Supplementary Materials for Online Class Incremental Learning on Stochastic Blurry Task Boundary via Mask and Visual Prompt Tuning

A. Details on the Compared Methods

In our experiments involving memory management, we utilized reservoir sampling as our method for memory management. We followed ER [10] to utilize memory in training, which combines half of the training batch from the streamed data with half of the training batch from memory. As online continual learning can not handle whole data in a task, other memory management methods such as herding selection [9] and mnemonics [7] are inapplicable. Also, memory management of Rainbow Memory [1] is inapplicable in online continual learning. Because they are based on the information of uncertainty from the whole task samples. Thus, we followed the rainbow memory training process from the CLIB [5].

LwF [6] is a classical method in continual learning which leverages knowledge distillation to prevent the model from catastrophic forgetting. LwF was introduced for offline learning. So, we modified the LwF to apply to online continual learning. Modified LwF distills the knowledge in every batch.

B. Additional Ablation Studies

We conducted ablation studies for the hyperparameter γ, m and α value used in gradient similarity-based focal loss, adaptive feature scaling, and total loss respectively.

B.1. Hyperparameters γ and m

Table 1 shows the result of the hyperparameter γ ablation study. Hyperparameter γ controls the ignore score calculated by a sample in gradient similarity-based focal loss. We set the γ value to 0.5, 1.0, 1.5, 2.0, 2.5 and evaluate the performance by A_{AUC} and A_{Last}. When the γ was 2.0, our method scored optimal performance in both A_{AUC} and A_{Last}. Also, we experimented with the performance variation by the hyperparameter m. Hyperparameter m is used in adaptive feature scaling to yield a marginal benefit score of a sample. Table 2 shows the result of the performance variation by the hyperparameter m. As we could see through Table 2, our novel method was robust to margin value. So, we set the m to 0.5 showing the highest performance among all seeds.

Method	γ	A_{AUC}	A_{Last}
Baseline	-	67.07±4.16	56.82±3.49
MVP (Ours)	0.5	67.25±5.08	60.39±1.55
	1.0	67.45±5.05	60.95±1.61
	1.5	67.52±5.11	61.05±1.37
	2.0	**68.10±4.91**	**62.59±2.38**
	2.5	67.62±5.17	61.11±1.55

Table 1. γ controls the loss value of gradient similarity-based focal lass. Underlined value denotes the used value for our method and the bold value represents the highest performance in the table.

Method	m	A_{AUC}	A_{Last}
Baseline	-	67.07±4.16	56.82±3.49
MVP (Ours)	0.1	67.20±4.72	58.82±1.27
	0.3	67.49±4.83	60.04±1.12
	0.5	**68.10±4.91**	**62.59±2.38**
	0.7	67.89±4.94	61.31±1.71
	0.9	67.29±4.84	61.81±0.47

Table 2. m is a margin value used in calculating the marginal benefit score by a sample. Underlined value denotes the used value for our method and the bold value represents the highest performance in the table.

α	**Memory = 0**		**Memory = 2,000**	
	A_{AUC}	A_{Last}	A_{AUC}	A_{Last}
-	40.11±1.27	29.24±4.63	49.00±2.06	37.96±0.34
0.1	40.38±1.67	31.63±3.39	52.13±0.14	50.50±3.11
0.3	40.52±1.59	31.81±3.66	52.14±0.28	50.51±2.76
0.5	**40.60±1.21**	**31.96±3.07**	**52.47±1.45**	**50.54±2.08**
0.7	40.53±1.01	31.56±2.05	52.28±2.34	50.43±1.53

Table 3. α is a balancing value in total losses. Underlined value denotes the used value for our method and the bold value represents the highest performance in the table.

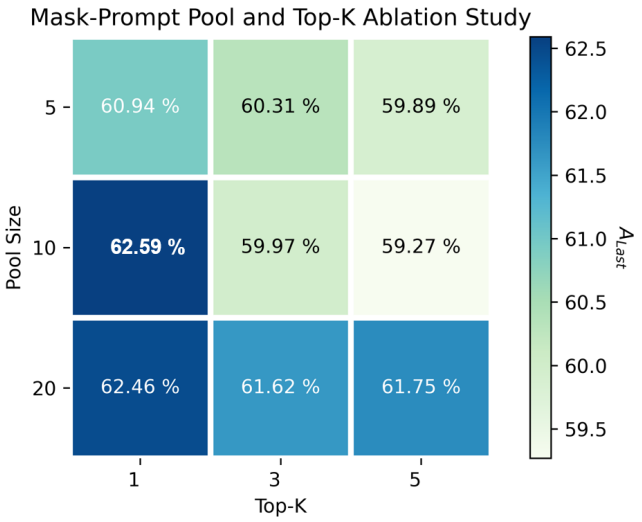


Figure 1. We set the prompt pool size and the number of selected prompts to 5, 10, and 20 and 1, 3, and 5 respectively. Top-K denotes the number of selected prompts.

Method	Forgetting
Baseline	46.61±5.30
+ GSF,AFS	45.35±4.40
+ Cont,Mask	39.98±4.02
+ Cont,Mask,GSF,AFS	**39.68±3.98**

Table 4. The ablation study on forgetting on ImageNet-R. The results demonstrate that our approach significantly mitigates forgetting and ensures better retention of previously learned knowledge.

B.2. Hyperparameter α

Table 3 presents the performance of our method for various values of α. Notably, the optimal performance is observed when an α value was 0.5, demonstrating consistent and robust results across different α values. Based on these findings, we set the α value to 0.5 as the most suitable choice for our experiments. This decision is grounded in the stability and high performance exhibited by the method at this particular alpha value, ensuring reliable and reproducible outcomes in our research.

B.3. Mask-Prompt Pool Size and Prompt Selection

Since the number of mask-prompt pairs has a large impact on our method, we ran experiments with a variety of mask-prompt pool sizes and prompt selection. Figure 1 represents the A_{Last} scores from the mask-prompt pool size and the number of selections. Top-K denotes the number of selected prompts. As shown in this figure, when the prompt pool size was fixed, a performance drop happened when more prompts were selected. Since selecting more masks and prompts induced much severe forgetting in each prompt, selecting a lot of masks and prompts exacerbated the performance. We set the mask-prompt pool size to 10 and the number of selection sizes to 1 to ensure the optimal performance of our method.

B.4. Forgetting

As shown in Table 4, we conducted experiments to assess the impact of each method on forgetting. Our findings revealed that GSF and AFS had limited effects on forgetting, as they predominantly targeted minor and major classes, respectively, in the class imbalance scenario. In contrast, our proposed approach, contrastive prompt tuning, demonstrated significant effectiveness in addressing the challenges of key floating and selection. Additionally, the utilization of masking proved to be highly effective in preventing forgetting by inhibiting the backpropagation of fully learned knowledge. These results collectively emphasize the robustness and efficiency of our method in effectively mitigating forgetting during the learning process.

C. Visualization of Masks and Prompt Keys

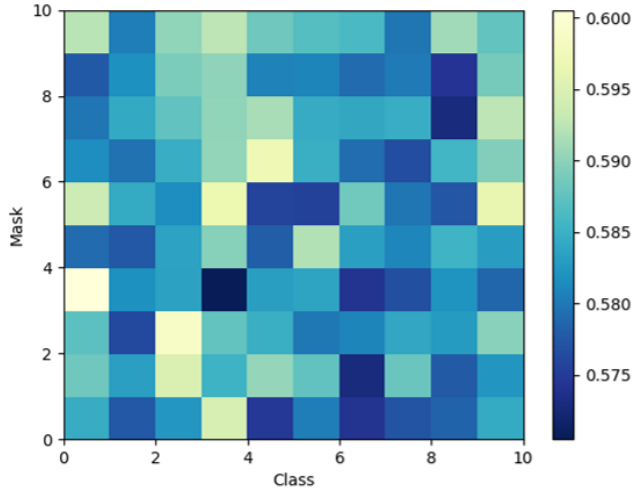
We performed visualizations to verify the suggested method experimentally and to understand our novel method MVP further. We visualized the mask and key of prompt methods.

C.1. Instance-wise Logit Mask

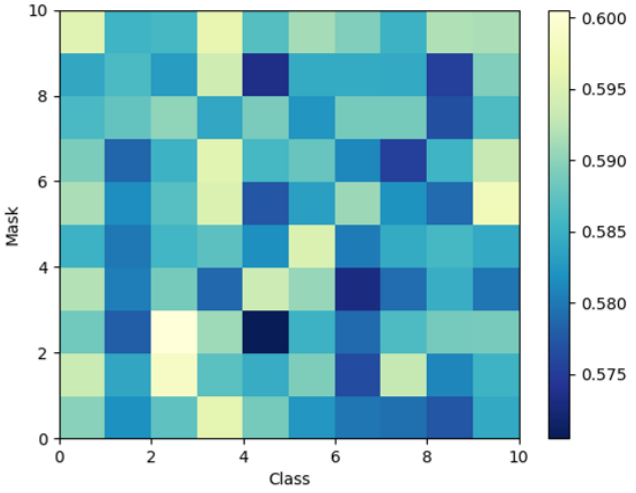
In order to validate the effectiveness of the mask used in our proposed method, we conducted a mask visualization experiment. The purpose of this experiment was to gain a better understanding of how the mask is utilized during the learning process. As Figure 2 illustrates, we could see that each mask opens for a different class. We could also see that on some parts of the masks, classes had their values decreased, preventing further updates. Also, some of them got increased value, allowing the model to learn relevant knowledge. The results demonstrated that the mask is effective in facilitating the division of tasks and protecting knowledge as intended. By facilitating the division of tasks and protecting knowledge, the mask enabled our method to perform well even in scenarios with blurry boundaries and multiple classes in a single batch. This is a significant contribution to the field of continuous learning and has important implications for real-world applications.

C.2. Prompt Key

Figure 3 shows the t-SNE visualization of the prompt key used in each prompt-based methods. Since DualPrompt [11] do not have any constraints on key learning in common,



(a) Visualization of each mask value from class 0 to 9 after training task 0



(b) Visualization of each mask value from class 0 to 9 after training task 4

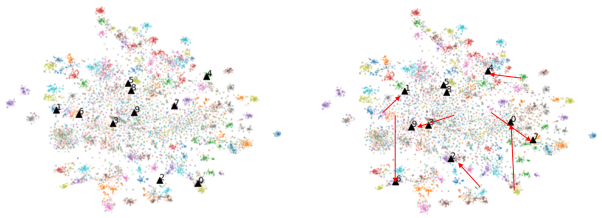
Figure 2. Visualization of mask value from class 0 to 9 (a) after task 0 (b) after task 4. Because each mask blocks logit from a different class, it seems to be noisy. It could be observed the value of the mask change as it trained.

we saw the keys floating as the task changes. This changes the function of each prompt in the feature space and could cause severe semantic drift. In MVP, we could see that the keys are kept at a reasonable distance from each other and the movement is suppressed once learning is sufficiently advanced.

D. Discussions

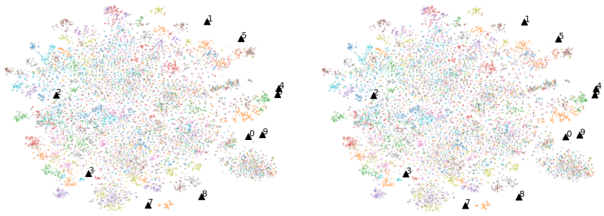
D.1. Additional Results for the Forgetting Score

Table 5 shows the performance of our proposed method with respect to the accuracy score and forgetting score. We used the forgetting measurement in [2] to report the forgetting results. As shown in this table, our method not only scored the highest accuracy in the table but also the lowest forgetting score. It denoted that MVP performs at best in accuracy while minimizing the forgetting knowledge.



(a) t-SNE Visualization of prompt key of DualPrompt [11] after training task 0

(b) t-SNE Visualization of prompt key of DualPrompt [11] after training task 4



(c) t-SNE Visualization of prompt key of MVP after training task 0

(d) t-SNE Visualization of prompt key of MVP after training task 4

Figure 3. t-SNE visualization of the prompt key of (a) DualPrompt [11] after task 0 (b) Dualprompt [11] after task 4 (c) MVP after task 0 (d) MVP after task 4. DualPrompt suffered from the semantic drift because the key is constantly changing as the task changes.

Memory Size	Methods	Metrics	
		A_{Last} (↑)	Forgetting (↓)
0	FineTuning	10.42±4.92	45.11±5.98
	LwF [6]	36.53±10.96	56.43±12.91
	L2P [12]	41.63±12.73	55.46±13.15
	DualPrompt [11]	56.82±3.49	40.35±1.25
	MVP (Ours)	**62.59±2.38**	**34.63±2.46**
500	ER [10]	60.68±1.15	28.85±3.51
	EWC++ [4]	25.62±3.35	47.16±9.72
	RM [1]	23.94±0.61	24.28±2.90
	CLIB [5]	67.16±0.72	15.45±0.94
	MVP (Ours)	**79.32±1.28**	**14.57±1.60**
2000	ER [10]	71.81±0.69	15.45±0.94
	EWC++ [4]	46.93±1.44	28.75±7.58
	RM [1]	65.51±0.55	9.50±1.49
	CLIB [5]	72.09±0.49	**8.07±0.98**
	MVP (Ours)	**84.42±0.44**	8.79±1.49

Table 5. We compared our method, MVP, to other existing methods in two metrics. Forgetting is measured with the best accuracy of each class and the inference accuracy after all the tasks are trained.

Method	Memory = 500		Memory = 2,000	
	A_{AUC}	A_{Last}	A_{AUC}	A_{Last}
L2P	69.91±1.49	56.58±0.64	75.24±0.82	68.73±0.80
DualPrompt	75.07±1.01	62.12±1.50	79.76±0.47	72.09±0.80
MVP-R (Ours)	**76.52±0.73**	**65.19±0.58**	**80.67±0.75**	**74.34±0.32**

Table 6. Comparison of ours with L2P and DualPrompt on Tiny ImageNet.

Method	TFLOPs	Training (s) /Iter
CLIB	69.6	11.590
DualPrompt	4.37	0.906
MVP (Ours)	**4.19**	**0.882**

Table 7. Computational cost Analysis of each method.

Note that low forgetting score do not mean a better method than others. If a model did not train with newly streamed data, there is no forgetting. However, reporting the low forgetting score while keeping the high prediction accuracy represents that the model can capture the knowledge from the new data while preventing the model from forgetting existing knowledge. Thus, forgetting measurement considering prediction accuracy is crucial to estimate the stability-plasticity of the method.

D.2. Additional Results with Memory

L2P and DualPrompt were initially not explicitly designed to incorporate memory, although they can be utilized in conjunction with memory. As shown in Table 6, we evaluated their performance in the presence of additional memory. Through extensive experiments conducted on the TinyImageNet Dataset, we observed that our method significantly surpassed DualPrompt and L2P. This compelling outcome confirms that the performance enhancement achieved by our method over the baseline is attributed to additional factors brought into play by memory utilization. These findings reinforce the effectiveness and advantages of our approach, particularly when memory is incorporated, leading to notable improvements in performance compared to the baseline methods.

D.3. Computational Cost

In Table 7, we conducted a thorough analysis of the computational cost associated with each method. This analysis encompassed a comparison of all methods using a memory capacity of 2000. Notably, the CLIB method necessitates forwarding for every individual sample to calculate the memory importance, resulting in a substantial computational overhead. In contrast, our method achieves a lower computational cost in comparison to DualPrompt (DP) by strategically reducing certain operations during the prompt selection process.

D.4. Task Configuration of Best and Worst Cases

We classified the classes into 3 categories: disjoint, major, and minor in the Si-Blurry scenario. Disjoint classes mean newly incomed classes that never appeared before. Since disjoint classes appear only once with all the training data, there is no overlap between tasks. Major classes and

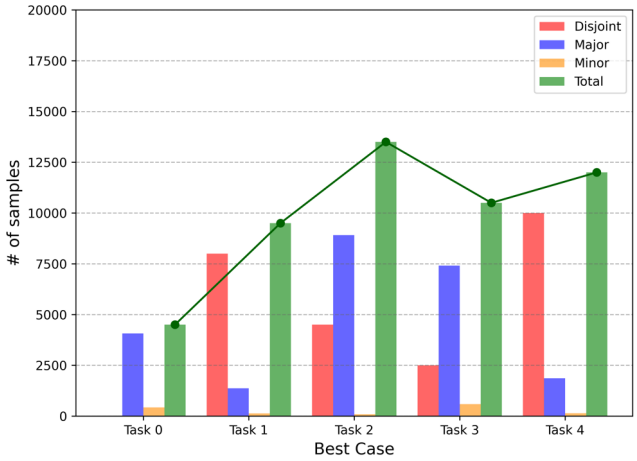


Figure 4. This figure represents the task configuration of training data in the best case. We reported the number of samples from each task. Total means the summation of training samples. We observed that training data are impartially distributed among the tasks in the best case.

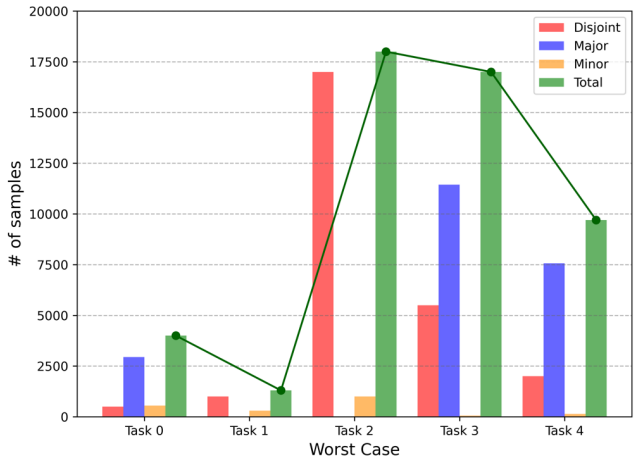


Figure 5. The above figure represents the task configuration of the worst case. Total means the summation of training samples. We observed that training data were concentrated on some tasks in the worst case.

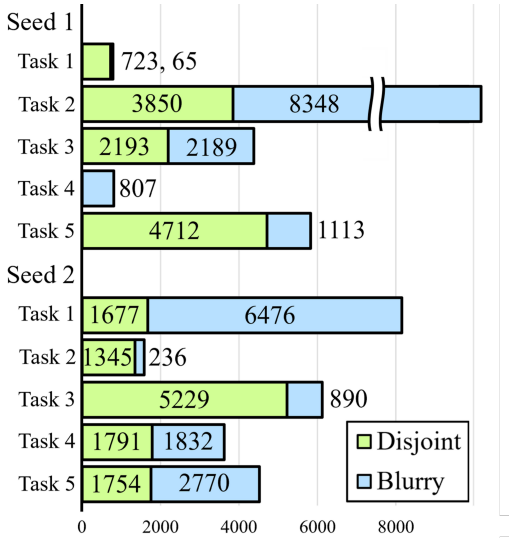


Figure 6. Example of Si-Blurry Scenario.

minor classes have a blurry task boundary. If a major class appeared in a task, that class turn to minor classes in other tasks. Hence, the major class can be overlapped between the tasks and once the major class appeared, it becomes the minor class in other tasks.

The Figure 6 shows another variety of possible Si-Blurry scenarios. Among these possibilities, we analysed the highest and lowest performing cases. Figure 4 and Figure 5 show the task configuration when our method scored the highest and lowest performance among the all seeds. It is noteworthy that the task configuration in the best case seemed like training samples are distributed impartially and in the worst case, training samples are concentrated on some tasks.

In other words, in the best case, the training data were impartially distributed to all tasks and it leaded to relatively low biased in tasks. The model could learn the knowledge among the tasks without severe weight drift or biased to some tasks. In the worst case, however, the training data were highly focused on some tasks and it leaded to a different amount of learning in the model training between tasks. In this case, the model could suffer severe catastrophic forgetting [8, 3] and be highly biased to the tasks with a lot of training samples. Our novel method MVP resolved this bias problem, showed better result than prior works.

References

[1] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *CVPR*, 2021. 1, 3

[2] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, 2018. 3

[3] Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999. 5

[4] 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. 3

[5] Hyunseo Koh, Dahyun Kim, Jung-Woo Ha, and Jonghyun Choi. Online continual learning on class incremental blurry task configuration with anytime inference. In *International Conference on Learning Representations*. 1, 3

[6] Zhizhong Li and Derek Hoiem. Learning without forgetting. *TPAMI*, 2017. 1, 3

[7] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *CVPR*, 2020. 1

[8] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press, 1989. 5

[9] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017. 1

[10] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. 2019. 1, 3

[11] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *ECCV*, 2022. 2, 3

[12] 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 *CVPR*, 2022. 3