Adapter Merging with Centroid Prototype Mapping for Scalable Class-Incremental Learning

Supplementary Material

A. Dataset Details

This section outlines the benchmark datasets used in our experiments. Figure A shows sample images from CIFAR-100, CUB, ImageNet-R, ImageNet-A, and VTAB. Since the pre-trained model is trained on ImageNet-21K [37], the standard ImageNet dataset is excluded as a benchmark dataset to avoid data leakage.

CIFAR-100: CIFAR-100 [23] contains 100 object classes and is widely used for image classification. CIFAR-100 is a standard CIL benchmark due to its small image size and diverse object categories. CIFAR-100 is especially suitable for evaluating performance in simple CIL scenarios.

CUB: Caltech-UCSD Birds 200 (CUB) [47] contains 200 bird species and is a benchmark for fine-grained visual classification. Its main challenge lies in distinguishing visually similar categories, such as birds with subtle differences in shape, color, and texture. This requires models to perform fine-grained feature extraction and to handle high intra-class variability. This challenge makes it a valuable dataset for evaluating CIL performance in fine-grained settings.

ImageNet-R: ImageNet-R [15], a variant of the ImageNet dataset, consists of 200 classes with images drawn from diverse visual domains such as art, cartoons, and paintings. This dataset is commonly used to evaluate a model's ability to generalize across domains and is particularly useful for testing how models adapt to new domains in CIL.

ImageNet-A: ImageNet-A [14], a subset of ImageNet, contains 200 classes with adversarial or out-of-distribution examples that models often misclassify. This dataset, designed to challenge models trained on standard ImageNet, includes images that are difficult to classify. ImageNet-A is a benchmark for testing model robustness to adversarial attacks and generalization to unseen inputs.

VTAB: Visual Task Adaptation Benchmark (VTAB) [56] consists of various datasets aimed at evaluating model adaptability across diverse tasks. This benchmark is primarily used to evaluate transfer learning and domain adaptation performance. Following the protocol in [59], this study constructs a 50-class dataset using five VTAB subsets: Resisc45 (classes 1-10), Describable Textures Dataset (DTD) (classes 11-20), Oxford IIIT Pet dataset (classes 21-30), EuroSAT (classes 31-40), and 102 Category Flower Dataset (classes 41-50).

(a) CIFAR-100.

(b) CUB.

(c) ImageNet-R.

(d) ImageNet-A.

(e) VTAB.

Figure A. Example images from (a) CIFAR-100, (b) CUB, (c) ImageNet-R, (d) ImageNet-A, and (e) VTAB.

B. Implementation Details

This section provides details of the implementation setup. Table A lists the batch size, learning rate, weight decay, and number of training epochs for each dataset. Each experiment was run five times using seeds 1993, 1994, 1995, 1996, and 1997. The experiments were conducted on a single NVIDIA RTX A5000 GPU using PyTorch for model training and inference.

Model Architecture: The backbone model used in the experiments is ViT-B/16[2], with an embedding dimension of 768, a patch size of 16, and 12 transformer blocks. The multi-head attention employs 12 attention heads. The adapter is configured with a bottleneck dimension of 64, a dropout rate of 0.1, and an up-projection scale of 0.1.

Preprocessing: The preprocessing pipeline involves ran-

[2]https://github.com/huggingface/pytorch-image-models

Table A. Details of the training settings for each dataset, based on the configurations provided in [62].

dataset	batch size	learning rate	weight decay	epochs
CIFAR-100	48	2.5×10^{-2}	5.0×10^{-4}	20
CUB	32	8.0×10^{-3}	5.0×10^{-4}	20
ImageNet-R	16	5.0×10^{-2}	5.0×10^{-3}	20
ImageNet-A	32	5.0×10^{-2}	5.0×10^{-3}	20
VTAB	16	3.0×10^{-2}	5.0×10^{-3}	45

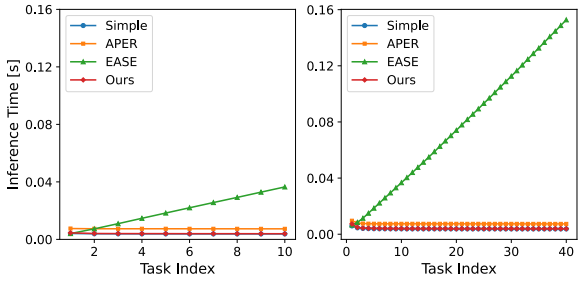


Figure B. Inference time curves per instance on ImageNet-R B0 Inc20 (left) and B0 Inc5 (right).

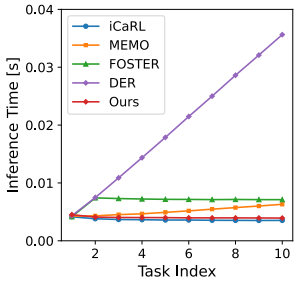


Figure C. Inference time curve per instance on ImageNet-R B0 Inc20.

dom cropping with scales ranging from 0.05 to 1.0 and aspect ratios between 3:4 and 4:3, followed by horizontal flipping with a probability of 0.5. The images are resized to 224×224 and normalized to the range [0, 1].

C. Inference Time Comparison

This section presents additional inference time results comparing ACMap (ours) and baseline methods.

Figure B shows the inference time curves per instance for both ImageNet-R B0 Inc20 (left) and B0 Inc5 (right), demonstrating how inference time scales with the number of tasks. EASE (green) [59] demonstrates a nearly linear increase in inference time as the number of tasks T increases, which is consistent with its $\mathcal{O}(T)$ complexity. In contrast, SimpleCIL [61], APER [61], and ACMap (ours) achieve an

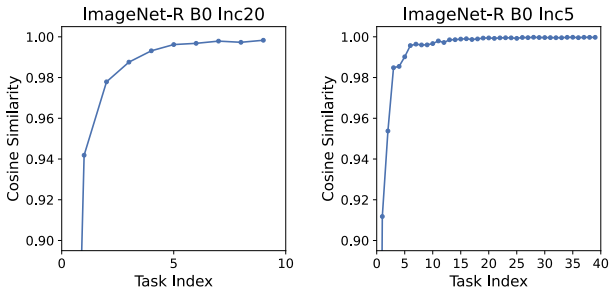


Figure D. Cosine similarity curves of $\mathrm{Sim}(\boldsymbol{P}_t(\bar{\mathcal{A}}_{t-1}), \boldsymbol{P}_t(\bar{\mathcal{A}}_t))$ on ImageNet-R B0 Inc20 (left) and B0 Inc5 (right).

$\mathcal{O}(1)$ inference time, maintaining a constant inference cost, regardless of the number of tasks. As mentioned in the main paper, our method achieves a T-fold speedup in inference compared to the state-of-the-art EASE, while maintaining an accuracy comparable to that of EASE. Moreover, while matching the inference time of SimpleCIL and APER, our method achieves higher accuracy.

Figure C further compares the inference time per instance between ACMap and other methods, including iCaRL [36], DER [54], FOSTER [48], and MEMO [57]. While DER and MEMO exhibit a linear increase in inference time as the number of tasks grows, ACMap maintains a constant inference time, similar to other parameter-efficient methods. This result highlights the scalability advantage of ACMap in CIL scenarios.

D. Early Stopping Threshold

The early stopping threshold L can be determined using the cosine similarity between prototypes before and after adapter merging, defined as $\mathrm{Sim}(\boldsymbol{P}_t(\bar{\mathcal{A}}_{t-1}), \boldsymbol{P}_t(\bar{\mathcal{A}}_t))$, where $\mathrm{Sim}(\cdot, \cdot)$ denotes cosine similarity. As t increases, it approaches 1, indicating that the difference between $\bar{\mathcal{A}}_t$ and $\bar{\mathcal{A}}_{t-1}$ becomes negligible, as shown in Figure D. This value guides the selection of an appropriate threshold.

E. Additional Experiments

This section presents supplementary experimental results that expand upon the findings of the main paper.

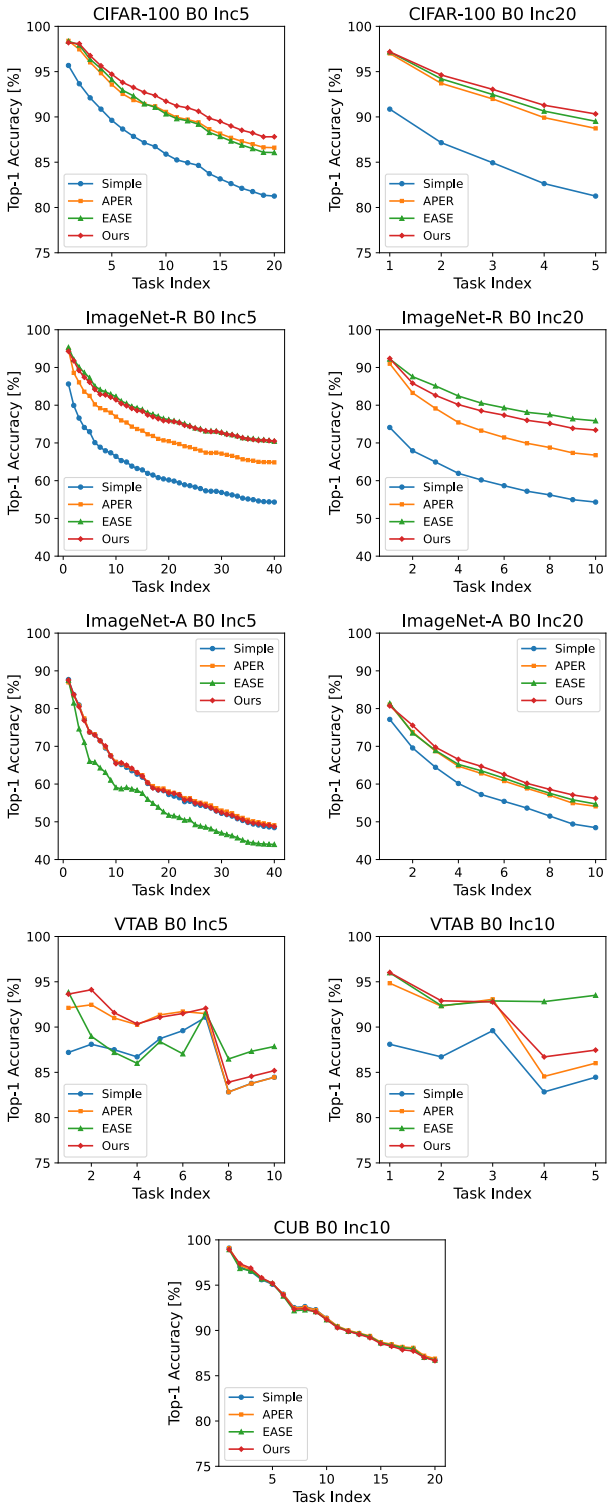


Figure E. Top-1 accuracy curves during CIL for all experiments conducted, comparing ACMap (ours) with SimpleCIL, APER, and EASE. These graphs include the results from the main paper for comparison and reference.

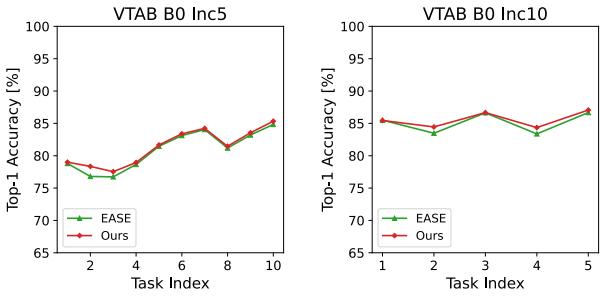


Figure F. Top-1 accuracy curves on balanced VTAB B0 Inc5 (left) and B0 Inc10 (right).

E.1. Top-1 Accuracy Comparison

Figure E presents the top-1 accuracy curves for all experiments conducted in this study, comparing ACMap (ours) with SimpleCIL, APER, and EASE. The graphs include the results from the main paper for comparison and reference.

The experimental results are consistent with those reported in the main paper, showing that ACMap outperforms or matches the accuracy of the other methods across all datasets except for VTAB. While ACMap achieves accuracy comparable to EASE, it is important to recall, as discussed in Appendix C, that ACMap is T-times faster than EASE. This emphasizes that ACMap delivers state-of-the-art accuracy while maintaining constant inference time, making it well-suited for scalable CIL.

When the number of tasks in VTAB reaches four in Figure E (fourth row), ACMap exhibits a significant drop in accuracy, resulting in lower performance than EASE. As discussed in Appendix G, this decline is likely caused by the data imbalance, which may lead to overfitting. EASE, by contrast, avoids this issue by maintaining separate adapters for each task, albeit at the cost of increased inference time.

E.2. Comparison on balanced VTAB

The accuracy drop in VTAB appears starting from the fourth task, as discussed in Appendix E.1. To examine whether this drop results from data imbalance, we conduct experiments under a balanced VTAB setting, where the number of samples per task is equalized. The results, as shown in Figure F, show that EASE and ACMap perform nearly identically, suggesting that the observed performance gap may stem from data imbalance rather than an inherent limitation of ACMap.

E.3. Comparison with Other Methods

Table B compares exemplar-based methods (iCaRL [36], DER [54], FOSTER [48], MEMO [57]) and exemplar-free methods (RanPAC [31], InfLoRA [27], and ACMap (ours)) using the average accuracy $\bar{A}$ and final accuracy A_T as evaluation metrics. The results for the exemplar-based methods

Table B. Average accuracy $\bar{A}$ and final accuracy A_T. iCaRL [36], DER [54], FOSTER [48], and MEMO [57] are exemplar-based methods, while RanPAC [31], InfLoRA [27], and ACMap (ours) are exemplar-free methods.

Method	Exemplars	CIFAR B0 Inc10		IN-R B0 Inc20	
		$\bar{A}$	A_T	$\bar{A}$	A_T
iCaRL [36]	20 / class	82.5	73.9	72.4	60.7
DER [54]	20 / class	86.0	77.9	80.5	74.3
FOSTER [48]	20 / class	**89.9**	**84.9**	**81.3**	**74.5**
MEMO [57]	20 / class	84.1	75.8	74.8	66.6
RanPAC [31]	-	**94.5**	**91.5**	**82.6**	**77.4**
InfLoRA [27]	-	91.7	86.5	80.8	75.7
Ours ($L = \infty$)	-	92.9	89.3	79.5	73.5

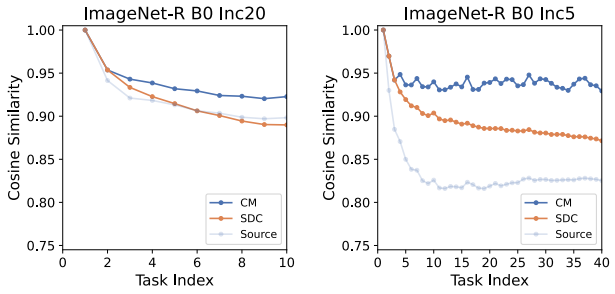


Figure G. Cosine similarity between the mapped and true prototypes on ImageNet-R B0 Inc20 (left) and B0 Inc5 (right). Blue represents CM, and orange represents SDC. Semi-transparent lines indicate cosine similarity between the source and true prototypes.

are taken from [59], and those for InfLoRA from [27], while the results for RanPAC and ACMap represent averages over five runs.

The exemplar-based methods use 20 exemplars per class. Despite being exemplar-free, ACMap achieves significantly better $\bar{A}$ and A_T on CIFAR and only slightly lower performance on IN-R. Among the exemplar-free methods, ACMap achieves comparable accuracy to RanPAC and outperforms InfLoRA on CIFAR. However, InfLoRA scales poorly due to the growth of the model size for each task. RanPAC requires M^2 non-trainable parameters ($M = 10^4$) for its random projection layer, which exceeds the total parameter count of ViT-B/16.

F. Evaluation of Prototype Alignment

Centroid prototype mapping (CM) improves upon existing approaches such as semantic drift compensation (SDC) [55] by achieving higher approximation accuracy. Unlike SDC, which sums incremental shifts and thus accumulates errors, CM applies a single centroid shift from task i to t. As shown in Figure G, CM achieves higher cosine similarity between

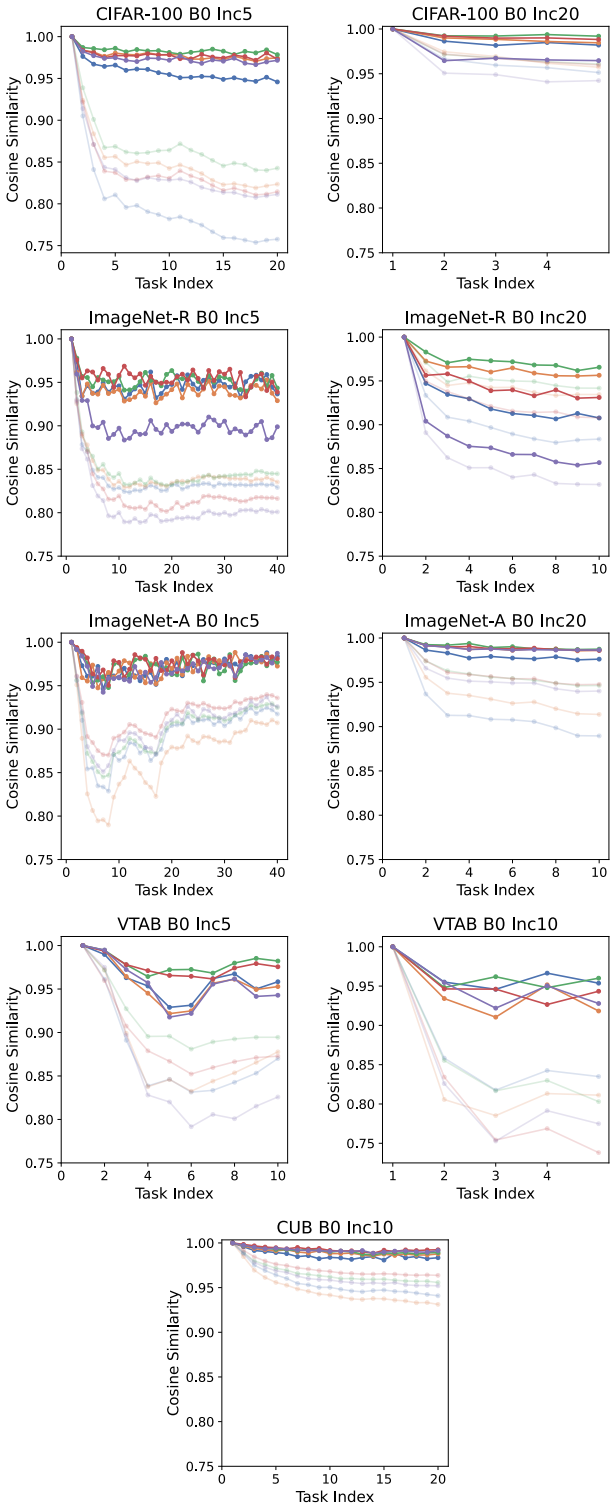


Figure H. Cosine similarity curves of $\mathrm{Sim}(\hat{\boldsymbol{P}}_1(\bar{\mathcal{A}}_1), \boldsymbol{P}_1(\bar{\mathcal{A}}_t))$, with solid lines showing the similarity between mapped and true prototypes, and semi-transparent lines between unmapped and true prototypes, illustrating the alignment achieved by centroid prototype mapping.

mapped and true prototypes, whereas SDC performance worsens as the number of tasks increases.

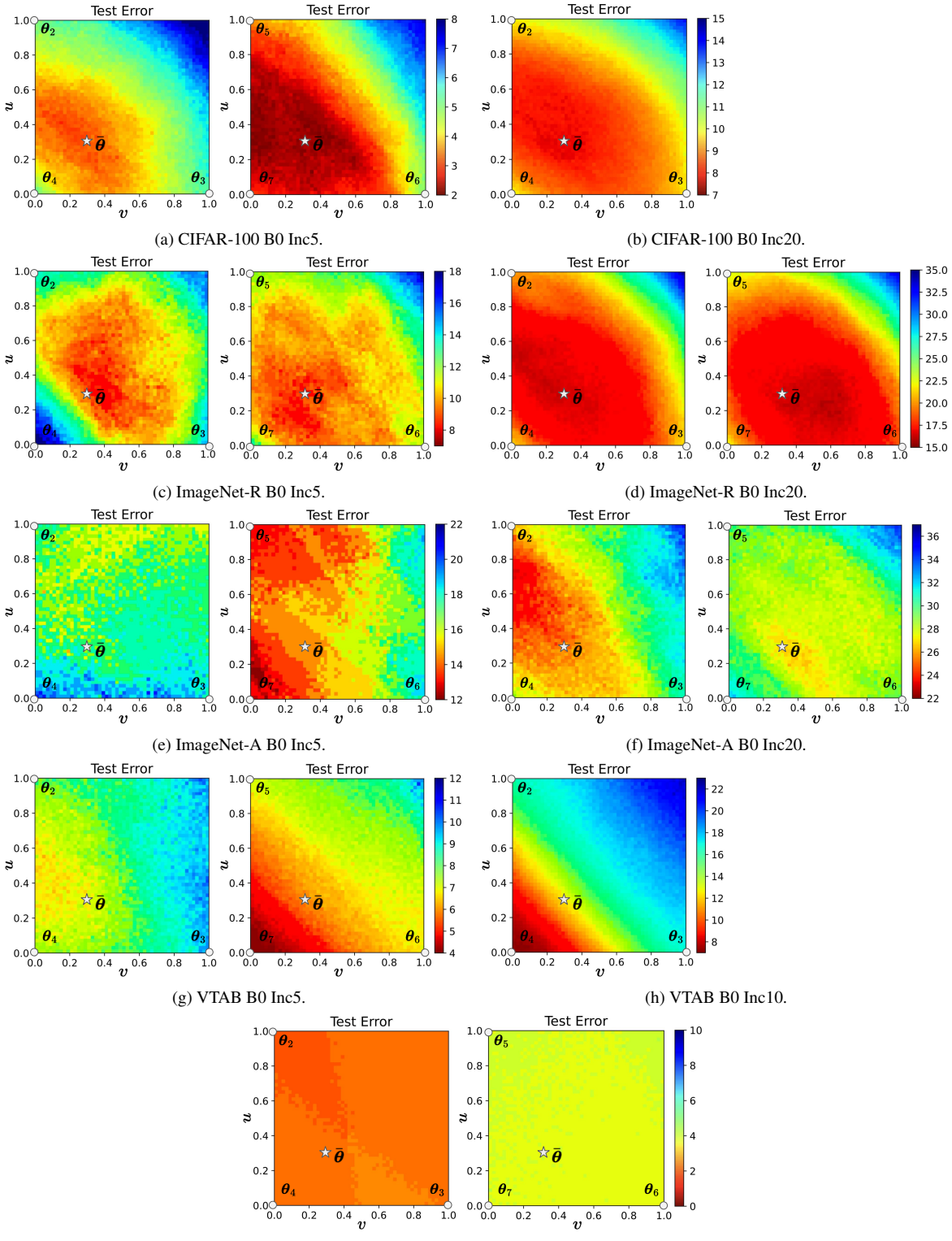
Figure H presents additional experiments evaluating CM's effectiveness. These experiments evaluate prototype alignment by measuring cosine similarities. The solid line shows the cosine similarity between the mapped and true prototypes, while the semi-transparent line shows that of the unmapped and true ones. Curve colors indicate the classes from the first task. Across all datasets, CM consistently improves alignment, as indicated by the solid lines exhibiting higher cosine similarity than the semi-transparent lines. This result demonstrates that CM effectively aligns previous task prototypes with the true prototypes in the current subspace. Interestingly, for CUB, a fine-grained classification dataset, the semi-transparent lines already exhibit high cosine similarity. This observation suggests that in fine-grained classification tasks, adapters and adapter merging may offer limited benefits. As shown in Table 1 of the main paper, SimpleCIL, which does not use adapters, achieves accuracy comparable to ACMap, APER, and EASE.

G. Landscape Analysis for Adapter Merging

Figure I presents the test-error landscapes of three successive adapters, $\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_t, \boldsymbol{\theta}_{t+1}$, obtained via linear interpolation on the datasets not covered in the main paper. For CIFAR-100 B0 Inc20 and VTAB B0 Inc10, only the result for $\boldsymbol{\theta}_2, \boldsymbol{\theta}_3, \boldsymbol{\theta}_4$ is presented because the task count is limited to five. Therefore, $\boldsymbol{\theta}_6, \boldsymbol{\theta}_7$ are not available.

Across all datasets except VTAB, these results indicate that ACMap promotes the formation of low-loss basins (red regions), suggesting favorable conditions for successful adapter merging. For CUB, instead of forming low-loss basins, the results show relatively flat landscapes. This result suggests that adapter merging may be unnecessary for achieving CIL in CUB, as discussed in Appendix F.

Moreover, as shown in Figure I (g), (h), low-loss basins are not observed for VTAB. As discussed in Section 5.2, the dataset size for the fourth task in VTAB is larger than that of the others, which may lead to overfitting on the fourth task. The low test-error rate (red) observed around $\boldsymbol{\theta}_4$ in Figure I (h) supports this hypothesis. This hypothesis is also supported by the VTAB B0 Inc10 result in Figure 5 of the main paper, where ACMap demonstrates a significant decline in accuracy starting from the fourth task.



(a) CIFAR-100 B0 Inc5.

(b) CIFAR-100 B0 Inc20.

(c) ImageNet-R B0 Inc5.

(d) ImageNet-R B0 Inc20.

(e) ImageNet-A B0 Inc5.

(f) ImageNet-A B0 Inc20.

(g) VTAB B0 Inc5.

(h) VTAB B0 Inc10.

(i) CUB B0 Inc10.

Figure I. Additional results for visualization of the test error on CIFAR-100, CUB, ImageNet-R, ImageNet-A, and VTAB using linearly interpolated adapter weights $\boldsymbol{\theta} = u\boldsymbol{\theta}_{t-1} + v\boldsymbol{\theta}_t + (1-u-v)\boldsymbol{\theta}_{t+1},\ (0 \leq u, v \leq 1)$ across three consecutive adapter weights $\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_t, \boldsymbol{\theta}_{t+1}$. For CIFAR-100 B0 Inc20 and VTAB B0 Inc10, only the result for $\boldsymbol{\theta}_2, \boldsymbol{\theta}_3, \boldsymbol{\theta}_4$ is presented because the task count is limited to five, meaning $\boldsymbol{\theta}_6, \boldsymbol{\theta}_7$ are not available.