A Appendix

In the appendix, we provide more details about the experimental results mentioned in the main paper, as well as empirical evaluations and discussions. Additionally, code for this paper is also provided in the supplementary materials. The appendix is organized as follows:

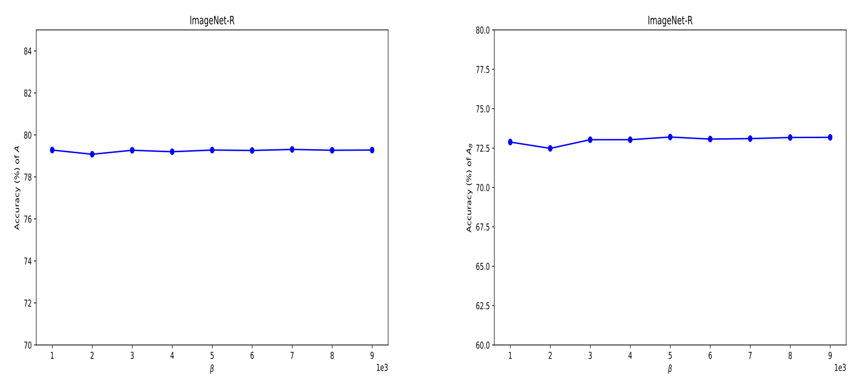
A.1 Further Implementation Details

A.1.1 Hyperparameters

For compared methods, we adopt the implementation[1] of L2P (Wang et al., 2022c) and DualPrompt (Wang et al., 2022b). We follow the implementations in ADAM[2] (Zhou et al., 2023) to re-implement other compared methods with ViT, i.e., ADAM-Finetune, ADAM-VPT-Shallow, ADAM-VPT-Deep, ADAM-SSF and ADAM-Adapter. Our PECTP is based on the ADAM-VPT-Deep and utilize the same hyperparameters in (Zhou et al., 2023). PRM in PECTP uses the hyperparameter α and β to maintain the balance between learning new task knowledge and preserving old task knowledge. We adopt the hyperparameters $\{\alpha = 1/3.5e5, \beta = 1/4e2\}$ for CIFAR100, $\{\alpha = 1/8e3, \beta = 1/5e2\}$ for CUB, $\{\alpha = 1/4.5e4, \beta = 1/5e2\}$ for ImageNet-R, $\{\alpha = 1/2e4, \beta = 1/5e2\}$ for ImageNet-A, $\{\alpha = 1/2e4, \beta = 1/2e2\}$ for ObjectNet, $\{\alpha = 1/1.5e4, \beta = 1/1e2\}$ for Omnibenchmark and $\{\alpha = 1/9e4, \beta = 1/2e2\}$ for VTAB. Following (Wang et al., 2022c), we use the same data augmentation for all methods, i.e., random resized crop and horizontal flip. Input images are resized to 224×224 before feeding into the model. Following (Rebuffi et al., 2017), all classes are randomly shuffled with Numpy random seed 1993 before splitting into incremental tasks.

A.1.2 Influence of Hyperparameters

In this section, we explore the influence of hyperparameters in PECTP. Specifically, since PECTP is optimized with with three parts losses, the additional hyperparameters compared to the ADAM-VPT-Deep are α and β. We train the PECTP on ImageNet-R INC10 with the PTM ViT-B/16-IN21K. We investigate the influence of the β in Figure 5. The results show that our PECTP's performance is robust with the change of β.



(a) (b)

Figure 5: The influence of the hyperparameter β of IPG with the accuracy of $\mathcal{A}$ (a) and $\mathcal{A}_B$ (b) on ImageNet-R.

A.2 T-SNE Visualization

In this section, we provide more T-SNE results of PECTP compared to Prompt-fixed methods. As described in Section 5.3 'Our Cross-Task Prompt vs. Key-Task Prompt', we have provided T-SNE visualization of PECTP and L-0P on task 1, task 3, task 4, task 7 and task 9 on CIFAR100. In this section, we provide T-SNE visualizatio on other tasks (as shown in Figure 6). Our PECTP performs better than L-0P on these tasks.

[1]https://github.com/google-research/l2p
[2]https://github.com/zhoudw-zdw/RevisitingCIL

A.3 ABLATION EXPERIMENTS OF THE FEATURE FUSION MODULE

In this section, we compare the performance between fused model $\mathcal{G}(f(x;\mathcal{P}), f(x))$ and its sub-models: VPT model $f(x;\mathcal{P})$ and SimpleCIL f (Zhou et al., 2023). The experiment results in the feature fusion module with different implementations: concatenation ($\mathcal{G} : Concat$) and point-wise addition ($\mathcal{G} : Pointadd$) are also provided in Table 4.

Fusion Method	CIFAR100		ImageNet-A	
	$\mathcal{A}$	$\mathcal{A}_B$	$\mathcal{A}$	$\mathcal{A}_B$
f	87.13	81.26	60.50	49.44
$f(x;\mathcal{P})$	91.91	87.04	63.48	50.36
$Pointadd(f(x;\mathcal{P}), f(x))$	92.59	87.94	66.21	55.43
$Concat(f(x;\mathcal{P}), f(x))$	92.4	87.86	66.29	55.63

Table 4: Influence of the feature fusion module.

A.4 IMPLEMENTATIONS OF THE CLASSIFICATION LAYER

In this section, we explore the experiment results in the classification layer with different initialization methods: augmentation section initialized with zeros (Zero-Init), override the previously learned classification layer (Override), augmentation section initialized with the previously learned classification layer (Old-Init), augmentation section initialized with U-distribution (U-Init) and augmentation section initialized with Kaming (Kaming-Init). The results are illustrated in Table 5.

Initialization Methods	only $\mathcal{L}_{cls}$		$\mathcal{L}_{cls}$ + PRM	
	$\mathcal{A}$	$\mathcal{A}_B$	$\mathcal{A}$	$\mathcal{A}_B$
Zero-Init	80.14	75.99	87.305	81.55
Override	89.93	85.65	92.32	87.58
Old-Init	89.73	85.34	92.275	87.5
U-Init	90.679	87.0	92.345	87.69
Kaming-Init	89.22	85.95	91.863	87.11

Table 5: Influence of different implementations in classification layer on CIFAR100.

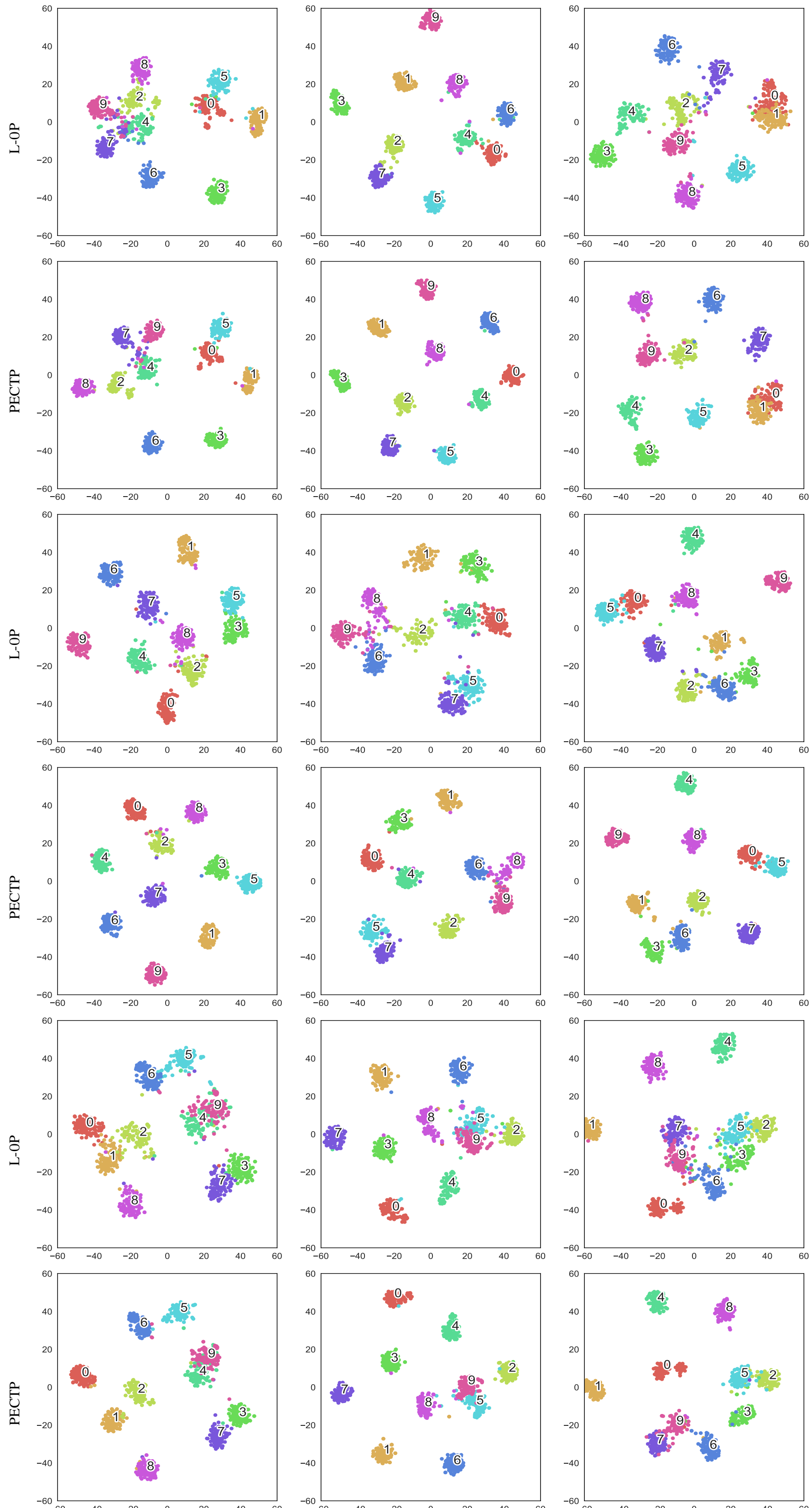


Figure 6: T-SNE visualization of features obtained by L-0P and PECTP on each incremental task on CIFAR100.