Supplemental Materials: Model-Agnostic Graph Regularization for Few-Shot Learning

Anonymous Author(s)
Affiliation
Address
email

Appendix A Problem Statement and Related Work

Episodic Training A common approach is to match the training and evaluation conditions by learning on C_{train} in an episodic manner, called *learning episodes* [21]. Note that training on support set examples during episode evaluation is distinct from training on C_{train}. Many metric learners and optimization-based learners use this training method, including Matching Networks [22], Prototypical Networks [17], Relation Networks [18], and MAML [5].

Non-episodic Baselines Inspired by the transfer learning paradigm of pre-training and fine-tuning, a natural non-episodic approach is to train a classifier on all examples in C_{train} at once. After training, the final classification layer is removed, and this neural network is used as an embedding function f that maps images $\mathbf{x}_i$ to $x_i \in \mathbb{R}$ feature representations, including those from novel classes. It then fine-tunes the final classifier layer using support set examples from the novel classes. The models are a function of the parameters of a softmax layer, $\theta \subset \mathbb{R}^d$. The softmax layer is formulated as the similarity between image feature embeddings and the classifier parameters where θ_j is the parameters for the j^{th} class, sim is the cosine similarity function.

$$p(y_i|x_i;\theta) = \frac{\exp(sim(x_i, \theta_{y_i}))}{\sum_{y' \in \mathcal{Y}} \exp(sim(x_i, \theta_{y'}))} \tag{1}$$

A.1 Related work

Few-Shot Learning Canonical approaches to few-shot learning include memory-based [7, 8, 13], metric learning [15, 17, 18, 22], and optimization-based methods [5, 16]. However, recent studies have shown that simple baseline learning techniques (i.e. simply training a backbone, then fine-tuning the output layer on a few labeled examples) outperform or match performance of many meta-learning methods [2, 4], prompting a closer look at the tasks [21] and contexts in which meta-learning is helpful for few-shot learning [14, 20].

Few-Shot Learning with Graphs Beyond the canonical few-shot literature, studies have explored learning GNNs over episodes as partially observed graphical models [6] and using GCNs to transfer knowledge of semantic labels and categorical relationships to unseen classes in zero-shot learning [23]. Recently, Chen et al. presented a knowledge graph transfer network (KGTN), which uses a Gated Graph Neural Network (GGNN) to propagate information from base categories to novel categories for few-shot learning [1]. Other works use domain knowledge graphs to provide task specific customization [19], and propagate prototypes [10, 11]. However, these models have highly complex architectures and consist of multiple sub-modules that all seem to impact performance.

Submitted to the 4th Workshop on Meta-Learning at NeurIPS 2020, Vancouver, Canada. Do not distribute.

Appendix B Experimental Setup

B.1 Mini-ImageNet

Dataset The Mini-ImageNet dataset is a subset of ILSVRC-2012 [3]. The classes are randomly split into 64, 16 and 20 classes for meta-training, meta-validation, and meta-testing respectively. Each class contains 600 images. We use the commonly-used split proposed in [22].

Training details We pre-train the feature extractor on $\mathcal{C}_{\text{train}}$ using the method proposed by [12]. Activations in the penultimate layer are pre-computed and saved as feature embeddings of 640 dimensions to simplify the fine-tuning process. For an N-way K-shot problem, we sample N novel classes per episode, sample K support examples from those classes, and sample 15 query examples. During pre-training and meta-training stages, input images are normalized using the mean and standard-deviation computed on ILSVRC-2012. We apply standard data augmentation including random crop, left-right flip, and color jitter in both the training or meta-training stage. We use ResNet-18, ResNet-50 [9], and WRN-28-10 [24] for our backbone architectures. For pre-training WRN-28-10, we follow the original hyperparameters and training procedures for S2M2$_R$ [12]. For meta-training ResNet-18, we follow the hyperparameters from [2]. At evaluation time, we choose hyperparameters based on performance on the meta-validation set. Some implementation details are adjusted for each method. Specifically, for ProtoNet and LEO, we include base examples during an additional adaptation step per class. We show that these alterations have a minimal contribution to performance in Appendix C.

B.2 ImageNet-FS

Dataset In the ImageNet-FS benchmark task, the 1000 ILSVRC-2012 categories are split into 389 base categories and 611 novel categories. From these, 193 of the base categories and 300 of the novel categories are used during cross-validation and the remaining 196 base categories and 311 novel categories are used for the final evaluation. Each base category has around 1,280 training images and 50 test images.

Training details We follow the procedure by [8] to pre-train the ResNet-50 feature extractor, and adopt the Square Gradient Magnitude loss to regularize representation learning, which we scale by 0.005. The model is trained using the SGD algorithm with a batch size of 256, momentum of 0.9 and weight decay of 0.0005. The learning rate is initialized as 0.1 and is divided by 10 for every 30 epochs. During fine-tuning, we train for 10,000 iterations using the SGD algorithm with a batch size of 256, momentum of 0.9, weight decay of 0.005, and learning rate of 0.01.

B.3 Label Graph

WordNet ontology ImageNet comprises of 82,115 ‘synsets’, which are based on the WordNet ontology. For both the Mini-ImageNet and ImageNet-FS experiments, we first choose the synsets corresponding to the output classes of each task – 100 for Mini-ImageNet and 1000 for ImageNet-FS. ImageNet provides IS-A relationships over the synsets, defining a DAG over the classes. We only consider the sub-graph consisting of the chosen classes and their ancestors. The classes are all leaves of the DAG.

Training details The hyperparameter settings used for the node2vec-based graph regularization objective are in line with typical values. For all experiments, we set $p = 1, q = 1$ and temperature $T = 2$. We set the batch size to 128 for Mini-ImageNet and 256 for ImageNet-FS. Empirically, we find that setting the regularization λ scaling higher for lower shots results in better performance, and set $\lambda = 5, 3, 1$ for 1-,2-, and 5-shot tasks respectively.

Appendix C Ablations

C.1 Mini-ImageNet Ablations

C.1.1 Model re-implementations with adaptation

For episodically-evaluated few-shot models, it is common practice to disregard base classes during evaluation. To implement graph regularization, we include both base and novel classes during test time and perform a further adaptation step per task. We show that the boost in performance is not due to these modifications.

Table 1: Validation of baseline model modifications.

Model	Backbone	1-shot	5-shot
ProtoNet	ResNet-18	54.16 ± 0.82	73.68 ± 0.65
ProtoNet (adaptation)†	ResNet-18	54.86 ± 0.73	74.14 ± 0.50
ProtoNet (adaptation) + Graph (Ours)	ResNet-18	$\mathbf{55.47 \pm 0.73}$	$\mathbf{74.56 \pm 0.49}$
LEO†	WRN 28-10	58.22 ± 0.09	74.46 ± 0.19
LEO (adaptation)	WRN 28-10	57.85 ± 0.20	74.25 ± 0.17
LEO (adaptation) + Graph (Ours)	WRN 28-10	$\mathbf{60.93 \pm 0.19}$	$\mathbf{76.33 \pm 0.17}$

C.1.2 Finding good parameter initializations for novel classes

Recent works have shown that good parameter initialization is important for few-shot adaptations [14]. For example, Dhillion et al. [4] showed that initializing novel classifiers with the mean of the support set improves few-shot performance.

Here, we explore various methods of incorporating graph relations to improve parameter initialization for novel classes. We compare our proposed method with simpler methods to show that the our graph regularization method is boosting performance in a non-trivial manner. For each method, we keep the adaptation procedure the same, namely, the fine-tuning procedure described by Baseline++ [2].

We then vary parameter initialization using the following methods: (A) random initialization, (B) initializing novel classes with the weights of the closest training class in graph distance in the knowledge graph, (C) our method.

Table 2: Mini-Imagenet with different parameter initialization methods (in % measured over 600 evaluation iterations).

Model	Backbone	1-shot	5-shot
S2M2$_R$ + Init A [12]	WRN 28-10	64.93 ± 0.18	83.18 ± 0.11
S2M2$_R$ + Init B	WRN 28-10	65.50 ± 0.81	83.32 ± 0.57
S2M2$_R$ + Init C	WRN 28-10	$\mathbf{66.93 \pm 0.65}$	$\mathbf{83.35 \pm 0.53}$

C.2 ImageNet-FS Ablations

Here, we justify our model design decisions by considering alternatives. We first probe the benefits of using random walk neighborhoods by defining $N(y)$ as only nodes that have direct edges with y ("child-parent loss"). We try separately learning label graph embeddings, and passing the information to the classifier layer via "soft target" classification loss ("Independent graph w/ soft targets"). Results show that computing the graph loss directly on the classifier parameters is important for performance. Finally, we show that the quality of the label graph affects performance by removing layers of internal nodes of the WordNet hierarchy, starting from the bottom-most nodes ("Remove last 5, 10 layers").

Table 3: Imagenet-FS ablations. Experiment setups, in order from the top: our proposed method, using only child-parent edges, independently learning graph embeddings, removing 5 layers of the ImageNet hierarchy, and removing 10 layers of the ImageNet hierarchy.

Ablation	**1-shot**
Ours	**61.09**
Child-parent loss	56.78
Independent graph w/ soft targets	56.22
Remove last 5 layers	57.80
Remove last 10 layers	54.86

References

[1] R. Chen, T. Chen, X. Hui, H. Wu, G. Li, and L. Lin. Knowledge graph transfer network for few-shot recognition. *arXiv preprint arXiv:1911.09579*, 2019.

[2] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[4] G. S. Dhillon, P. Chaudhari, A. Ravichandran, and S. Soatto. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*, 2019.

[5] C. Finn, P. Abbeel, and S. 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.

[6] V. Garcia and J. Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

[7] S. Gidaris and N. Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018.

[8] B. Hariharan and R. Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3018–3027, 2017.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] L. Liu, T. Zhou, G. Long, J. Jiang, L. Yao, and C. Zhang. Prototype propagation networks (ppn) for weakly-supervised few-shot learning on category graph. *arXiv preprint arXiv:1905.04042*, 2019.

[11] L. Liu, T. Zhou, G. Long, J. Jiang, and C. Zhang. Learning to propagate for graph meta-learning. In *Advances in Neural Information Processing Systems*, pages 1039–1050, 2019.

[12] P. Mangla, N. Kumari, A. Sinha, M. Singh, B. Krishnamurthy, and V. N. Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 2218–2227, 2020.

[13] S. Qiao, C. Liu, W. Shen, and A. L. Yuille. Few-shot image recognition by predicting parameters from activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7229–7238, 2018.

[14] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.

[15] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. 2016.

[16] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.

[17] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.

[18] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. 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.

[19] Q. Suo, J. Chou, W. Zhong, and A. Zhang. Tadanet: Task-adaptive network for graph-enriched meta-learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1789–1799, 2020.

[20] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola. Rethinking few-shot image classification: a good embedding is all you need? *arXiv preprint arXiv:2003.11539*, 2020.

[21] E. Triantafillou, T. Zhu, V. Dumoulin, P. Lamblin, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P.-A. Manzagol, and H. Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv preprint arXiv:1903.03096*, 2019.

[22] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[23] X. Wang, Y. Ye, and A. Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6857–6866, 2018.

[24] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.