Self-Expansion of Pre-trained Models with Mixture of Adapters for Continual Learning

Supplementary Material

A. More Details about SEMA

A.1. More Details of SEMA Training

We discuss more details of SEMA training using a more detailed example in Fig. 9, which contains more details (*i.e.*, different types of cases and the distribution shift detection/scanning procedure) compared to that in Fig. 1. At the start of the training, each transformer block at different layers is equipped with one adapter module containing one adapter and one representation descriptor, as well as an expandable weighting router, as shown in Fig. 9 (b). They are added as the default adapters and trained on the first task. After the first task, for the incoming new tasks, SEMA monitors the representations of each batch of samples at each layer with the AE-based representation descriptor. As discussed in Sec. 3.6, the distribution shift is measured using the z-score computed from the mean and standard deviation of reconstruction errors stored in a buffer. This buffer is implemented as a fixed stack of 500 samples, maintaining reconstruction errors from the most recent batches. New adapters are added if a significant enough representation/distribution shift is detected at each layer. Adding the adapters expands the model's representation ability for handling the new patterns. As introduced in the main paper, SEMA performs task-oriented expansion (in the class-incremental learning setting given the task boundary in training), adding at most one adapter per layer. As shown in Fig. 1 and Fig. 9, the detection and expansion operation starts from the transformer layers closest to the input. Once a significant distribution shift is detected at a specific layer that could not be handled by *all* existing adapters (detected by RDs), an expansion signal is triggered in this layer/block. A new adapter module will be added to the layer where the expansion signal is triggered, along with an expansion of the weighting router, and activated for training. After sufficient training, the detection phase will be restarted for the later layers. If no distribution shift is reported for a task in any layers, as shown in Fig. 9 (c), no adapter module will be added, and no training of adapters is required for this task.

B. More Details about Implementation and Evaluation

B.1. Details of Datasets

CIFAR-100 contains 100 classes with 500 training samples and 100 testing samples per class.
ImageNet-R contains renditions of 200 ImageNet classes, which is a challenging CL benchmark introduced by with great intra-class diversity.
ImageNet-A contains real-world images filtered from ImageNet in an adversarial manner which are hard to be classified by models pre-trained with ImageNet.
VTAB consists of 50 classes from 5 domains with 10 classes from each domain.

To construct class-incremental setting, for results reported in Tab. 1, CIFAR-100, ImageNet-A and VTAB are split in a manner where each task consists of 10 distinct classes. ImageNet-R is reported with results for 5 tasks (40 classes per task), 10 tasks (20 classes per task), and 20 tasks (10 classes per task).

B.2. Implementations of Compared Methods

For SimpleCIL and ADAM, we use the official implementation at https://github.com/zhoudw-zdw/RevisitingCIL. For InfLoRA, we use the official implementation at https://github.com/liangyanshuo/InfLoRA. For other prompting methods, namely L2P, DualPrompt and CODA-P, we adopt the open-source implementation from PILOT toolbox [69], available at https://github.com/sun-hailong/LAMDA-PILOT. In our experiments, we adhere to the hyperparameter configurations as specified in the original publications for each of the compared methods, We use ViT-B/16-IN1K as the backbone with the same data shuffling as [90] for all methods.

B.3. Details on Evaluation Metrics

Denote the accuracy of the i-th task after training on the N-th task as $\mathcal{A}_{i,N}$. The average accuracy $\mathcal{A}_N$ represents the average accuracy of all seen tasks after training on the N-th task:

$$\mathcal{A}_N = \frac{1}{N}\sum_{i=1}^{N}\mathcal{A}_{i,N},$$

which is often considered as the most important evaluation metric in continual learning.

The average incremental accuracy $\bar{\mathcal{A}}$ is the average accuracy along incremental stages, defined as:

$$\bar{\mathcal{A}} = \frac{1}{N}\sum_{t=1}^{N}\mathcal{A}_t.$$

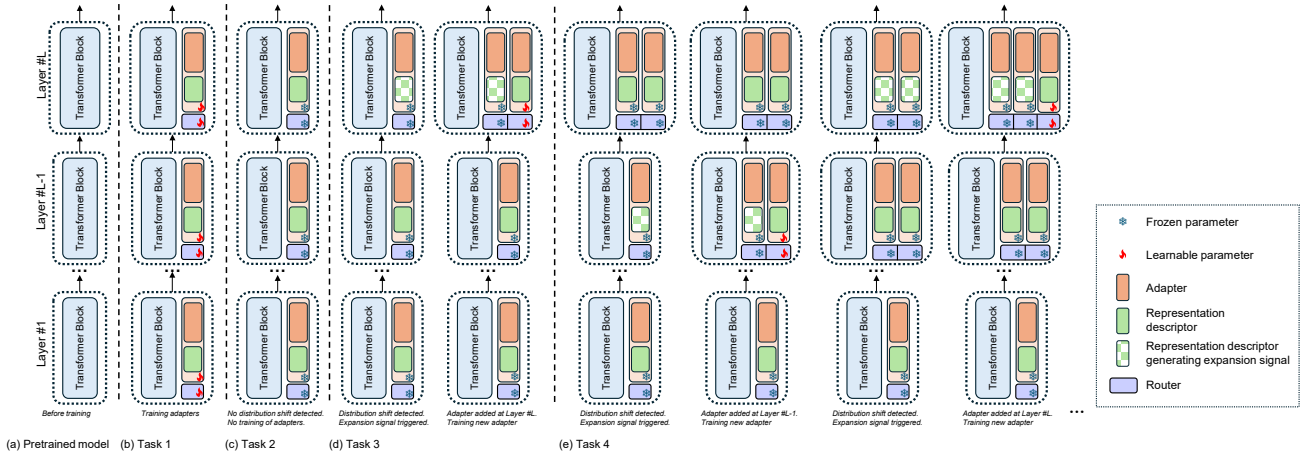


Figure 9. A more detailed example for the illustration of the learning process. (a) The pre-trained model with L transformer layers is provided for adaptation. (b) At the start of training, each transformer layer is equipped with one expandable weighting router and one adapter module, including one functional adapter and its paired representation descriptor. All modules are trainable at this stage. (c) All modules and routers are frozen after the training on Task 1. When Task 2 arrives, the detection of distribution shift is performed with all frozen representation descriptors in each transformer layer for all batches in Task 2. Since no distribution shift is observed, module addition is not performed and all modules are frozen. (d) As Task 3 arrives, the detection for the distribution shift is executed again and the distribution shift is observed in the L-th layer. Expansion signal is triggered and an adapter module is added in the L-th layer with the expanded router. Training for the newly added adapter and router is performed. Since the addition is performed at the last transformer layer, no further detection for distribution shift is required. (e) When Task 4 arrives, expansion signal is triggered in the $L-1$-th layer during the detection phase. After sufficient training, the newly added module is frozen and detection for distribution shift in later layers is executed. When both representation descriptors in the L-th layer consider the incoming feature as an outlier, expansion signal will be triggered. A new module is added for training in the L-th layer while all other modules are frozen.

Method	CIFAR-100		5-Task IN-R		10-Task IN-R		20-Task IN-R		ImageNet-A		VTAB	
	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$
L2P	89.51	85.02	72.90	65.83	74.55	69.75	74.49	65.82	46.67	39.30	79.17	63.56
DualPrompt	90.39	85.64	73.91	68.81	73.10	67.18	73.67	68.88	58.45	48.78	88.11	77.58
CODA-P	91.01	86.20	79.78	74.68	79.15	73.05	70.36	65.32	50.73	37.06	85.13	85.85
SimpleCIL	87.13	81.26	59.70	54.33	61.12	54.33	61.92	54.33	60.50	49.44	85.99	84.38
ADAM	92.18	87.47	77.28	70.58	76.71	69.18	75.08	67.30	60.53	49.57	85.95	84.35
InfLoRA	91.71	86.73	81.75	76.77	81.38	74.72	76.97	**69.65**	56.84	41.61	89.61	86.52
SEMA	**92.23**	**87.84**	**83.27**	**77.13**	**81.39**	**74.82**	**77.84**	69.60	**62.50**	**51.35**	**91.99**	**90.86**

Table 4. Experiments on class-incremental learning benchmarks with ViT-B/16-IN21K weight.

C. More Experiments and Ablation Studies

C.1. Influence of Pre-trained Weights

In the main paper, we experiment SEMA and other methods with ViT-B/16-IN1K in Tab. 1. To study the influence of pre-trained weights, we further experiment SEMA with another commonly used pre-trained ViT weight, i.e., ViT-B/16-IN21K. We evaluate the performance using average accuracy $\mathcal{A}_N$ and average incremental accuracy $\bar{\mathcal{A}}$. As shown in Tab. 4, SEMA consistently outperforms prompting and adaptation methods in most class-incremental learning settings. This indicates that our model is robust in performance regardless of different choices of pre-trained weights.

C.2. Further Analyses on the Effectiveness of Self-Expansion

The proposed method SEMA enables the model to add parameters and expand its capacity on demand. It allows the model to handle samples that could not be handled before by adding a small number of parameters. In continual learning, this process helps to alleviate forgetting by avoiding interference from new patterns while still encouraging knowledge reuse and transfer. Unlike some methods [68, 73, 92] that continually adding task-specific modules by task with a *lin-*

Dataset	Expansion by Task		SEMA	
	Params (M)	$\mathcal{A}_N$	Params (M)	$\mathcal{A}_N$
CIFAR-100	1.066	86.86	0.645	86.98
ImageNet-R	1.904	74.08	0.617	74.53
ImageNet-A	1.904	52.80	0.560	53.32
VTAB	0.647	89.09	0.554	89.64

Table 5. Comparison of added parameters and accuracy with different expansion strategies. "Expansion by Task" is a *naive* implementation of SEMA's variant that adds one set of adapters (at all layers allowing expansion) for every new task. SEMA only expands if a distribution shift is detected by the representation descriptor.

ear parameter growth rate, SEMA produces a *sub-linear* expansion rate, w.r.t. number of seen tasks. To analyze and show the effectiveness of this self-expansion process, we conducted comparisons on four different settings where CIFAR-100, ImageNet-R, ImageNet-A and VTAB contain 10 tasks, 20 tasks, 20 tasks and 5 tasks respectively, corresponding to four settings reported in Fig. 3. We compare with other related methods and a *naive implementation* of the "expansion-by-task" variant of SEMA. This simple variant model incrementally adds adapters to the layers that allow expansion for each incoming task. The number of parameters and accuracy are reported in Tab. 5. Despite the naive implementation of "expansion-by-task", the results in Tab. 5 show that SEMA with flexible self-expansion can achieve better performance than that using more parameters. We demonstrate that our expansion strategy is efficient in both controlling the size of added parameters, regardless of the length of task sequence, encouraging knowledge reuse and reducing potential task interference in adapter weighting.

Tab. 6 reports the size of added parameters in several different PTM-based methods. While L2P uses a fixed size of prompt pool with small amount of added parameters, the fixed size of trainable parameters may limit its capability to adapt to more distribution shifts in continual learning and comes with a higher chance of forgetting. Compared to other methods (*i.e.*, CODA-P and DualPrompt) that incrementally add parameters (*i.e.*, prompts in these methods) for each task, SEMA involves much fewer added parameters in the model. Apart from the adaptation approach and expansion strategy, the compared methods in this part use similar techniques as the proposed method (such as the classifier and PTMs). Note that the added parameters for SEMA only consider the functional adapters that are used in deployment. The RDs are maintained for training and updating of the model, which can be handled in parallel to other parameters and do not influence the deployment of the model. As shown in Fig. 10 (also demonstrated in the main paper Fig. 8), SEMA can dynamically expand the model with a small *sub-linear* rate, while the other methods are usually with a *linear* rate.

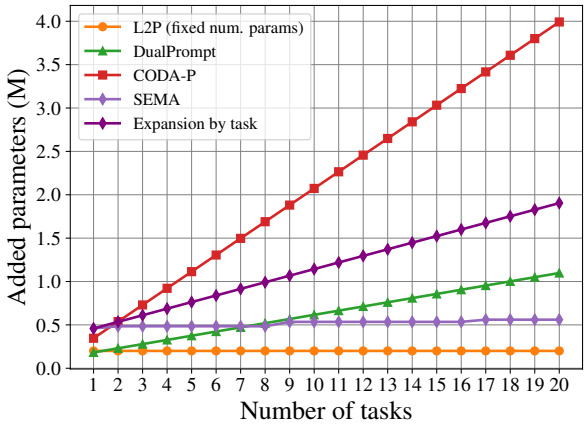


Figure 10. Analysis on added parameters (in Millions) during model deployment on ImageNet-A. We compare with methods using fixed number of prompts like L2P, and methods like DualPrompt and CODA-P that incrementally expand like SEMA but with prompts and on a linear basis according to tasks. Expansion by task adds adapters for every incoming task, whilst SEMA executes expansion on demand, which increments parameters on a sub-linear basis. Specifically, SEMA added more parameters (with expansions at more layers) at Task 9 than other steps with expansion.

C.3. Further Discussions on the Weighting Router

Routing relying on representation descriptor. In SEMA, we use the representation descriptors (RDs) to capture the distribution of the input representations corresponding to each modular adapter, which are used to detect novel patterns triggering the expansion signal. The RDs can be used to compose the adapters via hard selection, as in similar modular networks. Specifically, the reconstruction error of the AE-based RDs can provide the identity information of each inference sample, w.r.t. the adapters, at different layers. However, the RD-based adapter selection/routing can be unreliable for every single individual input, and related works usually rely on the statistics of a batch of samples [55], limiting the application. We thus propose directly learning the soft weighting router for mixture usage of the adapters. To analyze the behavior of the RDs in detail, we conduct the experiments that perform adapter composing relying on the RDs and show the results in Tab. 7. As shown in Tab. 7, the RD-based routing can achieve sound performance on most datasets, which validates the representation ability of RDs. SEMA with the soft weighting router can perform better, relying on the specifically learned router that is trained together with the adapters.

More discussions on adapter usage. Fig. 5 shows the average adapter usage of each task on VTAB. For clear visualization, we enable expansion to be performed only at the last layer and attach sample images from each task in Fig. 5. Adapter 1, Adapter 2, and Adapter 3 are automatically

Type	Method	CIFAR-100		ImageNet-R		ImageNet-A		VTAB	
		Params (M)	$\mathcal{A}_N$	Params (M)	$\mathcal{A}_N$	Params (M)	$\mathcal{A}_N$	Params (M)	$\mathcal{A}_N$
Fixed Param Size	L2P	0.123	77.87	0.200	62.90	0.200	38.48	0.085	80.83
Expandable Param Size	DualPrompt	1.022	80.43	1.098	61.97	1.098	50.23	0.983	79.79
	CODA-P	3.917	86.11	3.994	70.02	3.994	35.02	3.878	81.58
	SEMA	**0.645**	**86.98**	**0.617**	**74.53**	**0.560**	**53.32**	**0.554**	**89.64**

Table 6. Number of added parameters used in model deployment, measured in Millions. L2P uses a fixed size of prompts. DualPrompt and CODA-P incrementally add parameters (*i.e.*, prompts) sequentially by task. SEMA adds a small number of parameters with its dynamic expansion strategy.

Method	CIFAR-100		5-Task IN-R		10-Task IN-R		20-Task IN-R		ImageNet-A		VTAB	
	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$
SEMA	**91.37**	**86.98**	**84.75**	**79.78**	**83.56**	**78.00**	**81.75**	**74.53**	**64.53**	**53.32**	**91.26**	**89.64**
RD-based routing	90.91	83.61	84.46	79.50	82.76	76.63	81.02	74.13	61.80	50.36	90.83	88.53

Table 7. Comparison between routing with the expandable weighting router and RD-based routing.

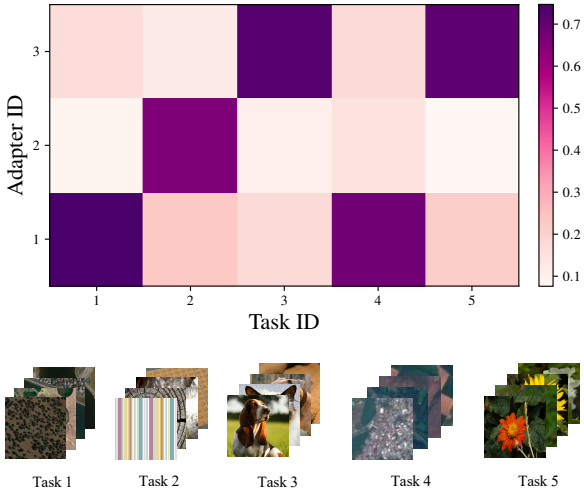


Figure 11. Adapter usage visualization on VTAB (same as Fig. 5). For clear and simplified visualization, we only allow expansion at the last transformer layer. We report the average adapter usage of each task. We also provide visual illustrations of sample images from each VTAB task.

Method	Train Time (s)			
	CIFAR-100	ImageNet-R	ImageNet-A	VTAB
L2P	0.27	0.27	0.29	0.28
DualPrompt	**0.25**	0.25	0.27	**0.29**
CODA-P	0.31	0.32	0.35	0.36
SEMA (Overall)	**0.25**	**0.11**	**0.15**	0.31
- Adapter	0.13	0.10	0.12	0.20
- RD	0.12	0.01	0.03	0.11

Table 8. Average per-batch train time of each method on each task measured in seconds. SEMA (overall) denotes the training time used when adapter and representation descriptor (RD) are trained sequentially.

Method	Inference Time (ms)			
	CIFAR-100	ImageNet-R	ImageNet-A	VTAB
L2P	9.44	9.53	9.86	9.46
DualPrompt	9.44	9.51	9.84	9.44
CODA-P	9.45	9.47	9.85	9.43
ADAM	9.95	10.03	10.36	9.45
SEMA	**4.48**	**7.39**	**9.01**	**7.38**

Table 9. Per-image inference time of each method measured in milliseconds.

added and trained when Task 1, Task 2, and Task 3 arrive, respectively. Task 1, Task 2, and Task 3 all present high preference for choosing the adapters that were trained with them, showing the effectiveness of the router to direct samples to the adapter that is trained with a similar distribution. While adapter expansion is not triggered for Task 4, Task 4 data largely employs Adapter 1 during inference. As visualized in Fig. 11, the data distribution between Task 1 (remote sensing images) and Task 4 (land cover) is similar. Similarly, Task 3 (pets) and Task 5 (flowers) both comprise natural images with similar characteristics, hence have higher similarity in distribution than Task 1 (remote sensing images) and Task 2 (texture images), and exhibit a preference for Adapter 3. Thus, we show that our expandable weighting router can effectively select the proper mixture pattern of adapters with various data distributions.

C.4. Training and Inference Time

All experiments can be produced on a single NVIDIA GeForce RTX 3090 GPU. To compare the training efficiency, we report the per-batch training time averaged over the incremental learning process in Tab. 8. Similar to Tab. 5, ImageNet-R here is split into 20 tasks with 10 classes per task. Note that the training processes of adapter and representation descriptor in each adapter module of SEMA are in parallel after expansion, thus the training of these two components can be performed in parallel with multiple GPUs. We report the training time of adapters (*i.e.*, "Adapter" in Tab. 8) and representation descriptors (*i.e.*, "RD" in Tab. 8) separately, along with the overall time usage of SEMA training if adapters and representation descriptors are trained sequentially.

SEMA with components trained in a parallel manner is highly efficient. Even without the parallel setup, training the adapters and RDs in SEMA in sequence can still be faster than other PTM-based CL methods on most datasets. As SEMA only expands while encountering distribution shifts in incoming new tasks, for tasks that do not trigger expansion, no training of adapters and representation descriptors is performed and training time on these tasks is minimized, leading to training efficiency in the long term. Note that the scanning for distribution shifts is stopped as long as a batch of data triggers expansion behaviour, which is more efficient comparing to InfLoRA which requires processing through all data in the given task twice for LoRA initialization before training and post-training computation for gradient projection memory.

We evaluate the inference efficiency and report the average inference time of each image measured in milliseconds in Tab. 9. We show that SEMA is efficient compared to other methods on all datasets. The inference latency of the listed prompting continual learning methods is caused by the extra procedure of processing the image with a frozen pre-trained model for the query function. Similarly, ADAM requires extra feature extraction with a frozen pre-trained model for the concatenation of pre-trained features and adapted features. SEMA relieves the dependency on the frozen pre-trained model as we focus on the intermediate feature distribution of each transformer block.

C.5. Additional Results on Longer Task Sequence

We perform the 50-step experiment on ImageNet-R and ImageNet-A, where each task contains 4 classes, and report the performance in Tab. 10. SEMA outperforms all other methods in longer task sequences.

Method	ImageNet-R		ImageNet-A	
	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$
L2P	69.11	63.53	40.77	33.31
DualPrompt	64.21	56.25	49.74	39.83
CODA-P	61.34	56.37	34.36	23.17
ADAM	69.59	62.58	59.44	48.58
InfLoRA	67.01	61.37	47.33	31.27
SEMA	**74.64**	**67.03**	**60.82**	**49.18**

Table 10. Evaluation on longer task sequence with 50 tasks.

C.6. Additional Results on Incremental Performance

We present a comparison of performance across incremental stages for CIFAR-100, 20-Task ImageNet-R, 20-Task

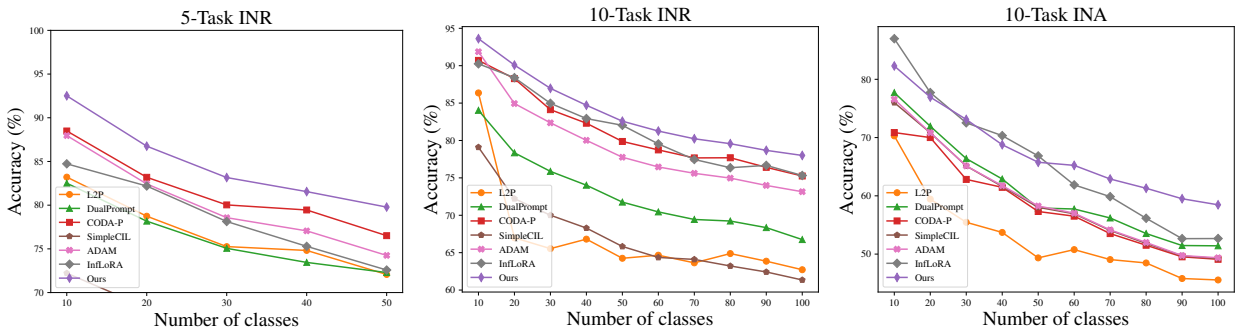


Figure 12. More results on incremental performance for ImageNet-R and ImageNet-A.

ImageNet-A and VTAB in Fig. 3 of the main paper. We further conduct experiments on ImageNet-A which is split into 10 tasks. We provide the incremental performance of 5-Task ImageNet-R, 10-Task ImageNet-R and 10-Task ImageNet-A in Fig. 12. Both figures show that SEMA performs consistently well with different dataset splits.

C.7. Analyses on Training with Less Data

We further conduct analyses on the scenario of training with less data. Benefiting from the better knowledge reuse/transfer ability, SEMA can achieve better performance with less data. We specifically compare with a state-of-the-art method, EASE [92], which expands task-specific adapters at all layers of the transformer. Unlike all other methods we compared with in the main paper, EASE also incrementally adds classification heads for all tasks and ensembles them in inference. In Tab. 11, we show the results of experiments on VTAB while removing 90% of samples in one and two tasks, respectively, denoted as VTAB-1 and VTAB-2. Although EASE uses a much stronger classification head, SEMA can perform better in this data efficiency learning experiment. We then further extend this data efficiency experiment to ImageNet-A by keeping only 10 or 20 percent of data for all tasks. As shown in Tab. 12, with sub-linear expansion, SEMA obtains performance comparable to EASE which requires task-oriented expansion at linear growth rate.

Method	VTAB-1		VTAB-2	
	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$
SEMA	86.74	81.33	85.99	80.06
EASE	86.56	78.37	86.76	78.86

Table 11. Experiments on setting with limited data samples on VTAB. VTAB-1 and VTAB-2 randomly removes 90 percent of data in one and two task(s), respectively.

Method	ImageNet-A 10%		ImageNet-A 20%	
	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$
SEMA	52.90	41.41	57.85	48.26
EASE	52.79	41.67	57.46	48.65

Table 12. Experiments on setting with limited data samples on ImageNet-A. ImageNet-A 10% contains only 10 percent of data in original ImageNet-A for all tasks and ImageNet-A 20% contains 20 percent.

C.8. Experimental Results with Different Seeds and Varying Class Orders

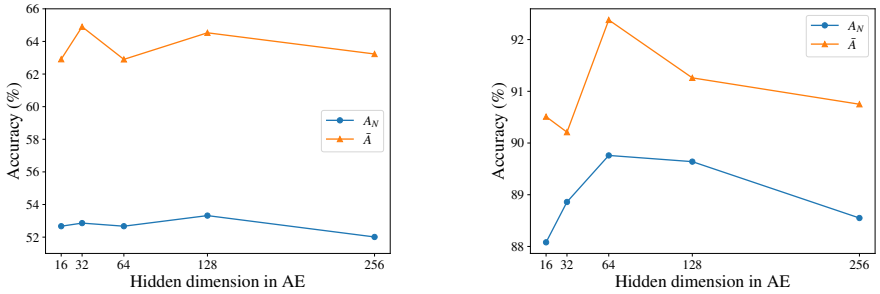
We conduct five independent runs with different seeds for SEMA on all datasets, and report the mean and standard deviation of accuracies over separate runs in Tab. 13. With different random seeds, each run is performed with different shuffling of class order and model initialization weights. This demonstrates the robustness of SEMA's performance with varying task/class orderings.

C.9. Ablation Study on the Hidden Dimension in AE

We test different values for hidden dimensions in the AE as representation descriptors. The AE-based representation descriptors enable the capture of the characteristics of the data for decision-making on whether to add a new adapter during continual training. According to Fig. 13, the proposed method can perform well with a wide range of settings on the AE's hidden dimension.

C.10. Results with Representation Enhancement

As discussed, different PTM-based continual learning methods focus on updating/adapting the backbone/representation (*e.g.*, SEMA, InfLoRA [47], CODA-P [68]) and continually conducting feature representation enhancement of frozen PTMs (*e.g.*, RanPAC [51]), respectively. These two types of methods are orthogonal and can work together. The pro-



(a) ImageNet-A (b) VTAB

Figure 13. Ablation on representation descriptor.

Method		CIFAR-100	5-Task IN-R	10-Task IN-R	20-Task IN-R	ImageNet-A	VTAB
SEMA	$\bar{\mathcal{A}}$	91.37 ± 0.38	84.75 ± 0.84	83.56 ± 0.41	81.75 ± 1.00	64.53 ± 0.99	91.26 ± 0.47
	$\mathcal{A}_N$	86.98 ± 0.57	79.78 ± 0.46	78.00 ± 0.49	74.53 ± 0.92	53.32 ± 0.69	89.64 ± 0.63

Table 13. Accuracies with standard deviation over 5 independent runs.

Method	CIFAR-100		5-Task IN-R		10-Task IN-R		20-Task IN-R		ImageNet-A		VTAB	
	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$
RanPAC	93.81	90.04	83.81	79.57	84.23	79.00	83.87	78.18	69.96	62.15	91.97	91.33
SEMA+RanPAC	**94.54**	**90.95**	**85.93**	**81.58**	**85.59**	**80.55**	**85.13**	**79.40**	**71.87**	**63.33**	**93.99**	**92.33**

Table 14. Results on different methods using random projection technique.

posed self-expansion learning in SEMA can also be combined with the statistical alignment techniques of RanPAC, *i.e.*, SEMA+RanPAC, to get better performance. Specifically, the feature enhancement with random projection and prototype classifiers in RanPAC is applied to the representations from SEMA's model. Tab. 14 demonstrates that the representations are benefited from the self-expansion strategy, as SEMA+RanPAC outperforms RanPAC implemented with a single adapter and first-session adaptation.

Method	CIFAR-100		10-Task IN-R	
	$\bar{\mathcal{A}}$	$\mathcal{A}_N$	$\bar{\mathcal{A}}$	$\mathcal{A}_N$
Zero-shot	76.36	66.96	79.17	77.08
ADAM	79.53	71.26	72.06	70.90
SEMA	**82.74**	**73.52**	**80.94**	**78.18**

Table 15. Performance on pre-trained CLIP model.

C.11. Experiments with CLIP.

We further conduct experiment with a pre-trained vision-language model, namely CLIP with a ViT-B/16 backbone [58], and report the performance in Tab. 15. SEMA outperforms zero-shot CLIP and ADAM which have no parameter expansion, highlighting the effectiveness of our dynamic expansion strategy and its broad applicability to pre-trained models.