Is Prompt Selection Necessary for Task-Free Online Continual Learning?

Supplementary Material

A. Evaluation Metrics

In this section, we provide a detailed description of the evaluation metrics. Let $a_{t,i}$ denote the classification accuracy evaluated on the i-th task after training up to the t-th task, given a total of T tasks. After completing training on all T tasks, the last accuracy A_{last}, is defined as:

$$A_{\text{last}} = \frac{1}{T}\sum_{i=1}^{T} a_{T,i}. \tag{6}$$

Also, the forgetting at the last task, F_{last}, is defined as:

$$F_{\text{last}} = \frac{1}{T-1}\sum_{i=1}^{T-1} \max_{j\in\{1,\cdots,T-1\}} (a_{j,i} - a_{T,i}). \tag{7}$$

The A_{auc} [12] is designed to evaluate any time inference accuracy of the model. It is calculated every time the model has trained on n samples. Let L be the total number of evaluation steps, and a_l denotes the accuracy at the l-th evaluation step. We define:

$$A_{\text{AUC}} = \frac{1}{L}\sum_{l=1}^{L} a_l. \tag{8}$$

Note that each evaluation is performed on all the classes the model has learned so far.

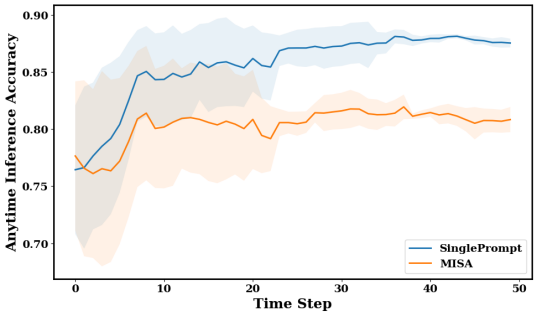


Figure 6. Anytime inference accuracy curves of SinglePrompt and MISA [10] on CIFAR100 [13]. Accuracy is measured every 1,000 training samples, and the curves visualize the mean and standard deviation over five random seeds.

B. Si-Blurry Scenario

To evaluate the effectiveness of our proposed framework in task-free online continual learning, we conducted experiments under the Stochastic incremental-Blurry (Si-Blurry) scenario [18] (see Figure 7). A total of C classes are randomly partitioned into disjoint and blurry classes according to a predefined disjoint class ratio. Each class subset is then randomly divided into T tasks. For the k-th task, we denote the disjoint and blurry datasets as T_k^D and T_k^B, respectively. Among the blurry class samples, a proportion determined by the blurry sample ratio is randomly shuffled across tasks, regardless of task boundaries. This process causes the boundaries between tasks to become ambiguous. As a result, the dataset for the k-th task $\mathcal{D}_k$ consists of both T_k^D and T_k^B.

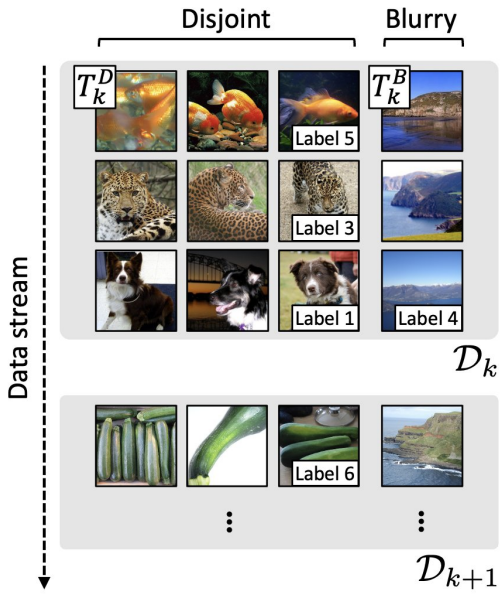


Figure 7. Visualization of the Si-blurry scenario. The dataset for the k-th task, $\mathcal{D}_k$, consists of the disjoint dataset T_k^D and the blurry dataset T_k^B.

C. Experimental Details

Pseudo-code for the proposed **SinglePrompt** is provided in Algorithm 1.

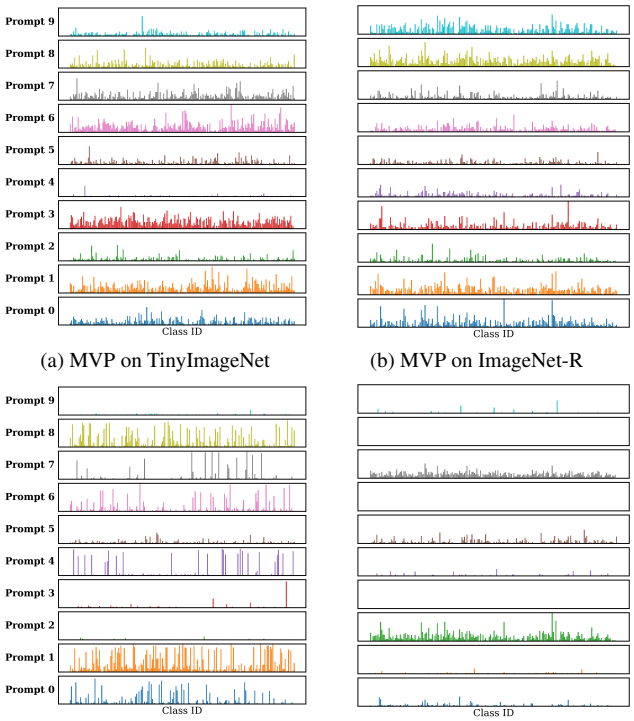
In-depth Analysis of Prompt Selection In Section 3.2, we present five types of prompt selection failures in continual learning. For L2P [22], DualPrompt [21], MVP [18], and MISA [10], we follow the original experimental setups described in their respective papers. While the main paper

Algorithm 1 SinglePrompt: Pytorch-like Pseudocode

Input: Input image x, label y
Parameter: Prompt injected layers K, self-attention blocks depth L, temperature τ, mask vector m
Output: Cross-entropy loss

$h_0 = f_0(x)$ // f_0: input embedding layer
for $i = 1$ **to** L **do**
  if $i \leq K$ **then**
    $h_i = f_i\big(h_{i-1}; p_i^k,\ p_i^v\big)$ // f_i: i-th self-attention block, $p_i^k,\ p_i^v$: prompt for block i (attention operation via equation (2))
  else
    $h_i = f_i(h_{i-1})$ // no prefix tuning (attention operation via equation (1))
  end if
end for
$\texttt{logits} = \frac{\mathbf{g}\cdot\mathbf{c}}{\|\mathbf{g}\|\|\mathbf{c}\|} \cdot \frac{1}{\tau} + m$ // $\mathbf{g}$: encoder output feature, $\mathbf{c}$: class prototypes
$\texttt{loss} = \texttt{CrossEntropy}(\texttt{logits},\ y)$
return loss

reports results on CIFAR100, we extend the analysis to additional datasets and observe that similar prompt selection failures consistently occur, as illustrated by the results of MVP and MISA in Figure 8. For ConvPrompt [20], we analyze the effectiveness of its prompt selection mechanism under both online and offline continual learning settings. In the offline setting, where the method is originally introduced, task boundaries are explicitly provided, and the similarity between the upcoming task and previously learned tasks is measured using GPT-3 generated class descriptors. Based on this task similarity, the number of additional prompts is dynamically determined, resulting in the final size of $P = 22$. In the online setting, however, the upcoming class information is not available. Consequently, the number of prompts allocated to each task is fixed to 5, leading to the final size of $P = 50$. ConvPrompt performs prompt selection through a soft routing mechanism, where the cosine similarity between the class token from the previous layer and a set of learned keys is used as routing weights. The prompts associated with each key are then combined into a weighted sum according to these similarity scores to produce the final prompt for prefix tuning. If the prompt selection mechanism operates as intended, each input sample should exhibit high similarity to the prompt key corresponding to its task. To examine this, we compute and compare the average similarity between input samples and the keys associated with each task's prompts. As shown in Figure 9, the average cosine similarity across all tasks is expected to be high, yet is observed to be close to zero. This holds for both the online (Figure 9b) and offline (Figure 9a) settings, indicating that soft routing mechanism fails



(a) MVP on TinyImageNet (b) MVP on ImageNet-R

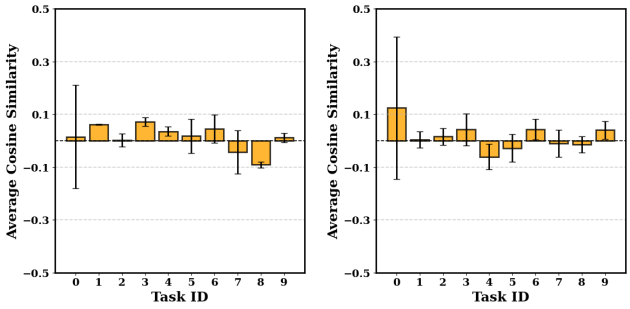
(c) MISA on TinyImageNet (d) MISA on ImageNet-R

Figure 8. Prompt selection failures on additional datasets, showing that the observed failure patterns generalize beyond CIFAR-100.

in task-based continual learning. Note that the result in the main paper is reported under the online setting.

Data preprocessing. We use CIFAR100 [13], Tiny ImageNet [14] and ImageNet-R [7] following prior works. Since we use the ViT-B/16, all input images are resized to 224×224. For each dataset, additional preprocessing was applied in the same manner as in previous studies [10, 18].

Computing Infrastructure. All experiments were conducted on a machine equipped with an NVIDIA GeForce RTX 4090 GPU, an Intel Xeon Gold 6526Y CPU, and 512 GB of RAM. The code was implemented using PyTorch 2.4.1+cu121, with CUDA 12.1 and cuDNN 9.0.1. Computational statistics are reported in Table 5.



(a) Offline Setup (b) Online Setup

Figure 9. Failure of ConvPrompt [20] selection in task-based continual learning on CIFAR100 [13]. The x-axis represents the task ID of input sample and the y-axis indicates the average cosine similarity between each task's samples and their assigned keys. (a) Result in the offline continual learning setting, where class information of upcoming tasks is available. Task similarity is computed using class descriptors, which determines the number of prompts allocated to each task (22 prompts in total). (b) Result in the online continual learning setting, where future task classes are unknown. Therefore, the number of prompts assigned to each task is fixed to 5, resulting in 50 prompts in total. In both cases, the cosine similarity is expected to be high for all tasks but remains close to zero. Results are shown for the 7-th layer. Note that consistent results are observed across other layers as well.

Method	#Params	FLOPs	Total time (s)
MVP[18]	$562K$	$34.21G$	2731.43
MISA[10]	$576K$	$35.67G$	3405.24
Online-LoRA[23]	$524K$	$17.64G$	1764.47
SinglePrompt	$230K$	$17.61G$	1589.74

Table 5. Computational statistics for ours on CIFAR100 [13] without buffer. We compare our method with MISA [10], MVP [18] and online-LoRA [23]. FLOPs refer to forward pass computations only, and total time is averaged over five runs with different random seeds. #Params refers to the number of learnable parameters. All methods use the same ViT-B/16 architecture to ensure a fair comparison.