms of the trade-off between accuracy and memory usage. As the reduction ra-

Table 1. Comparison of CL methods on three datasets. ▼/△ indicate that the corresponding method is statistically worse/better than the proposed method, based on a paired t -test ($p < 0.05$). Higher is better for Accuracy (↑), while lower is better for Forgetting (↓). All results are averaged over 10 runs.

Method	CIFAR-100		ImageNet-R		CUB-200	
	ACC_T (↑)	FGT_T (↓)	ACC_T (↑)	FGT_T (↓)	ACC_T (↑)	FGT_T (↓)
Upper-Bound	83.73 ± 0.17	-	63.04 ± 0.63	-	75.38 ± 1.45	-
SGD	9.91 ± 0.29 ▼	94.70 ± 0.50 ▼	7.50 ± 0.09 ▼	76.56 ± 0.37 ▼	8.72 ± 0.34 ▼	80.41 ± 0.54 ▼
LwF [13]	10.98 ± 0.32 ▼	93.29 ± 0.38 ▼	7.64 ± 0.23 ▼	74.20 ± 0.42 ▼	10.26 ± 0.54 ▼	80.00 ± 0.45 ▼
ER [2]	38.83 ± 3.04 ▼	62.51 ± 3.37 ▼	19.57 ± 0.52 ▼	67.96 ± 0.69 ▼	46.29 ± 1.35 ▼	33.14 ± 1.08 ▼
L2P [29]	62.96 ± 0.98 ▼	15.93 ± 1.67 ▼	45.08 ± 0.45 ▼	8.28 ± 0.92 △	49.79 ± 1.14 ▼	11.68 ± 1.13
DualPrompt [28]	64.14 ± 0.53 ▼	16.50 ± 1.04 ▼	46.60 ± 0.44 ▼	11.42 ± 1.26	51.54 ± 0.99 ▼	10.77 ± 1.16
CODA-Prompt [22]	67.06 ± 0.51	14.73 ± 0.98 ▼	50.24 ± 0.58	13.50 ± 0.99 ▼	53.96 ± 0.53 △	11.46 ± 1.15
C-Prompt [5]	68.34 ± 0.84 △	16.48 ± 1.31 ▼	53.32 ± 0.36 △	13.52 ± 0.42 ▼	52.64 ± 1.06	11.46 ± 1.15
OS-Prompt++ [9]	66.44 ± 0.71	16.30 ± 0.81 ▼	50.30 ± 0.52	11.71 ± 1.40 ▼	52.63 ± 0.82	13.11 ± 1.21 ▼
OS-Prompt [9]	66.64 ± 0.71	16.66 ± 1.21 ▼	50.30 ± 0.29	14.49 ± 0.63 ▼	52.92 ± 1.14	13.21 ± 1.05 ▼
CPS-Prompt (Ours)	66.89 ± 0.59	13.15 ± 0.70	49.96 ± 0.56	11.06 ± 1.06	52.85 ± 0.74	11.28 ± 1.08

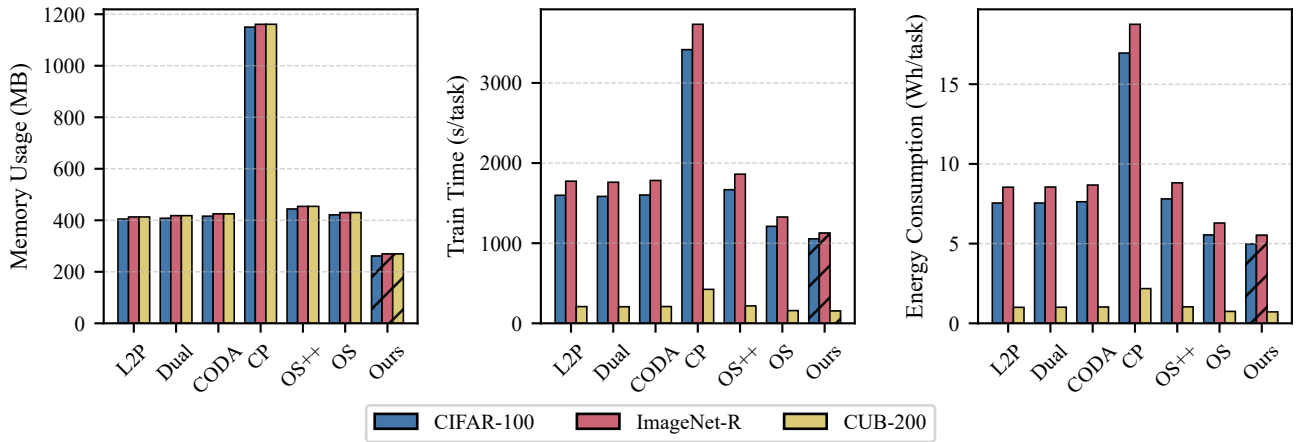


Figure 3. Comparison of memory usage, training time, and energy consumption between our method and other PCL methods on three datasets on the Jetson Orin Nano. For clarity, method names are abbreviated (e.g., CODA = CODA-Prompt, CP = C-Prompt, OS++ = OS-Prompt++, OS = OS-Prompt).

ratio increases, CPS-Prompt exhibits a gradual and controlled decrease in accuracy, maintaining robust performance even under aggressive reductions. In particular, even with a memory usage reduction of over 60%, our method retains over 90% of the baseline accuracy, demonstrating robust task-relevant patch selection. By contrast, ToMe suffers from a severe drop in accuracy once the memory usage reduction exceeds 40%, exhibiting poor stability under high sparsity. PD delivers moderate accuracy across the range but falls behind CPS-Prompt in both robustness and overall performance. These results highlight the superior robustness and efficiency of CPS-Prompt under varying levels of token sparsity.

As shown in Fig. 5, CPS-Prompt achieves the lowest training time across all memory reduction levels. This ad-

vantage stems from the combination of image patch-level token reduction and the decoupled training strategy. In contrast, ToMe initially reduces training time but incurs increased computational overhead beyond 40% memory reduction, owing to token-similarity computations at each transformer layer. While each operation is lightweight, these computations accumulate and become non-negligible on edge devices, such as the Jetson Orin Nano, increasing training time under aggressive sparsity. PD also maintains lower training time than the baseline, as its patch-reduction mechanism introduces negligible computational cost. However, PD consistently shows higher training time than CPS-Prompt across all reduction levels, reflecting the limited efficiency gains achievable without decoupled optimization. These results highlight that CPS-Prompt delivers superior

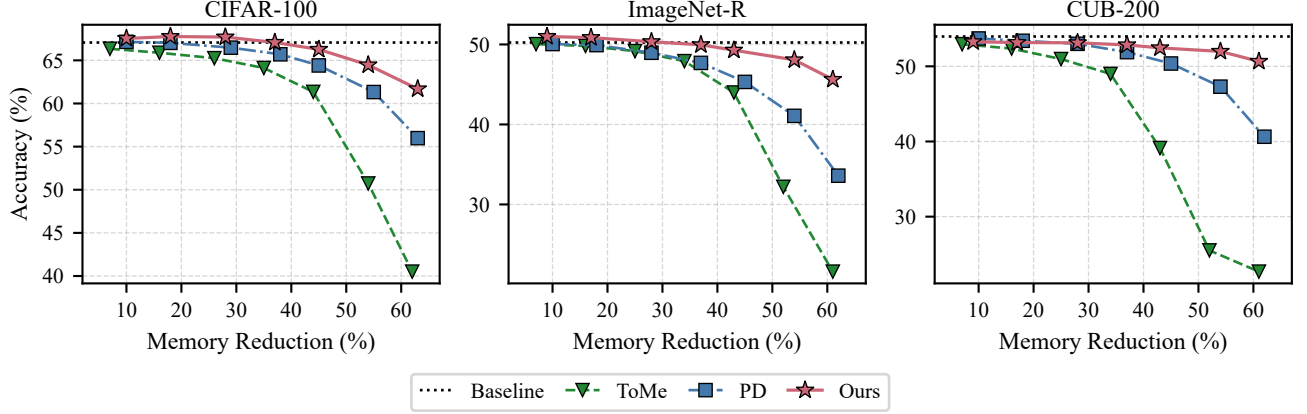


Figure 4. Comparison of accuracy and memory usage between CPS-Prompt and other token reduction methods based on the CODA-Prompt baseline under varying reduction ratios on the Jetson Orin Nano.

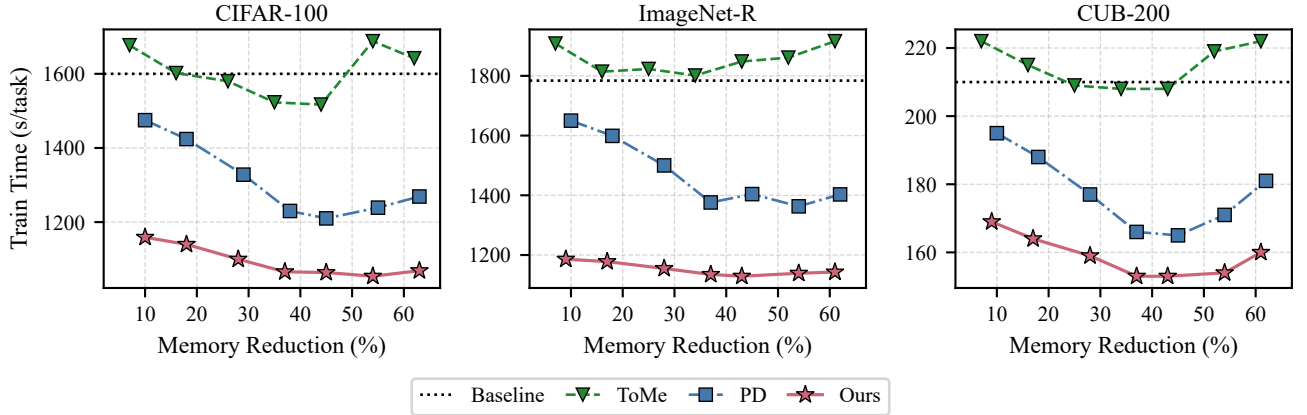


Figure 5. Comparison of training time and memory usage between CPS-Prompt and other token reduction methods based on the CODA-Prompt baseline under varying reduction ratios on the Jetson Orin Nano.

Table 2. Ablation results on ImageNet-R at a reduction ratio of 0.5. We compare combinations of CPS and DPCT, using PD as the baseline for CPS.

Modules	ACC_N ($\uparrow$)	Memory ($\downarrow$)	Train Time ($\downarrow$)
CODA-Prompt	50.24	440 MB	1,788 s/task
w/ PD	45.32	253 MB	1,388 s/task
w/ CPS	47.16	253 MB	1,389 s/task
w/ PD + DPCT	47.96	253 MB	1,126 s/task
w/ CPS + DPCT	49.28	253 MB	1,126 s/task

computational efficiency and is better suited for deployment in resource-constrained edge environments.

4.3. Ablation Study and Analysis

Effect of the proposed modules. We conduct an ablation study on the ImageNet-R dataset with a reduction ratio of

0.5, halving the number of input patches. We select this setting because it represents a critical point at which the representational capacity of the input is significantly reduced, making it suitable for assessing the individual contributions of CPS and DPCT. Table 2 summarizes the performance of different configurations. PD is used as a baseline random-patch selection method and as a comparison point for CPS. We apply DPCT to both the PD and CPS variants to isolate its effect. The results demonstrate that CPS consistently outperforms PD in terms of accuracy, with no additional memory usage and only a marginal training-time overhead. This highlights the efficiency of CPS, which effectively utilizes task-specific signals from the query stage in the two-stage PCL structure to guide patch selection without incurring additional computational overhead. In addition, DPCT recovers approximately 2% accuracy in both PD and CPS settings by mitigating the representation mismatch in-

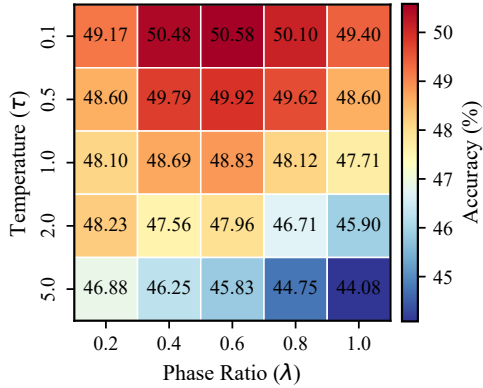


Figure 6. Effect of temperature and phase ratio on CUB-200. Accuracy is averaged over reduction ratios of 0.2, 0.4, 0.6, and 0.8.

troduced by token reduction. At the same time, freezing the prompt parameters during classifier fine-tuning significantly reduces training time. These results demonstrate that CPS and DPCT provide complementary benefits, improving both accuracy and efficiency under sparse training conditions.

Effect of Temperature and Phase Ratio. Figure 6 presents the accuracy on CUB-200 with respect to temperature τ and phase ratio λ . The temperature controls the stochasticity of CPS sampling, while the phase ratio determines the portion of epochs used for prompt training under sparse inputs. A lower temperature produces a more focused sampling distribution, favoring high-confidence patches and yielding higher accuracy, whereas higher values cause excessive randomness and degrade performance. Moderate phase ratios ($\lambda \in [0.4, 0.6]$) provide the best trade-off by balancing prompt learning and classifier alignment, highlighting that controlled stochasticity with a balanced training schedule yields the most effective configuration.

Stochastic vs. Deterministic Sampling. We compare multinomial sampling (stochastic) at the optimal temperature of 0.1 with deterministic top- k selection over $\mathbf{p}$ across different phase ratios. Results in Fig. 7 show that stochastic sampling achieves the best overall accuracy and performs particularly better at lower phase ratios. Figure 8 visualizes this effect on the CUB-200 dataset. From top to bottom, the images increase in visual complexity; the second column illustrates deterministic top- k patch selection, and subsequent columns show stochastic CPS sampling with different temperatures. Top- k deterministically selects patches driven by the prior knowledge of the pretrained backbone, yielding identical selections for repeated inputs. In contrast, CPS employs controlled stochasticity guided by patch significance, selecting mainly object-relevant yet more diverse patches. Such stochastic exploration enhances generaliza-

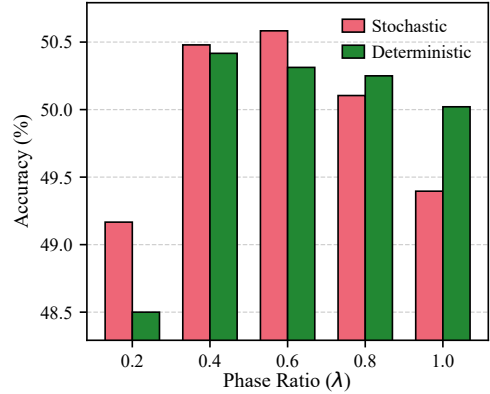


Figure 7. Comparison between stochastic and deterministic sampling on CUB-200. Accuracy is averaged over reduction ratios of 0.2, 0.4, 0.6, and 0.8.

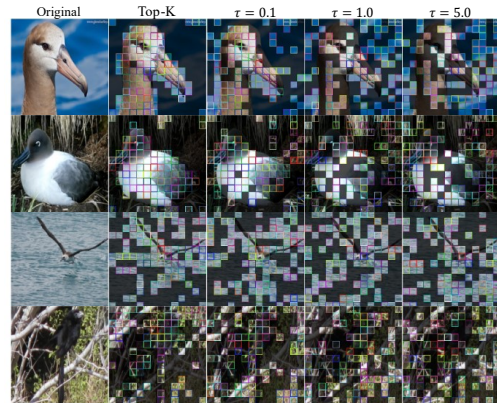


Figure 8. Qualitative comparison between deterministic top- k and stochastic sampling at a reduction ratio of 0.5 on CUB-200.

tion to novel or complex images, consistent with our probabilistic sampling design.

5. Conclusion

In this paper, we propose CPS-Prompt, a resource-efficient prompt-based continual learning framework that addresses the memory and computation bottlenecks of on-device training. CPS reduces stored activations by selecting task-relevant patches, and DPCT further improves alignment and training-time efficiency via input-aware decoupled optimization. Experiments across multiple datasets demonstrate that CPS-Prompt offers a well-balanced trade-off between accuracy and training-time efficiency on memory-constrained edge hardware. Future work will explore CPS-Prompt under dynamic resource settings and broader continual learning scenarios. Overall, our findings highlight task-aware token sparsity as a foundation for scalable, resource-efficient continual learning.

Acknowledgments

This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2021-II211341, Artificial Intelligence Graduate School Program (Chung-Ang University)) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2023R1A2C1006745).

References

- [1] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023. 2, 5
- [2] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc’Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019. 2, 5, 6
- [3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 1, 2
- [4] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *European Conference on Computer Vision*, pages 396–414, Cham, 2022. Springer Nature Switzerland. 4
- [5] Zhanxin Gao, Jun Cen, and Xiaobin Chang. Consistent prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28463–28473, 2024. 2, 5, 6
- [6] Yasir Ghunaim, Adel Bibi, Kumail Alhamoud, Motasem Alfarra, Hasan Abed Al Kader Hammoud, Ameya Prabhu, Philip H.S. Torr, and Bernard Ghanem. Real-time evaluation in online continual learning: A new hope. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11888–11897, 2023. 1
- [7] Tyler L. Hayes and Christopher Kanan. Online continual learning for embedded devices. In *Conference on Lifelong Learning Agents*, 2022. 1
- [8] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021. 5
- [9] Youngeun Kim, Yuhang Li, and Priyadarshini Panda. One-stage prompt-based continual learning. In *European Conference on Computer Vision*, pages 163–179, Cham, 2024. Springer Nature Switzerland. 2, 5, 6
- [10] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009. 5
- [11] Timothée Lesort, Andrei Stoian, and David Filliat. Regularization shortcomings for continual learning. *arXiv preprint arXiv:1912.03049*, 2019. 2
- [12] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3925–3934. PMLR, 2019. 2
- [13] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018. 2, 5, 6
- [14] Yue Liu, Christos Matsoukas, Fredrik Strand, Hossein Azizpour, and Kevin Smith. Patchdropout: Economizing vision transformers using patch dropout. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3953–3962, 2023. 2, 5
- [15] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, pages 109–165. Elsevier, 1989. 2
- [16] Martin Menabue, Emanuele Frascaoli, Matteo Boschini, Enver Sangineto, Lorenzo Bonicelli, Angelo Porrello, and Simone Calderara. Semantic residual prompts for continual learning. In *European Conference on Computer Vision*, pages 1–18, Cham, 2024. Springer Nature Switzerland. 2
- [17] Martial Mermillod, Aurélie Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in psychology*, 4:504, 2013. 2
- [18] Lorenzo Pellegrini, Vincenzo Lomonaco, Gabriele Graffieti, and Davide Maltoni. Continual learning at the edge: Real-time training on smartphone devices. In *29th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2021, Bruges, Belgium, October 6-8, 2021*, pages 23–28, 2021. 1
- [19] Ameya Prabhu, Hasan Abed Al Kader Hammoud, Puneet K. Dokania, Philip H.S. Torr, Ser-Nam Lim, Bernard Ghanem, and Adel Bibi. Computationally budgeted continual learning: What does matter? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3698–3707, 2023. 5
- [20] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, pages 13937–13949. Curran Associates, Inc., 2021. 2
- [21] 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*, pages 2001–2010, 2017. 2, 5
- [22] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11909–11919, 2023. [1](#), [2](#), [4](#), [5](#), [6](#)
- [23] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357, 2021. [2](#)
- [24] Eli Verwimp, Matthias De Lange, and Tinne Tuytelaars. Rehearsal revealed: The limits and merits of revisiting samples in continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9385–9394, 2021. [2](#)
- [25] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. [5](#)
- [26] Qipeng Wang, Mengwei Xu, Chao Jin, Xinran Dong, Jinliang Yuan, Xin Jin, Gang Huang, Yunxin Liu, and Xuanzhe Liu. Melon: breaking the memory wall for resource-efficient on-device machine learning. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, pages 450–463, New York, NY, USA, 2022. Association for Computing Machinery. [1](#)
- [27] Zifeng Wang, Zheng Zhan, Yifan Gong, Geng Yuan, Wei Niu, Tong Jian, Bin Ren, Stratis Ioannidis, Yanzhi Wang, and Jennifer Dy. Sparcl: Sparse continual learning on the edge. In *Advances in Neural Information Processing Systems*, pages 20366–20380. Curran Associates, Inc., 2022. [1](#)
- [28] 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*, pages 631–648, Cham, 2022. Springer Nature Switzerland. [1](#), [2](#), [4](#), [5](#), [6](#)
- [29] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022. [1](#), [2](#), [5](#), [6](#)
- [30] Hongxu Yin, Arash Vahdat, Jose M. Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10809–10818, 2022. [2](#)
- [31] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018. [2](#)
- [32] Geng Yuan, Xiaolong Ma, Wei Niu, Zhengang Li, Zhenglun Kong, Ning Liu, Yifan Gong, Zheng Zhan, Chaoyang He, Qing Jin, et al. Mest: Accurate and fast memory-economic sparse training framework on the edge. *Advances in Neural Information Processing Systems*, 34:20838–20850, 2021. [5](#)
- [33] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3987–3995. PMLR, 2017. [2](#)
- [34] Yuqing Zhao, Divya Saxena, and Jiannong Cao. Memory-efficient domain incremental learning for internet of things. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pages 1175–1181, New York, NY, USA, 2023. Association for Computing Machinery. [1](#)