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

Supplementary Material

A. Additional Experimental Details

Datasets For our evaluation, we utilize three widely used benchmark datasets for class-incremental learning: CIFAR-100 ([3]), ImageNet-R ([2]), and CUB-200 ([7]). Each dataset adheres to the augmentation procedures listed in Table 1.

Evaluation Metrics To evaluate the performance of each algorithm, we use average accuracy and forgetting ([5]). Average accuracy measures overall performance across completed tasks, while average forgetting quantifies how much the model has forgotten about previously completed tasks. Average accuracy is defined as follows:

$$\boldsymbol{ACC}_i = \frac{1}{i}\sum_{j=1}^{i} a_{i,j} \tag{1}$$

And average forgetting is defined as follows:

$$\boldsymbol{FGT}_i = \frac{1}{i-1}\sum_{j=1}^{i-1} f_{i,j} \tag{2}$$

$$\text{where } f_{i,j} = \max_{l\in 1,\ldots,i-1} a_{l,j} - a_{i,j}$$

where i denotes the number of tasks seen during training, j indexes the test tasks, $a_{i,j}$ is the test accuracy on task j after training up to task i.

To evaluate the training efficiency of each algorithm, we use GPU peak memory usage and training time per task. GPU peak memory usage is measured using PyTorch via the function `torch.cuda.max_memory_allocated()`. Energy consumption is measured on the Jetson Orin Nano using `tegrastats`, which reports real-time power usage. For training time and energy consumption, we report measurements on task 1, repeated 10 times, ensuring a fair comparison across all methods under identical conditions. Running all tasks across 10 repetitions on the Jetson Orin Nano is prohibitively time-consuming. Task 1 provides a representative estimate of per-task workload because its data size matches the task-wise average for CIFAR-100 and CUB-200, and is well within the expected variation for ImageNet-R. Since every task is processed by the same backbone, prompt modules, and patch-selection pipeline, such small differences in data size lead to negligible runtime variation. Thus, task 1 serves as a valid and fair proxy for reporting training time and energy consumption.

Table 1. Image augmentation procedures used for each dataset.

Stage	CIFAR-100	ImageNet-R, CUB-200
Train	RandomResizedCrop(224) RandomHorizontalFlip(0.5) Normalize((0,0,0), (1,1,1))	RandomResizedCrop(224) RandomHorizontalFlip(0.5) Normalize((0,0,0), (1,1,1))
Test	Resize(224) Normalize((0,0,0), (1,1,1)) -	Resize(256) CenterCrop(224) Normalize((0,0,0), (1,1,1))

Table 2. Configurations of PCL methods used in implementation.

Method	Prompt pool size	Prompt length	Layer Index
L2P	30	(10)	1
DualPrompt	one per task	(5,5,20,20,20)	(1,2,3,4,5)
CODA-Prompt	100	(4,4,4,4,4)	(1,2,3,4,5)
C-Prompt	number of classes	50	(1, 6)
OS-Prompt++/OS-Prompt	100	(4,4,4,4,4)	(1,2,3,4,5)
Ours	100	(4,4,4,4,4)	(1,2,3,4,5)

Table 3. Configurations of token reduction methods used in implementation. *ToMe: number of tokens merged *per layer*. †PD and Ours: number of tokens reduced *from image patch tokens*.

Method	Reduction Ratios	Number of Reduction Tokens
ToMe	(0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7)	(6, 9, 12, 15, 19, 26, 38)*
PD	(0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7)	(19, 39, 58, 78, 98, 117, 137)†
Ours	(0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7)	(19, 39, 58, 78, 98, 117, 137)†

PCL and Token Reduction Methods We summarize the configurations of the compared methods, including both PCL and token reduction approaches. Table 2 presents the settings used for each PCL method, including the prompt pool size, prompt length, and the layer index where the prompt is injected. Table 3 summarizes the configurations of token reduction methods, including the reduction ratios and the corresponding number of reduced tokens. Note that for ToMe ([1]), the values indicate the number of tokens merged per layer, while for PD ([4]) and our method, the values represent the number of tokens reduced from the input patch tokens.

B. Additional Results

Extended hyperparameter sweep results Following CODA-Prompt ([6]), we performed a hyperparameter sweep using 20% of the training data as a validation set. Results of the hyperparameter search over three datasets are shown in Fig. 1. This process resulted in optimal phase ratios of 0.4, 0.2, and 0.6, and temperatures of 0.1, 0.1, and

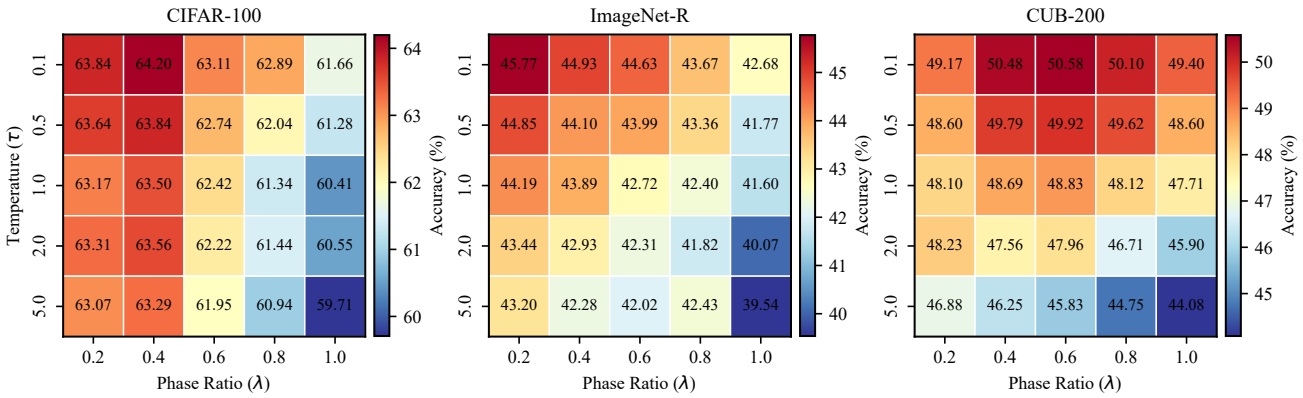

Figure 1. Effect of temperature and phase ratio on CIFAR-100 and ImageNet-R. Accuracy is averaged over reduction ratios of 0.2, 0.4, 0.6, and 0.8

0.1 for each dataset, respectively. The optimal phase ratios exhibit a consistent dataset-dependent trend. ImageNet-R, which is closest to the ImageNet pretraining distribution in terms of class space, favors a relatively small phase ratio (λ=0.2). CIFAR-100, a lower-resolution dataset with moderate domain shift, selects an intermediate ratio (λ=0.4). In contrast, CUB-200, which differs more substantially from the pretraining domain and contains fine-grained categories, benefits from a larger ratio (λ=0.6). This pattern suggests that datasets with greater domain or granularity mismatch may require a longer prompt-training phase before classifier alignment.

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. 1

[2] 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. 1

[3] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009. 1

[4] 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. 1

[5] 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. 1

[6] 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

[7] 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. 1