Supplementary Document for "Revisiting Prototypical Network for Cross Domain Few-Shot Learning"

A. Overview

This supplementary material includes:

- Implementation details for exploiting the full data in the few-shot task (in Sec.B);
- Hyper-parameters validation, *e.g.*, the weight coefficient of the loss function, the momentum, and the number and size of multi-crop (in Sec.C);
- More feature visualization results (in Sec.D);
- Results with 95% confidence intervals. (in Sec.E);

B. The implementation details of exploiting the full data in the few-shot task

State-of-the-art methods usually exploit the full data in the few-shot task to improve performance, such as TPN+ATA† [1], TPN+AFA† [2] and RDC† [3]. Exploiting the full data means that the samples in the query set are also used but in an unsupervised fashion. In order to maintain a fair comparison with the methods, the proposed method (LDP-net†) also exploited the full data in the few-shot task. We take a few-shot task as an example, and summarize the implementation details in algorithm 1 in detail. Specifically, we first train a Logistic Regression Classifier (LRC) using the support set. Then, we use the trained classifier to make predictions on the query set. Next, we select some query samples with high confidence for each class according to the predictions. Then, we add these selected query samples to expand the support set. Finally, the classifier is re-trained on the expanded support set. We repeat this process several times, and use the last predictions as the final results for the query set. It is worth noting that, in 1-shot setting, since there is only one original support sample for each category, if the selected query samples are directly used to expand the support set of each category, this will cause noise interference. To avoid this problem, we average the expanded support samples of each class. In this way, we always keep only one support sample per class to train the classifier. For the 5-shot setting, we directly use all support samples after expansion to train the classifier.

Algorithm 1 The implementation details of exploiting the full data

Input: Feature extractor f_{θ_s}, Logistic Regression Classifier (LRC), the support set $\mathcal{T}_S$, the query set $\mathcal{T}_Q$
Output: The predictions of $\mathcal{T}_Q$
1: Utilize f_{θ_s} to extract features for all samples
2: Train the LRC using $\mathcal{T}_S$
3: Use the trained LRC to make predictions on $\mathcal{T}_Q$
4: According to the predictions of $\mathcal{T}_Q$, at most ten query samples are selected for each category to expand the support set
5: Retrain LRC using the expended support set
6: Use the trained LRC to predict $\mathcal{T}_Q$
7: Repeat steps 4 to 6 seven times
8: Use the last predictions as the final results for $\mathcal{T}_Q$
9: **return** The predictions of $\mathcal{T}_Q$

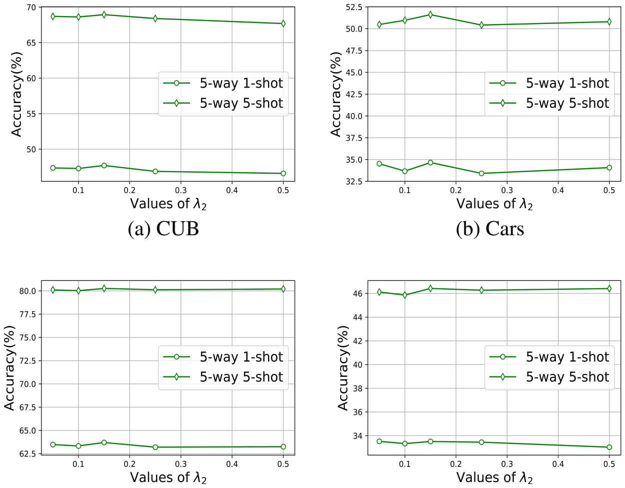
C. Hyper-parameters validation

The hyper-parameters of the proposed method include the coefficient λ_1 of self-image knowledge distillation loss, the coefficient λ_2 of cross-image knowledge distillation loss, the momentum parameter m in cross-episode knowledge distillation, and the size and number of crops. For λ_1, we fix it to 1. We validate the remaining hyper-parameters on the *CUB*, *Cars*, *EuroSAT* and *ISIC* datasets respectively.

For the hyper-parameter λ_2, we evaluate the performance under the values of 0.05, 0.10, 0.15, 0.25 and 0.50 respectively. The experimental results are shown in Fig. 1. Overall, the performance is not sensitive to the value of λ_2. We set λ_2 to 0.15.

For the hyper-parameter m, we vary the values within the set of $\{0.990, 0.993, 0.996, 0.997, 0.998, 0.999\}$. The experimental results are shown in Fig. 2. It can be seen that when m changes between 0.990 and 0.999, the four datasets observe different but minor performance variation. Without otherwise stated, the value m is set to 0.998 uniformly on all the datasets.

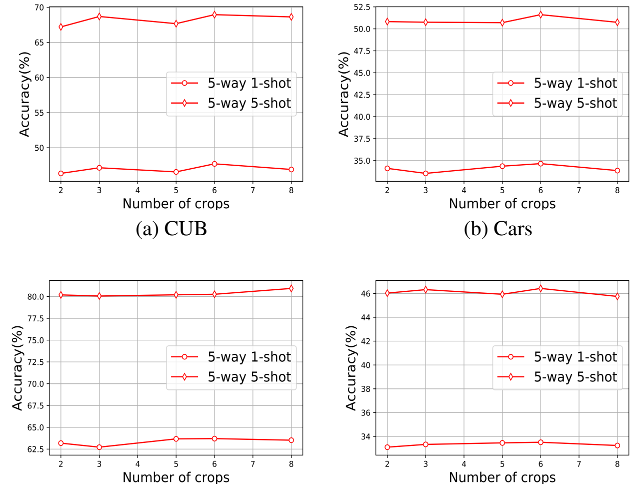
For the size of crops, we evaluate four cases, *i.e.*, 48×48, 72×72, 96×96 and 128×128. The experimental results are shown in Table 1. It can be seen that in most cases, crop



(a) CUB (b) Cars

(c) EuroSAT (d) ISIC

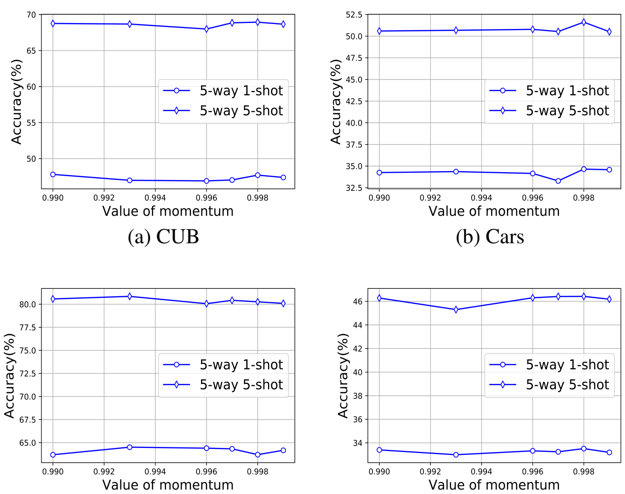
Figure 1. Classification accuracy w.r.t values of λ_2.



(a) CUB (b) Cars

(c) EuroSAT (d) ISIC

Figure 3. Classification accuracy w.r.t the number of crops.



(a) CUB (b) Cars

(c) EuroSAT (d) ISIC

Figure 2. Classification accuracy w.r.t values of momentum.

size of 96×96 achieves superior performance.

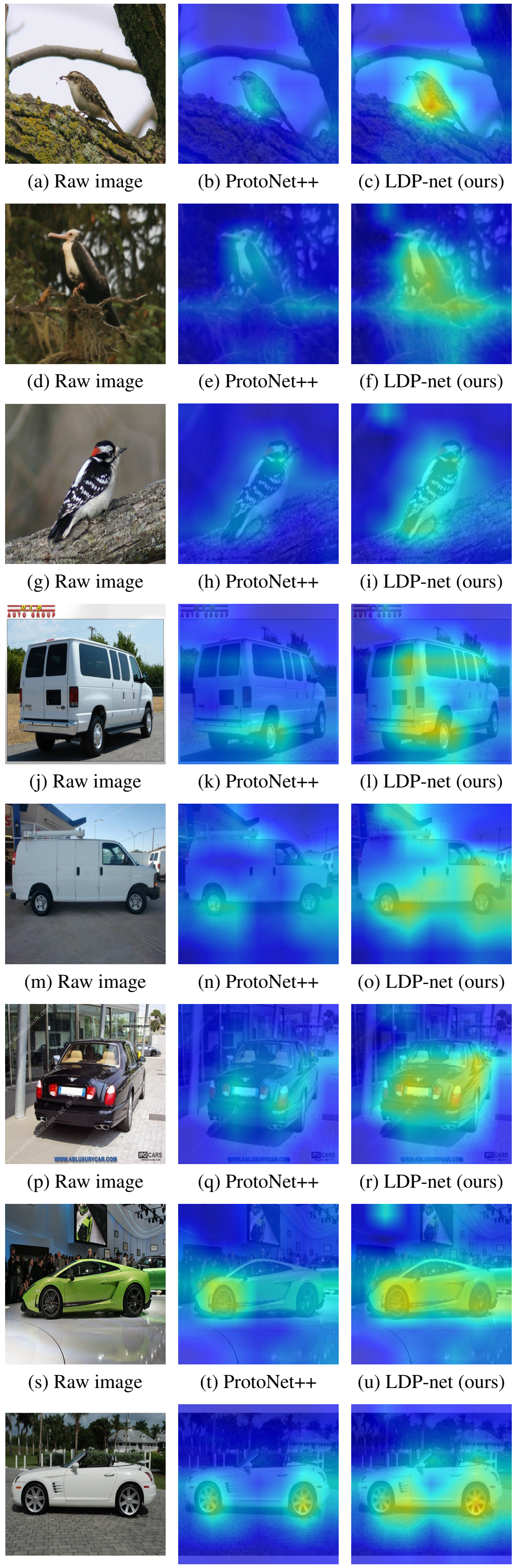
For the number of crops within each image, we evaluate the crop number of 2, 3, 5, 6 and 8 respectively. The experimental results are shown in Fig. 3. Overall, the performance is optimal when the number of crops is 6.

D. More feature visualization results

We provide more feature visualization results in Fig. 4. It can be seen that ProtoNet++ only pays attention to some local regions of the object, *e.g*., Fig. 4 (b), (e), (h), (k), (n), (q), (t), (w). In contrast, the proposed method can focus on a wider range of the object, *e.g*., Fig. 4 (c), (f), (i), (l), (o), (r), (u), (x), which means that the proposed method can capture more comprehensive semantic information and thus generalize better.

E. Results with 95% confidence intervals

In the manuscript, due to space constraints, we did not present the results with 95% confidence intervals in the comparative experiments with the state-of-the-art methods. We add the results with confidence intervals to this supplementary material, as shown in Table 2 and Table 3.



(a) Raw image (b) ProtoNet++ (c) LDP-net (ours)
(d) Raw image (e) ProtoNet++ (f) LDP-net (ours)
(g) Raw image (h) ProtoNet++ (i) LDP-net (ours)
(j) Raw image (k) ProtoNet++ (l) LDP-net (ours)
(m) Raw image (n) ProtoNet++ (o) LDP-net (ours)
(p) Raw image (q) ProtoNet++ (r) LDP-net (ours)
(s) Raw image (t) ProtoNet++ (u) LDP-net (ours)
(v) Raw image (w) ProtoNet++ (x) LDP-net (ours)

Figure 4. Feature visualization for ProtoNet++ and the proposed LDP-net.

Table 1. Classification accuracy w.r.t the crop size. Average classification accuracies (%) are provided. The best results are in bold.

	CUB		**Cars**		**EuroSAT**		**ISIC**	
Crop size	**1-shot**	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
48×48	47.16	68.14	**35.05**	51.36	64.18	80.16	32.54	44.88
72×72	46.67	67.86	34.61	50.28	62.74	80.34	33.38	45.36
96×96	**47.70**	**68.94**	34.65	**51.61**	63.70	80.26	**33.51**	**46.42**
128×128	46.60	68.17	34.15	50.46	**64.96**	**80.81**	33.16	45.65

Table 2. Comparison with state-of-the-art methods. Average classification accuracies (%) are provided.† stands for exploiting the full data of FSL task. * stands for fine-tuning on target domain. The best results are in bold.

			CUB		**Cars**	
Methods	**Mark**	**Ft**	**5-way 1-shot**	**5-way 5-shot**	**5-way 1-shot**	**5-way 5-shot**
MatchingNet [4]	NeurIPS-16	✗	35.89%±0.51%	51.37%±0.77%	30.77%±0.47%	38.99%±0.64%
RelationNet [5]	CVPR-18	✗	41.27%±0.40%	56.77%±0.40%	30.09%±0.30%	40.46%±0.40%
GNN [6]	ICLR-18	✗	44.40%±0.50%	62.87%±0.50%	31.72%±0.40%	43.70%±0.40%
RelationNet+FT [7]	ICLR-20	✗	43.33%±0.40%	59.77%±0.40%	30.45%±0.30%	40.18%±0.40%
RelationNet+ATA [1]	IJCAI-21	✗	43.02%±0.40%	59.36%±0.40%	31.79%±0.30%	42.95%±0.40%
GNN+FT [7]	ICLR-20	✗	45.50%±0.50%	64.97%±0.50%	32.25%±0.40%	46.19%±0.40%
GNN+ATA [1]	IJCAI-21	✗	45.00%±0.50%	66.22%±0.50%	33.61%±0.40%	49.14%±0.40%
MatchingNet+AFA [2]	ECCV-22	✗	41.02%±0.40%	59.46%±0.40%	33.52%±0.40%	46.13%±0.40%
GNN+AFA [2]	ECCV-22	✗	46.86%±0.50%	68.25%±0.50%	34.25%±0.40%	49.28%±0.50%
LDP-net (ours)	-	✗	**49.82%±0.78%**	**70.39%±0.67%**	**35.51%±0.64%**	**52.84%±0.74%**
TPN+ATA† [1]	IJCAI-21	✗	50.26%±0.50%	65.31%±0.40%	34.18%±0.40%	46.95%±0.40%
TPN+AFA† [2]	ECCV-22	✗	50.85%±0.40%	65.86%±0.40%	38.43%±0.40%	47.89%±0.40%
RDC† [3]	CVPR-22	✗	47.77%±0.50%	63.39%±0.40%	**38.74%±0.50%**	52.75%±0.40%
LDP-net† (ours)	-	✗	**55.94%±1.09%**	**73.34%±0.75%**	37.44%±0.88%	**53.06%±0.82%**
Fine-tuning* [8]	ECCV-20	✓	43.53%±0.40%	63.76%±0.40%	35.12%±0.40%	51.21%±0.40%
TPN+ATA*† [1]	IJCAI-21	✓	51.89%±0.50%	70.14%±0.40%	38.07%±0.40%	55.23%±0.40%
RDC*† [3]	CVPR-22	✓	50.09%±0.50%	67.23%±0.40%	39.04%±0.50%	53.49%±0.50%

			Places		**Plantae**	
Methods	**Mark**	**Ft**	**5-way 1-shot**	**5-way 5-shot**	**5-way 1-shot**	**5-way 5-shot**
MatchingNet [4]	NeurIPS-16	✗	49.86%±0.79%	63.16%±0.77%	32.70%±0.60%	46.53%±0.68%
RelationNet [5]	CVPR-18	✗	48.16%±0.50%	64.25%±0.40%	31.23%±0.30%	42.71%±0.30%
GNN [6]	ICLR-18	✗	52.42%±0.50%	70.91%±0.50%	33.60%±0.40%	48.51%±0.40%
RelationNet+FT [7]	ICLR-20	✗	49.92%±0.50%	65.55%±0.40%	32.57%±0.30%	44.29%±0.30%
RelationNet+ATA [1]	IJCAI-21	✗	51.16%±0.50%	66.90%±0.40%	33.72%±0.30%	45.32%±0.30%
GNN+FT [7]	ICLR-20	✗	53.44%±0.50%	70.70%±0.50%	32.56%±0.40%	49.66%±0.40%
GNN+ATA [1]	IJCAI-21	✗	53.57%±0.50%	75.48%±0.40%	34.42%±0.40%	52.69%±0.40%
MatchingNet+AFA [2]	ECCV-22	✗	**54.66%±0.50%**	68.87%±0.40%	37.60%±0.40%	52.43%±0.40%
GNN+AFA [2]	ECCV-22	✗	54.04%±0.60%	**76.21%±0.50%**	36.76%±0.40%	54.26%±0.40%
LDP-net (ours)	-	✗	53.82%±0.84%	72.90%±0.70%	**39.84%±0.75%**	**58.49%±0.69%**
TPN+ATA† [1]	IJCAI-21	✗	57.03%±0.50%	72.12%±0.40%	39.83%±0.40%	55.08%±0.40%
TPN+AFA† [2]	ECCV-22	✗	60.29%±0.50%	72.81%±0.40%	40.27%±0.40%	55.67%±0.40%
RDC† [3]	CVPR-22	✗	58.82%±0.50%	72.83%±0.40%	**41.88%±0.50%**	55.30%±0.40%
LDP-net† (ours)	-	✗	**62.21%±1.13%**	**75.47%±0.73%**	41.04%±0.94%	**59.64%±0.77%**
Fine-tuning* [8]	ECCV-20	✓	50.57%±0.40%	70.68%±0.40%	38.77%±0.40%	56.45%±0.40%
TPN+ATA*† [1]	IJCAI-21	✓	57.26%±0.50%	73.87%±0.40%	40.75%±0.40%	59.02%±0.40%
RDC*† [3]	CVPR-22	✓	61.17%±0.60%	74.91%±0.40%	41.30%±0.60%	57.47%±0.40%

Table 3. Comparison with state-of-the-art methods. Average classification accuracies (%) are provided.† stands for exploiting the full data of FSL task. * stands for fine-tuning on target domain. The best results are in bold.

			Chest		ISIC	
Methods	**Mark**	**Ft**	**5-way 1-shot**	**5-way 5-shot**	**5-way 1-shot**	**5-way 5-shot**
MatchingNet [4]	NeurIPS-16	✗	-	22.40%±0.70%	-	36.74%±0.53%
MAML [9]	ICML-17	✗	-	23.48%±0.96%	-	40.13%±0.58%
RelationNet [5]	CVPR-18	✗	21.95%±0.20%	24.07%±0.20%	30.53%±0.30%	38.60%±0.30%
MetaOptNet [10]	CVPR-19	✗	-	22.53%±0.91%	-	36.28%±0.50%
GNN [6]	ICLR-18	✗	21.94%±0.20%	23.87%±0.20%	30.14%±0.30%	42.54%±0.40%
RelationNet+FT [7]	ICLR-20	✗	21.79%±0.20%	23.95%±0.20%	30.38%±0.30%	38.68%±0.30%
RelationNet+ATA [1]	IJCAI-21	✗	22.14%±0.20%	24.43%±0.20%	31.13%±0.30%	40.38%±0.30%
GNN+FT [7]	ICLR-20	✗	22.00%±0.20%	24.28%±0.2%	30.22%±0.30%	40.87%±0.40%
GNN+ATA [1]	IJCAI-21	✗	22.10%±0.20%	24.32%±0.4%	33.21%±0.40%	44.91%±0.40%
MatchingNet+AFA [2]	ECCV-22	✗	22.11%±0.20%	23.18%±0.20%	32.32%±0.30%	39.88%±0.30%
GNN+AFA [2]	ECCV-22	✗	22.92%±0.20%	25.02%±0.20%	33.21%±0.30%	46.01%±0.40%
LDP-net (ours)	-	✗	**23.01%±0.43%**	**26.67%±0.43%**	**33.97%±0.64%**	**48.06%±0.61%**
TPN+ATA† [1]	IJCAI-21	✗	21.67%±0.20%	23.60%±0.20%	**34.70%±0.40%**	45.83%±0.30%
TPN+AFA† [2]	ECCV-22	✗	21.69%±0.10%	23.47%±0.20%	34.25%±0.40%	46.29%±0.30%
RDC† [3]	CVPR-22	✗	**22.66%±0.20%**	25.10%±0.20%	32.29%±0.30%	42.10%±0.30%
LDP-net† (ours)	-	✗	22.21%±0.44%	**26.88%±0.46%**	33.44%±0.73%	**48.44%±0.67%**
Fine-tuning* [8]	ECCV-20	✓	22.13%±0.20%	25.37%±0.20%	34.60%±0.30%	49.51%±0.30%
NSAE(CE+CE)* [11]	ICCV-21	✓	-	27.10%±0.40%	-	54.05%±0.60%
ConFeSS* [12]	ICLR-22	✓	-	27.09%	-	48.85%
TPN+ATA*† [1]	IJCAI-21	✓	22.45%±0.20%	24.74%±0.20%	35.55%±0.40%	49.83%±0.30%
RDC*† [3]	CVPR-22	✓	22.32%±0.20%	25.07%±0.20%	36.28%±0.40%	49.91%±0.30%

			EuroSAT		CropDisease	
Methods	**Mark**	**Ft**	**5-way 1-shot**	**5-way 5-shot**	**5-way 1-shot**	**5-way 5-shot**
MatchingNet [4]	NeurIPS-16	✗	-	64.45%±0.63%	-	66.39%±0.78%
MAML [9]	ICML-17	✗	-	71.70%±0.72%	-	78.05%±0.68%
RelationNet [5]	CVPR-18	✗	49.08%±0.40%	65.56%±0.40%	53.58%±0.40%	72.86%±0.40%
MetaOptNet [10]	CVPR-19	✗	-	64.44%±0.73%	-	68.41%±0.73%
GNN [6]	ICLR-18	✗	54.61%±0.50%	78.69%±0.40%	59.19%±0.50%	83.12%±0.40%
RelationNet+FT [7]	ICLR-20	✗	53.53%±0.40%	69.13%±0.40%	57.57%±0.50%	75.78%±0.40%
RelationNet+ATA [1]	IJCAI-21	✗	55.69%±0.50%	71.02%±0.40%	61.17%±0.50%	78.20%±0.40%
GNN+FT [7]	ICLR-20	✗	55.53%±0.50%	78.02%±0.40%	60.74%±0.50%	87.07%±0.40%
GNN+ATA [1]	IJCAI-21	✗	61.35%±0.50%	83.75%±0.40%	67.47%±0.50%	**90.59%±0.30%**
MatchingNet+AFA [2]	ECCV-22	✗	61.28%±0.50%	69.63%±0.50%	60.71%±0.50%	80.07%±0.40%
GNN+AFA [2]	ECCV-22	✗	63.12%±0.50%	**85.58%±0.40%**	67.61%±0.50%	88.06%±0.30%
LDP-net (ours)	-	✗	**65.11%±0.92%**	82.01%±0.64%	**69.64%±0.85%**	89.40%±0.51%
TPN+ATA† [1]	IJCAI-21	✗	65.94%±0.50%	79.47%±0.30%	77.82%±0.50%	88.15%±0.50%
TPN+AFA† [2]	ECCV-22	✗	66.17%±0.40%	80.12%±0.40%	72.44%±0.60%	85.69%±0.40%
RDC† [3]	CVPR-22	✗	67.58%±0.50%	79.12%±0.40%	80.88%±0.50%	88.03%±0.30%
LDP-net† (ours)	-	✗	**73.25%±1.13%**	**84.05%±0.66%**	**81.24%±1.05%**	**91.89%±0.50%**
Fine-tuning* [8]	ECCV-20	✓	66.17%±0.50%	81.59%±0.30%	73.43%±0.50%	89.84%±0.30%
NSAE(CE+CE)* [11]	ICCV-21	✓	-	83.96%±0.60%	-	93.14%±0.50%
ConFeSS* [12]	ICLR-22	✓	-	84.65%	-	88.88%
TPN+ATA*† [1]	IJCAI-21	✓	70.84%±0.50%	85.47%±0.30%	82.47%±0.50%	93.56%±0.20%
RDC*† [3]	CVPR-22	✓	70.51%±0.50%	84.29%±0.30%	85.79%±0.50%	93.30%±0.30%

References

[1] Haoqing Wang and Zhi-Hong Deng. Cross-domain few-shot classification via adversarial task augmentation. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1075–1081. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Main Track. 1, 4, 5

[2] Yanxu Hu and Andy J Ma. Adversarial feature augmentation for cross-domain few-shot classification. In *European Conference on Computer Vision*, pages 20–37. Springer, 2022. 1, 4, 5

[3] Pan Li, Shaogang Gong, Chengjie Wang, and Yanwei Fu. Ranking distance calibration for cross-domain few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9099–9108, 2022. 1, 4, 5

[4] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 4, 5

[5] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 4, 5

[6] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. In *6th International Conference on Learning Representations, ICLR 2018*, 2018. 4, 5

[7] Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. Cross-domain few-shot classification via learned feature-wise transformation. In *International Conference on Learning Representations*, 2019. 4, 5

[8] Yunhui Guo, Noel C Codella, Leonid Karlinsky, James V Codella, John R Smith, Kate Saenko, Tajana Rosing, and Rogerio Feris. A broader study of cross-domain few-shot learning. In *European conference on computer vision*, pages 124–141. Springer, 2020. 4, 5

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017. 5

[10] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019. 5

[11] Hanwen Liang, Qiong Zhang, Peng Dai, and Juwei Lu. Boosting the generalization capability in cross-domain few-shot learning via noise-enhanced supervised autoencoder. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9424–9434, 2021. 5

[12] Debasmit Das, Sungrack Yun, and Fatih Porikli. Confess: A framework for single source cross-domain few-shot learning. In *International Conference on Learning Representations*, 2021. 5