Tripartite Weight-Space Ensemble for Few-Shot Class-Incremental Learning

-Supplementary Materials-

Anonymous CVPR submission

Paper ID 3089

S-1. The degree of data amplification in ADKD

For tth session, we amplify the few-shot training dataset $D^{(t)}$ into $D^{(t)}_{\text{amp}}$ using CutMix. In order to apply CutMix, we randomly sample μNK pairs from N-way K-shot training examples. To verify the impact of the degree of the data amplification, we report the average Acc varying μ in Table S-1. From this result, we experimentally set μ as 4 for all the three benchmark datasets.

Table S-1. Impact of the degree of data amplification. We report the average Acc (%) across the sessions of miniImageNet dataset.

μ	1 (wo mixture)	2	3	4	5	6
Average Acc (%)	69.21	69.43	70.53	70.68	70.68	70.69

S-2. Sensitivity to the learning rate of feature extractor

In the previous works, the feature extractor has been fixed after the base session learning. Whereas, owing to the effectiveness of the proposed Tri-WE and ADKD, we can update the feature extractor, and then more evolve the model to accommodate new classes suppressing the catastrophic forgetting and overfitting. However, we still use a lower learning rate (lr) of 0.001 on the feature extractor, compared to the lr of the classification head (0.1). Here, we show the sensitivity to the learning rate of feature extractor in the incremental learning. As in Table S-2, the model is still degenerated when the lr is somewhat high. Nonetheless, the model's classification capability is restricted with smaller lr.

Table S-2. Sensitivity to the lr of feature extractor.

lr	10e-5	10e-4	10e-3	10e-2
Average Acc (%)	69.99	70.32	70.68	70.21

S-3. Base session training

To increase the generalization capability of the base session model, we learn the model in multi-task fashion using an auxiliary geometric transform classifier. In specific, we first generate B transformed images of an input by applying B different transformations. We apply 12 types of transformations where each is a combination of rotation, scale, aspect-ratio transformations, i.e. $B = 12$. Then, the transformed inputs are classified into one of B transformation categories. As the auxiliary classifier, we utilize an MLP (multilayer perceptron) module including two linear layers.

Algorithm 1 Pseudo code for the proposed FSCIL method

Input: Previous feature extractor $g_\theta^{(t-1)}$, previous classification head $h_\phi^{(t-1)}$, base classification head $h_\phi^{(0)}$, Memory buffer $\mathcal{M}$

Output: Evolved feature extractor $g_\theta^{(t)}$, classification head $h_\phi^{(t)}$, and Memory buffer $\mathcal{M}$

Initialize ϕ_0 from ϕ of sessions $\mathcal{S}^{(0)}$, ϕ_{old} & ϕ_{all} from ϕ of $\mathcal{S}^{(t-1)}$, θ from θ of $\mathcal{S}^{(t-1)}$, and α_1 & α_2 from pre-defined values
while not done **do**
  $h_\phi^{(t)}$'s weight ϕ is obtained by interpolating ϕ_0, ϕ_{old}, ϕ_{all} with learnable scalars α_1, α_2 as in Eq.(1),(2)
  For the few-shot example set $D^{(t)}$, compute the loss $\mathcal{L}_{\text{Cls}}$ and $\mathcal{L}_{\text{Cls-Old}}$ on the predictions of $h_\phi^{(t)}$ and $h_{\phi_{\text{old}}}^{(t)}$, respectively as in Eq.(3),(4)
  Amplify the few-shot example set $D^{(t)}$ to $D^{(t)}_{\text{amp}}$
  For the amplified set $D^{(t)}_{\text{amp}}$, compute the loss $\mathcal{L}_{\text{ADKD}} = \mathcal{L}_{\text{feat}} + \mathcal{L}_{\text{logit}}$ as in Eq.(5)
  Compute total loss $\mathcal{L}$ in Eq.(8)
  Update $\{\phi_{\text{all}}, \phi_{\text{old}}, \theta, \alpha_1, \alpha_2\}$ with gradient $\Delta\mathcal{L}$
end while
Deploy $g_\theta^{(t)}$,$h_\phi^{(t)}$

S-4. Further comparative results

In Tables S-3 and S-4, we provide more detailed results on CUB200 and CIFAR100 datasets. On the CUB200 dataset,

Table S-3. **Comparative results on CUB200 dataset.**

Method	Acc in each session (%)											Avg	Last sess. impro.
	0	1	2	3	4	5	6	7	8	9	10		
CEC [15]	75.85	71.94	68.5	63.5	62.43	58.27	57.73	55.81	54.83	53.52	52.28	61.33	+11.63
SynthFeat [3]	68.78	59.37	59.32	54.96	52.58	49.81	48.09	46.32	44.33	43.43	43.23	51.84	+20.68
MetaFSCIL [4]	75.90	72.41	68.78	64.78	62.96	59.99	58.3	56.85	54.78	53.82	52.64	61.93	+11.27
VarOpen [1]	79.60	73.46	70.32	66.38	63.97	59.63	58.19	57.56	55.01	54.31	52.98	62.86	+10.93
FACT [17]	75.90	73.23	70.84	66.13	65.56	62.15	61.74	59.83	58.41	57.89	56.94	64.42	+6.97
Replay [10]	75.90	72.14	68.64	63.76	62.58	59.11	57.82	55.89	54.92	53.58	52.39	61.52	+11.52
ALICE [12]	77.42	72.71	70.62	67.24	65.91	63.40	62.92	61.91	60.54	60.62	60.12	65.75	+3.79
S3C [6]	80.62	77.55	73.19	68.54	68.05	64.33	63.58	62.07	60.61	59.79	58.95	67.03	+4.96
WaRP [8]	77.74	74.15	70.82	66.90	65.01	62.64	61.40	59.86	57.95	57.77	57.01	64.66	+6.90
SoftNet [7]	78.07	74.58	71.37	67.54	65.37	62.60	61.07	59.37	57.53	57.21	56.75	64.68	+7.16
NC-FSCIL [14]	80.45	75.98	72.3	70.28	68.17	65.16	64.43	63.25	60.66	60.01	59.44	67.28	+4.47
GKEAL [18]	78.88	75.62	72.32	68.62	67.23	64.26	62.98	61.89	60.20	59.21	58.67	66.35	+5.24
BiDistill [16]	79.12	75.37	72.8	69.05	67.53	65.12	64.00	63.51	61.87	61.47	60.93	67.34	+2.98
SAVC [13]	81.85	77.92	74.95	70.21	69.96	67.02	66.16	65.3	63.84	63.15	62.5	69.35	+1.41
OrCo [2]	75.59	66.85	64.05	63.69	62.20	60.38	60.18	59.20	58.00	58.42	57.94	62.41	+5.97
CLOSER [11]	79.40	75.92	73.50	70.47	69.24	67.22	66.73	65.69	64.00	64.02	63.58	68.04	+0.33
Ours	81.56	78.57	76.05	73.55	71.84	69.12	67.82	66.84	65.82	65.04	63.91	70.92	

Table S-4. **Comparative results on CIFAR100 dataset.**

Method	Acc in each session (%)									Avg	Last sess. impro.
	0	1	2	3	4	5	6	7	8		
CEC [15]	73.07	68.88	65.26	61.19	58.09	55.57	53.22	51.34	49.14	59.53	+9.08
SynthFeat [3]	62.00	57.00	56.70	52.00	50.60	48.80	45.00	44.00	41.64	50.86	+16.58
MetaFSCIL [4]	74.50	70.10	66.84	62.77	59.48	56.52	54.36	52.56	49.97	60.79	+8.25
FACT [17]	74.60	72.09	67.56	63.52	61.38	58.36	56.28	54.24	52.10	62.24	+6.12
Replay [10]	74.40	70.20	66.54	62.51	59.71	56.58	54.52	52.39	50.14	60.78	+8.08
ALICE [12]	79.00	70.50	67.10	63.40	61.20	59.20	58.10	56.30	54.10	63.21	+4.12
S3C [6]	78.16	74.03	70.17	66.09	63.66	59.91	58.37	56.78	54.06	64.58	+4.16
WaRP [8]	80.31	75.86	71.87	67.58	64.39	61.34	59.15	57.10	54.74	65.82	+3.48
SoftNet [7]	80.33	76.23	72.19	67.83	64.64	61.39	59.32	57.37	54.94	66.03	+3.28
NC-FSCIL [14]	82.52	76.82	73.34	69.68	66.19	62.85	60.96	59.02	56.11	**67.50**	+2.11
GKEAL [18]	74.01	70.45	67.01	63.08	60.01	57.30	55.50	53.39	51.40	61.35	+6.82
BiDistill [16]	79.45	75.38	71.84	67.95	64.96	61.95	60.16	57.67	55.88	66.14	+2.34
SAVC [13]	78.77	73.31	69.31	64.93	61.70	59.25	57.13	55.19	53.12	63.63	+5.10
OrCo [2]	80.08	68.16	66.99	60.97	59.78	58.60	57.04	55.13	52.19	62.10	+6.03
CLOSER [11]	75.72	71.83	68.32	64.62	61.91	59.25	57.53	55.43	53.32	63.10	+4.90
Ours	81.92	77.55	74.46	71.14	66.82	64.02	62.14	61.71	58.22	68.66	

with the lower base session Acc to the previous SOTA [13], we attain better results over all the incremental sessions. And, our method even outperforms the NC-FSCIL [14] and Orco [2] which have advantage of knowing the topology of all the incrementally appeared classes at the start of base session learning. Also, on the CIFAR100 dataset, our method shows slightly lower performance than NC-FSCIL for average Acc, but attains the higher last session Acc.

Moreover, in FSCIL field, increasing the base session model's generalization capability has been also key aims. Hence, the existing methods used their own base session models. Accordingly, as aforementioned in Sec. S-3, we also added the auxiliary classifier on top of ALICE [12] during the base session learning. For a more fair comparison, when we starts from the base session model learned without this auxiliary classifier (i.e. using the base session model of the ALICE as it is), we attain 70.62% session-wise Acc in average on the miniImageNet, which is still clearly better than 67.82% of the second-best NC-FSCIL.

S-5. More implementation detail

We implement our method using the PyTorch library. We follow most of experimental set up in the previous works [7, 9, 12, 13, 16]. ResNet18 [5] serves as our feature extractor. The entire model is trained via the SGD optimizer. For the base session, the learning rates are initialized by 0.01, 0.001, and 0.01 for miniImageNet, CUB200, and CIFAR 100 respectively, and decayed by 0.1 at 60 and 70 epochs. In each incremental session, the learning rate for the weight-space ensembled ϕ_{all} is 0.1, while it is 0.001 for the rest. Hence, the feature extractor is minimally updated in incremental learning. For ADKD, the given K training examples are amplified to $4K$ for each class. Also, the learnable scalars α_1 and α_2 are initially set to 1.0. Loss weights λ_1 and λ_2 are empirically determined to be 1.2 and 10.0, respectively. Also, we learn the model during 100 epochs for all the datasets in each incremental session.

S-6. Session-wise Accuracy for fixed (α_1, α_2)

(α_1, α_2) means the dependency to the knowledge of base and previous classes. To compare with the proposed learnable approaches, we provide the session-wise accuracy for different fixed (α_1, α_2). As the novel classes are typically less dominant than the base and previous classes in a session, it is intuitive that higher α_1, α_2 seems beneficial in overall. However, our learnable approach consistently better than the fixed cases.

(α_1, α_2)	0	1	2	3	4	5	6	7	8
(0.5, 0.5)	84.13	79.83	75.91	72.34	67.42	64.82	61.83	61.11	57.77
(0.7, 0.7)	84.13	80.93	76.17	72.47	68.23	65.34	62.43	61.08	58.73
(1.0, 1.0)	84.13	81.04	76.37	72.32	68.45	65.41	62.61	61.39	59.49
Learnable (ours)	84.13	81.41	76.65	73.59	70.10	65.13	63.42	61.02	60.13

References

[1] Touqeer Ahmad, Akshay Raj Dhamija, Mohsen Jafarzadeh, Steve Cruz, Ryan Rabinowitz, Chunchun Li, and Terrance E Boult. Variable few shot class incremental and open world learning. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2022. 2

[2] Noor Ahmed, Anna Kukleva, and Bernt Schiele. Orco: Towards better generalization via orthogonality and contrast for few-shot class-incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024. 2

[3] Ali Cheraghian, Shafin Rahman, Sameera Ramasinghe, Pengfei Fang, Christian Simon, Lars Petersson, and Mehrtash Harandi. Synthesized feature based few-shot class-incremental learning on a mixture of subspaces. In *Int. Conf. Comput. Vis.*, 2021. 2

[4] Zhixiang Chi, Li Gu, Huan Liu, Yang Wang, Yuanhao Yu, and Jin Tang. Metafscil: A meta-learning approach for few-shot class incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 2

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 3

[6] Jayateja Kalla and Soma Biswas. S3c: Self-supervised stochastic classifiers for few-shot class-incremental learning. In *Eur. Conf. Comput. Vis.*, 2022. 2

[7] Haeyong Kang, Jaehong Yoon, Sultan Rizky Hikmawan Madjid, Sung Ju Hwang, and Chang D Yoo. On the soft-subnetwork for few-shot class incremental learning. In *Int. Conf. Learn. Represent.*, 2023. 2

[8] Do-Yeon Kim, Dong-Jun Han, Jun Seo, and Jaekyun Moon. Warping the space: Weight space rotation for class-incremental few-shot learning. In *Int. Conf. Learn. Represent.*, 2023. 2

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Adv. Neural Inform. Process. Syst.* 2012. 3

[10] Huan Liu, Li Gu, Zhixiang Chi, Yang Wang, Yuanhao Yu, Jun Chen, and Jin Tang. Few-shot class-incremental learning via entropy-regularized data-free replay. In *Eur. Conf. Comput. Vis.*, 2022. 2

[11] Junghun Oh, Sungyong Baik, and Kyoung Mu Lee. Closer: Towards better representation learning for few-shot class-incremental learning. In *Eur. Conf. Comput. Vis.*, 2024. 2

[12] Can Peng, Kun Zhao, Tianren Wang, Meng Li, and Brian C Lovell. Few-shot class-incremental learning from an open-set perspective. In *Eur. Conf. Comput. Vis.*, 2022. 2, 3

[13] Zeyin Song, Yifan Zhao, Yujun Shi, Peixi Peng, Li Yuan, and Yonghong Tian. Learning with fantasy: Semantic-aware virtual contrastive constraint for few-shot class-incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 2, 3

[14] Yibo Yang, Haobo Yuan, Xiangtai Li, Zhouchen Lin, Philip Torr, and Dacheng Tao. Neural collapse inspired feature-classifier alignment for few-shot class incremental learning. In *Int. Conf. Learn. Represent.*, 2023. 2

[15] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 2

[16] Linglan Zhao, Jing Lu, Yunlu Xu, Zhanzhan Cheng, Dashan Guo, Yi Niu, and Xiangzhong Fang. Few-shot class-incremental learning via class-aware bilateral distillation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 2, 3

[17] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 2

[18] Huiping Zhuang, Zhenyu Weng, Run He, Zhiping Lin, and Ziqian Zeng. Gkeal: Gaussian kernel embedded analytic learning for few-shot class incremental task. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 2