Learning Instance and Task-Aware Dynamic Kernels for Few-Shot Learning Supplementary Material

Rongkai Ma[1], Pengfei Fang[1,2,3(✉)], Gil Avraham[4], Yan Zuo[3], Tianyu Zhu[1], Tom Drummond[5], Mehrtash Harandi[1,3]

[1]Monash University [2]Australian National University [3]CSIRO
[4]Amazon Australia [5]The University of Melbourne
✉: Corresponding author (pengfei.fang@monash.edu)

1 Implementation Details

1.1 Datasets

***mini*-ImageNet.** The *mini*-ImageNet is sampled from ImageNet [3]. This dataset has 100 classes, with each having 600 samples. We follow the standard protocol [14] to split the dataset into 64 training, 16 validation, and 20 testing classes.
***tiered*-ImageNet.** Similar to *mini*-ImageNet, *tiered*-ImageNet is also a subset of the ImageNet. This dataset consists of 608 classes from 34 categories and is split into 351 classes from 20 categories for training, 97 classes from 6 categories for validation, and 160 classes from 8 categories for testing.
CUB. The CUB is a fine-grained dataset, which consists of 11,788 images from 200 different breeds of birds. We follow the standard settings [9], in which the dataset is split into 100/50/50 breeds for training, validation, and testing, respectively.
FC100. FC100 dataset is a variant of the standard CIFAR100 dataset [7], which contains images from 100 classes, with each class containing 600 samples. We follow the standard setting [12], where the dataset is split into 60/20/20 classes for training, validation and testing, respectively.
MS COCO and PASCAL VOC Datasets. In the few-shot detection task, we follow the protocol used in [4] to construct the dataset, where images from 60 categories of the MS COCO dataset are used for training and images from the rest of 20 common categories between MS COCO and PASCAL VOC datasets are used for testing.

1.2 Few-Shot Classification Hyperparameters

Network and Optimizer. We use the ResNet-10 backbone [2] for the MAML and the ResNet-12 backbone [20, 21] for the other two baselines across all four benchmarks. Noted additional ResNet-18 back1bone [23] is employed for the ProtoNet experiments on CUB. We fix the size of input images to 84×84 for ProtoNet and DeepEMD baselines and 224×224 for MAML baseline (We strictly

follow the same pre-processing protocol[1] in the original DeepEMD implementation to implement the metric-based baselines and our model, *i.e.*, DeepEMD, ProtoNet, INSTA-DeepEMD, and INSTA-ProtoNet. For MAML, we strictly follow the pre-processing protocol implemented in [2][2]). We use SGD optimizer for ProtoNet and DeepEMD experiments [20,21] and AdamW optimizer for MAML experiments [2] across all the datasets. For ProtoNet and DeepEMD baselines, we use L2 regularizer with 0.0005 weight decay factor. In the MAML baseline, the weight dacay factor is 0.01. For ResNet-18 and ResNet-10 backbones, we disable the average pooling and remove the last fully connected (FC) layer to produce the feature maps with size of $512 \times 11 \times 11$ and $512 \times 7 \times 7$, respectively. In the ResNet-12 backbone, the network produces the feature map with a size of $640 \times 5 \times 5$.

Training. We follow the good practice in the state-of-the-art models [16,20,21], where the network training is split into two stages, *i.e.* pre-training and meta-training stages. During the pre-training stage, the backbone network with an FC layer is trained on all the training classes of the dataset with the standard classification task. We select the network with the highest validation accuracy as the pre-trained backbone network for the meta-training stage. During the meta-training stage, we follow the standard episodic training protocol [18] to train the entire model. We set a small learning rate (0.0002) for the backbone and a larger learning rate (0.0002×25) for the other modules during the meta-training stage. Additionally, we use the cosine annealing learning rate scheduler over 200 epochs.

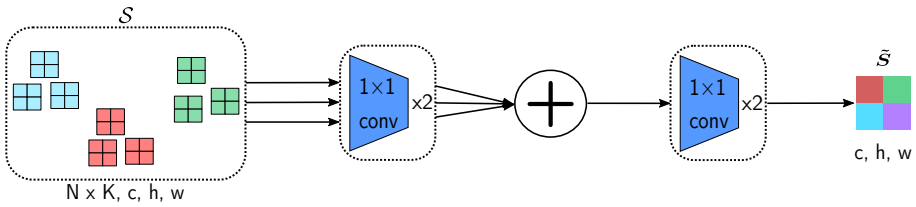

Fig. 1: The conceptual diagram of the context learning module

1.3 Few-Shot Detection

For the few-shot detection task, we consider the method proposed by Fan *et al.* [4] as our baseline model, which inherits from the faster-R-CNN [15] framework. Similar to the few-shot classification task, we implement our method to produce dynamic kernels and perform convolution on the feature maps extracted

[1] https://github.com/icoz69/DeepEMD.
[2] https://github.com/wyharveychen/CloserLookFewShot.

by the backbone network (*i.e.* ResNet-50), which is followed by a region proposal network (RPN), a region of interest pooling (ROI pooling) operation, and the classification and bounding box regression heads, outputting the categories and bounding boxes for the objects to the query image. Notably, For a fair comparison, we do not use any additional data augmentation. Please refer to [4] for more details of the framework and training strategy.

2 Additional Experiment Results

2.1 *mini*-ImageNet, *tiered*-ImageNet, and CUB

In this part, we provide extra comparison between our model and other state-of-the-art models on *mini*-ImageNet, *tiered*-ImageNet, and CUB. We follow the same evaluation protocol in [21][3] to evaluate the metric-based baseline models and our models (*i.e.*, DeepEMD, ProtoNet, INSTA-DeepEMD, and INSTA-ProtoNet), where 5,000 and 600 episodes are randomly sampled for 1-shot and 5-shot settings. For MAML baseline and INSTA-MAML, we strictly follow the same evaluation protocol in [2][4], where 600 episodes are randomly sampled for both 1-shot and 5-shot settings.

Table 1: Few-shot classification accuracy and 95% confidence interval on *mini*-ImageNet and *tiered*-ImageNet with ResNet backbones

Model	Backbone	*mini*-ImageNet		*tiered*-ImageNet	
		5-way 1-shot	5-way 5-shot	5-way 1-shot	5-way 5-shot
MetaOptNet-SVM [8]	ResNet-12	64.09 ± 0.62	80.00 ± 0.45	-	-
Neg-Margin [9]	ResNet-12	63.85 ± 0.81	81.57 ± 0.56	-	-
TPN [10]	ResNet-12	59.46	75.65	-	-
DSN-MR [16]	ResNet-12	64.60 ± 0.72	79.51 ± 0.50	67.39 ± 0.82	82.82 ± 0.56
E^3BM [11]	ResNet-12	63.80 ± 0.40	80.10 ± 0.30	71.20 ± 0.40	85.30 ± 0.30
ConstellationNet [19]	ResNet-12	64.89 ± 0.23	79.95 ± 0.37	-	-
MELR [5]	ResNet-12	67.40 ± 0.43	83.40 ± 0.28	72.14 ± 0.51	87.01 ± 0.35
CNL [22]	ResNet-12	67.96 ± 0.98	83.36 ± 0.51	73.42 ± 0.95	87.72 ± 0.75
MAML* [6]	ResNet-10	54.73 ± 0.87	66.72 ± 0.81	59.85 ± 0.97	73.20 ± 0.81
INSTA-MAML*	ResNet-10	$\mathbf{56.41 \pm 0.87}$	$\mathbf{71.56 \pm 0.75}$	$\mathbf{63.34 \pm 0.92}$	$\mathbf{78.01 \pm 0.71}$
ProtoNet* [17]	ResNet-12	62.29 ± 0.33	79.46 ± 0.48	68.25 ± 0.23	84.01 ± 0.56
INSTA-ProtoNet*	ResNet-12	$\mathbf{67.01 \pm 0.30}$	$\mathbf{83.13 \pm 0.56}$	$\mathbf{70.65 \pm 0.33}$	$\mathbf{85.76 \pm 0.59}$
DeepEMD*♣ [21]	ResNet-12	67.37 ± 0.45	83.17 ± 0.75	73.19 ± 0.32	86.79 ± 0.61
INSTA-DeepEMD*♣	ResNet-12	$\mathbf{68.46 \pm 0.48}$	$\mathbf{84.21 \pm 0.82}$	$\mathbf{73.87 \pm 0.31}$	$\mathbf{88.02 \pm 0.61}$

Table 2: Few-shot classification accuracy and 95% confidence interval on CUB with ResNet backbones

Model	Backbone	5-way 1-shot	5-way 5-shot
MAML* [6]	ResNet-10	70.46 ± 0.97	80.15 ± 0.73
INSTA-MAML*	ResNet-10	$\mathbf{73.08 \pm 0.97}$	$\mathbf{84.26 \pm 0.66}$
DeepEMD*♣ [21]	ResNet-12	74.55 ± 0.30	87.55 ± 0.54
INSTA-DeepEMD*♣	ResNet-12	$\mathbf{75.26 \pm 0.31}$	$\mathbf{88.12 \pm 0.54}$
ProtoNet*	ResNet-18	75.06 ± 0.30	87.39 ± 0.48
INSTA-ProtoNet*	ResNet-18	$\mathbf{77.18 \pm 0.29}$	$\mathbf{89.54 \pm 0.44}$

Table 3: Few-shot classification results on Meta-dataset with ResNet-18 backbone

Dataset	Simple-CNAPS	INSTA-Simple-CNAPS
ILSVRC	55.5 ± 1.1	$\mathbf{58.5 \pm 1.1}$
Omniglot	91.0 ± 0.6	$\mathbf{91.9 \pm 0.6}$
Aircraft	81.2 ± 0.7	$\mathbf{82.4 \pm 0.8}$
Birds	74.3 ± 0.9	$\mathbf{75.7 \pm 0.8}$
Textures	66.9 ± 0.8	$\mathbf{67.8 \pm 0.8}$
Quick Draw	76.7 ± 0.8	$\mathbf{76.8 \pm 0.8}$
Fungi	47.5 ± 1.0	$\mathbf{49.2 \pm 1.1}$
VGG Flowers	$\mathbf{90.5 \pm 0.6}$	90.4 ± 0.6
Traffic Signs	72.0 ± 0.7	$\mathbf{74.1 \pm 0.7}$
MSCOCO	47.3 ± 1.1	$\mathbf{53.9 \pm 1.1}$

2.2 Meta-Dataset

To verify the effectiveness of our method on the cross-domain few-shot classification problem, we incorporate INSTA into the baseline model simple-CNAPS [1]. In this experiment, we fix the trained baseline model and only fine-tune the modules to generate the INSTA dynamic kernels (*i.e.*, dynamic kernel generator and context learning module). We follow the same implementation of simple-CNAPS[5] to conduct this experiment (*e.g.*, 8×10^{-3} as learning rate, Adam as the optimizer, *etc.*). As the results in Table 3 suggested, INSTA improves the baseline over almost all the datasets, which again shows the effectiveness of our proposed INSTA dynamic kernels.

[3] https://github.com/icoz69/DeepEMD/blob/master/eval.py.
[4] https://github.com/wyharveychen/CloserLookFewShot/blob/master/test.py.
[5] https://github.com/peymanbateni/simple-cnaps/tree/master/simple-cnaps-src.

2.3 Ablation Study

In this part, we provide extra ablation studies on the effect of the residual connection and the spatial size of our dynamic kernel.

Residual Connection. In this experiment, we study the effect of the residual connection in our framework. In setting (i) of the Table 4, we disable the residual connection between the adapted and original features. The result suggests that the residual connection is an essential design choice for our framework.

Kernel Size. We provide an extra study on the effect of the spatial size of our dynamic kernel. Given that the feature map extracted by ResNet-12 has spatial size 5×5, the dynamic kernel size is constrained smaller or equal to 5×5. Therefore, in this study, we compare the results when the dynamic kernel size $k = 5 \times 5$ to our final design ($k = 3 \times 3$).

Table 4: The extra ablation study for the effect of the residual connection and spatial size of our dynamic kernel

ID	Model	5-way 5-shot
(i)	INSTA-ProtoNet w/o residual	80.03
(ii)	INSTA-ProtoNet w/ 5×5 $\boldsymbol{G}^{dy}$	82.43
(iii)	INSTA-ProtoNet	83.13

3 Multi-Spectral Attention

In this section, we provide more details for using the 2D-DCT to obtain the frequency-encoded vector. We first introduce the basis function of the 2D-DCT. The basis $B^{a,b}_{u,v}$ of the 2D-DCT is given by:

$$B^{a,b}_{u,v} = \cos(\frac{\pi u}{h}(a + \frac{1}{2}))\cos(\frac{\pi v}{w}(b + \frac{1}{2})), \tag{1}$$

where u, v are the frequency components of a basis. Then the frequency-encoded vector of a 3D-tensor $\boldsymbol{S}^i \in \mathbb{R}^{\frac{c}{n} \times h \times w}$ can be obtained by:

$$\boldsymbol{\tau}^i = \sum_{a=0}^{h-1} \sum_{b=0}^{w-1} \boldsymbol{S}_{:,a,b} B^{a,b}_{u_i,v_i}$$
$$s.t.\ i \in \{0, 1, \ldots, n-1\},$$

where $\boldsymbol{\tau}^i \in \mathbb{R}^{\frac{c}{n}}$ is the frequency-encoded vector, h and w are the height and width of the input signal. We pick the lowest 16 frequency components for our basis function according to [13]. Finally, we concatenate all the frequency-encoded vectors as:

$$\boldsymbol{\tau} = \text{concat}(\boldsymbol{\tau}^0, \boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^{n-1}), \tag{2}$$

where $\boldsymbol{\tau} \in \mathbb{R}^c$.

References

1. Bateni, P., Goyal, R., Masrani, V., Wood, F., Sigal, L.: Improved few-shot visual classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020) 4
2. Chen, W.Y., Liu, Y.C., Kira, Z., Wang, Y.C.F., Huang, J.B.: A closer look at few-shot classification. arXiv preprint arXiv:1904.04232 (2019) 1, 2, 3
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009) 1
4. Fan, Q., Zhuo, W., Tang, C.K., Tai, Y.W.: Few-shot object detection with attention-rpn and multi-relation detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4013–4022 (2020) 1, 2, 3
5. Fei, N., Lu, Z., Xiang, T., Huang, S.: Melr: Meta-learning via modeling episode-level relationships for few-shot learning. In: International Conference on Learning Representations (2020) 3
6. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. arXiv preprint arXiv:1703.03400 (2017) 3, 4
7. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009) 1
8. Lee, K., Maji, S., Ravichandran, A., Soatto, S.: Meta-learning with differentiable convex optimization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10657–10665 (2019) 3
9. Liu, B., Cao, Y., Lin, Y., Li, Q., Zhang, Z., Long, M., Hu, H.: Negative margin matters: Understanding margin in few-shot classification. arXiv preprint arXiv:2003.12060 (2020) 1, 3
10. Liu, Y., Lee, J., Park, M., Kim, S., Yang, E., Hwang, S.J., Yang, Y.: Learning to propagate labels: Transductive propagation network for few-shot learning. arXiv preprint arXiv:1805.10002 (2018) 3
11. Liu, Y., Schiele, B., Sun, Q.: An ensemble of epoch-wise empirical bayes for few-shot learning. In: European Conference on Computer Vision. pp. 404–421. Springer (2020) 3
12. Oreshkin, B.N., Rodriguez, P., Lacoste, A.: Tadam: task dependent adaptive metric for improved few-shot learning. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. pp. 719–729 (2018) 1
13. Qin, Z., Zhang, P., Wu, F., Li, X.: Fcanet: Frequency channel attention networks. arXiv preprint arXiv:2012.11879 (2020) 5
14. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: ICLR (2017) 1
15. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28**, 91–99 (2015) 2
16. Simon, C., Koniusz, P., Nock, R., Harandi, M.: Adaptive subspaces for few-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4136–4145 (2020) 2, 3
17. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in neural information processing systems. pp. 4077–4087 (2017) 3
18. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: Advances in neural information processing systems. pp. 3630–3638 (2016) 2

19. Xu, W., Wang, H., Tu, Z., et al.: Attentional constellation nets for few-shot learning. In: International Conference on Learning Representations (2020) 3
20. Ye, H.J., Hu, H., Zhan, D.C., Sha, F.: Few-shot learning via embedding adaptation with set-to-set functions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8808–8817 (2020) 1, 2
21. Zhang, C., Cai, Y., Lin, G., Shen, C.: Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12203–12213 (2020) 1, 2, 3, 4
22. Zhao, J., Yang, Y., Lin, X., Yang, J., He, L.: Looking wider for better adaptive representation in few-shot learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 10981–10989 (2021) 3
23. Ziko, I., Dolz, J., Granger, E., Ayed, I.B.: Laplacian regularized few-shot learning. In: International Conference on Machine Learning. pp. 11660–11670. PMLR (2020) 1