KAC: Kolmogorov-Arnold Classifier for Continual Learning

Supplementary Material

7. More Experimental Results

Experiments on CIFAR-100. Tab. 5 compares the average incremental accuracy between the baseline methods and those with KAC in the CIFAR-100 dataset. Replacing the linear classifiers with KAC improves most of the methods, with a little drop in CPrompt and L2P. Due to the low pixel resolution of CIFAR-100, it is generally suitable for training smaller-scale networks. For pre-trained backbones, performance tends to be saturated, which is why our method does not show significant improvement on this dataset.

Table 5. The average incremental accuracy of CIFAR-100 10 steps scenario.

Method	Linear	KAC
L2P	83.78	83.71 ((-0.07))
DualPrompt	84.80	85.74 ((+0.94))
CODAPrompt	86.65	87.26 ((+0.61))
CPrompt	87.50	87.19 ((-0.31))

Table 6. Comparison of the KAC and KAC with shortcut on the last accuracy in ImageNet-R 20 steps scenario.

Method	Baseline	KAC + Shortcut	KAC
L2P	70.35	71.09	72.11
DualPrompt	70.68	72.83	74.68
CODAPrompt	74.40	75.57	77.94
CPrompt	74.32	73.55	75.73

Results of multiple runs on CUB200. To further evaluate the robustness of KAC, we generate random class sequences using multiple seeds and evaluate the performance of various methods using KAC and conventional linear classifiers. Specifically, we report the last accuracy for each setting on CUB200. Table 7 reports the mean and standard deviation of these metrics. The results demonstrate that KAC consistently outperforms the baseline methods across most experiments.

8. More Ablation Studies

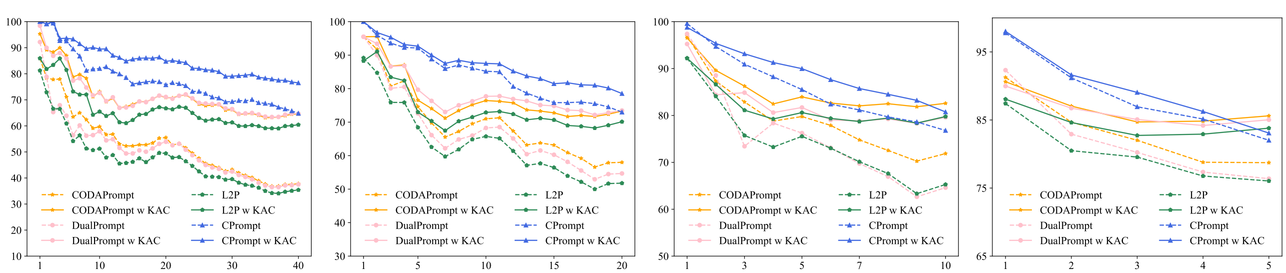
Ablation on the linear shortcut. In KAC, we don't follow conventional KAN, in which a linear shortcut is added with the spline functions. In this section, we show that the linear shortcut cannot help KAC achieve better performance. Tab. 6 reports the accuracy of the last task in ImageNet-R 20 steps scenario. It demonstrates that when linear shortcut is added, it achieves even worse accuracy, supporting our decision to remove the linear shortcut.

9. More Visualizations

Visualization of performance on CUB200. To investigate the reasons behind the superior performance of KAC on CUB200, we make an observation on the accuracy curves of CUB200 across experiments with different steps. As shown in Fig. 6, with the arriving of tasks, KAC demonstrates a growing advantage. In several steps, the baseline frequently experiences significant forgetting, while KAC often exhibits less forgetting compared to the linear classifier during these steps, which helps KAC accumulate a higher final accuracy.

Table 7. The results of multiple runs on CUB200 dataset with more seeds. The mean and std of the last accuracy are reported.

Method	5 steps		10 steps		20 steps		40 steps	
	mean	std	mean	std	mean	std	mean	std
L2P	76.60	0.79	69.23	2.86	59.11	5.06	41.65	4.71
w KAC	80.35	2.60	77.07	1.92	73.32	0.57	65.89	4.09
DualPrompt	77.35	1.68	71.13	1.88	61.91	5.03	44.98	5.21
w KAC	82.39	1.95	79.55	1.88	75.94	1.76	69.86	4.28
CODAPrompt	75.62	2.65	70.77	0.71	62.58	3.44	45.14	5.49
w KAC	82.88	1.96	78.74	2.73	74.74	0.194	70.57	4.27



(a) 40 steps (b) 20 steps (c) 10 steps (d) 5 steps

Figure 6. The accuracy curves for scenarios of different steps on the CUB200 dataset. The x-axis represents the gradually increasing tasks and the y-axis represents accuracy at each step. It can be observed that KAC follows the same trend as the baseline, but exhibits less forgetting at each step.