Model-Agnostic Graph Regularization for Few-Shot Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

In many domains, relationships between categories are encoded in the knowledge graph. Recently, promising results have been achieved by incorporating knowledge graphs as a side-information in hard classification tasks with severely limited data. However, prior models consist of highly complex architectures with many sub-components that all seem to impact performance. In this paper, we present a comprehensive empirical study on graph embedded few-shot learning. We introduce a graph regularization approach that allows deeper understanding of the impact of incorporating graph information between labels. Our proposed regularization is widely applicable and model-agnostic, and boosts performance of any few-shot learning model, including metric-learning, meta-learning, and fine-tuning. Our approach improves strong base learners by up to 2% on Mini-ImageNet and 6.7% on ImageNet-FS, outperforming state-of-the-art models and other graph embedded methods. Additional analyses reveal that graph regularizing models results in lower loss for more difficult tasks such as lower-shot and less informative few-shot episodes.

1 Introduction

Few-shot learning refers to the task of generalizing from a very few examples, an ability that humans have but machines lack. Recently, major breakthroughs have been achieved with meta-learning, which leverages prior experience from many related tasks to effectively learn to adapt to unseen tasks [2, 24]. At a high level, meta-learning has been divided into metric-based approaches that learn a transferable metric across tasks [25, 26, 29], and optimization-based approaches that learn initializations for fast adaptation on new tasks [6, 22]. Beyond meta-learning, transfer learning by pretraining and fine-tuning on novel tasks has achieved surprisingly competitive performance on few-shot tasks [4, 5, 31].

In many domains, external knowledge about the class labels can be used. For example, this information is crucial in the zero-shot learning paradigm, which seeks to generalize to novel classes without seeing any training examples [10, 12, 33]. Prior knowledge often takes the form of a knowledge graph [30], such as the WordNet hierarchy [19] in computer vision tasks, or GeneOntology [1] in biology. In such cases, relationships between categories in the graph are used to transfer knowledge from base to novel classes. This idea dates back to hierarchical classification [11, 23].

Recently, few-shot learning methods have been enhanced with graph information, achieving state-of-the-art performance on benchmark image classification tasks [3, 13, 14, 15, 27]. Proposed methods typically employ sophisticated and highly parameterized graph models on top of convolutional feature extractors. However, the complexity of these methods prevents deeper understanding of the impact of incorporating graph information. Furthermore, these models are inflexible and incompatible with other approaches in the rapidly-improving field of meta-learning, demonstrating the need for a model-agnostic graph augmentation method.

Here, we conduct a comprehensive empirical study of incorporating knowledge graph information into few-shot learning. First, we introduce a *graph regularization* approach for incorporating graph relationships between labels applicable to any few-shot learning method. Motivated by node embedding [7] and graph regularization principles [8], our proposed regularization enforces category-level representations to preserve neighborhood similarities in a graph. By design, it allows us to directly measure benefits of enhancing few-shot learners with graph information. We incorporate our proposed regularization into three major approaches of few-shot learning: (i) metric-learning, represented by Prototypical Networks [25], (ii) optimization-based learning, represented by LEO [22], and (iii) fine-tuning, represented by SGM [21] and S2M2$_R$ [17]. We demonstrate that graph regularization consistently improves each method and can be widely applied whenever category relations are available. Next, we compare our approach to state-of-the-art methods, including those that utilize the same category hierarchy on standard benchmark Mini-ImageNet and large-scale ImageNet-FS datasets. Remarkably, we find that our approach improves the performance of strong base learners by as much as 6.7% and outperforms graph embedded baselines, even though it is simple, easy to tune, and introduces minimal additional parameters. Finally, we explore the behavior of incorporating graph information in controlled synthetic experiments. Our analysis shows that graph regularizing models yields better decision boundaries in lower-shot learning, and achieves significantly higher gains on more difficult few-shot episodes.

2 Model-Agnostic Graph Regularization

Our approach is a model-agnostic graph regularization objective that is based on the idea that the graph structure of class labels can guide learning of model parameters. The graph regularization objective ensures labels in the same graph neighborhood have similar parameters. The regularization is combined with a classification loss to form the overall objective. The classification term is flexible and depends on the base learner. For instance, the classification term can correspond to cross-entropy on classifier layer outputs or distance between example embeddings and class prototoypes.

2.1 Problem Setup

We assume that we are given a dataset defined as a pair of examples $X \subseteq \mathcal{X}$ with corresponding labels $Y \subseteq \mathcal{Y}$. We say that point $\mathbf{x}_i \in X$ has the label $y_i \in Y$. For each episode, we learn from a support set $\mathcal{D}_s = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_K, y_K)\}$ and evaluate on a held-out query set $\mathcal{D}_q = \{(\mathbf{x}_1^*, y_1^*), (\mathbf{x}_2^*, y_2^*), ..., (\mathbf{x}_T^*, y_T^*)\}$, $\mathcal{Q} \cap \mathcal{S} = \emptyset$. For each dataset, we split all classes into C_{train} and C_{test}, $C_{train} \cap C_{test} = \emptyset$. During evaluation, we sample the N classes from a larger set of classes C_{test}, and sample K examples from each class. During training, we use a disjoint set of classes C_{train} to train the model. Non-episodic training approaches treat C_{train} as a standard supervised learning problem, while episodic training approaches match the conditions which the model is trained and evaluated by sampling episodes from C_{train}. More details on the problem setup can be found in Appendix **??**. Additionally, we assume that there exists side information about the labels in the form of a graph $G(\mathcal{Y}, E)$ where $\mathcal{Y}$ is the set of all nodes in the label graph, and E is the set of edges.

2.2 Regularization

We incorporate graph information using the random walk-based node2vec objective [7]. Random walk methods for graph embedding [20] are fit by maximizing the probability of predicting the neighborhoods for each target node in the graph. Node2vec performs biased random walks by introducing hyperparameters to balance between breadth-first search (BFS) and depth-first search (DFS) to capture local structures and global communities. We formulate the node2vec loss below:

$$\mathcal{L}_{graph}(G, \theta) = -\sum_{y \in \mathcal{Y}} \Bigg[-\log Z_y + \sum_{n \in N(y)} \frac{1}{T} sim(\theta_n, \theta_y) \Bigg], \tag{1}$$

where θ are node representations, sim is a similarity function between the nodes, $N(y)$ is the set of neighbor nodes of node y, T is the temperature hyperparameter, and Z_y is partition function defined as $Z_y = \sum_{v \in \mathcal{Y}} \exp(\frac{1}{T} sim(\theta_y, \theta_v))$. The partition function is approximated using negative sampling [18]. Neighborhood $N(y)$ is obtained by performing a random walk starting from a source node y. The similarity function sim depends on the base learner, which we outline in Section 2.3.

2.3 Augmentation Strategies

Our graph-regularization framework is model-agnostic, and is intuitively applicable to a wide variety of few-shot approaches. Here, we describe augmentation strategies for high-performing learners from metric-learning, meta-learning, and fine-tuning by formulating each as a joint learning objective.

2.3.1 Augmenting Metric-Learning Models

Metric-learning approaches learn an embedding function to compare query set examples. Prototypical networks are a high-performing learner of this class, especially when controlling for model complexity [4, 28]. Prototypical networks construct a prototype p_j of the j^{th} class by taking the mean of support set examples, and comparing query examples using euclidean distance. We regularize these prototypes so they respect class similarities and get the joint objective:

$$\sum_{(x_i,y_i)\in\mathcal{D}_s}\left[||x_i - p_{y_i}||_2^2 + \sum_{y'\in\mathcal{Y}}\exp(-||x_i - p_{y'}||_2^2)\right] + \lambda\mathcal{L}_{graph}(G,\theta). \tag{2}$$

We set the graph similarity function to negative euclidean distance, $sim(p_i, p_j) = -||p_i - p_j||_2^2$. Note that our approach can easily be extended to other metric-learners, for example regularizing the output of the relation module for Relation Networks [26].

2.3.2 Augmenting Meta-Learning Models

Optimization-based meta-learners such as MAML [6] and LEO [22] consist of two optimization loops: the outer loop updates the neural network parameters to an initialization that enables fast adaptation, while the inner loop performs a few gradient updates over the support set to adapt to the new task. Graph regularization enforces class similarities among parameters during inner-loop adaptation.

Specifically for LEO, support set examples are passed through an encoder to produce latent class encodings z, which are decoded to generate classifier parameters θ. Given instantiated model parameters learned from the outer loop, gradient steps are taken in the latent space to get z' while freezing all other parameters to produce final adapted parameters θ'. For more details, please refer to [22]. Concretely, we obtain the joint regularized objective below for the inner-loop adaptations:

$$\sum_{(x_i,y_i)\in\mathcal{D}_s}\left[-z_{y_i}^T x_i + \sum_{y'\in\mathcal{Y}}\exp(z_{y_i}^T x_i)\right] + \lambda\mathcal{L}_{graph}(G,z). \tag{3}$$

We set the graph similarity function to the inner product, $sim(z_i, z_j) = z_i^T z_j$, though in practice cosine similarity, $sim(z_i, z_j) = z_i^T z_j/||z_i||||z_j||$ results in more stable learning.

2.3.3 Augmenting Fine-tuning Models

Recent approaches such as Baseline++ [4] and S2M2$_R$ [17] have demonstrated remarkable performance by pre-training a model on the training set, and fine-tuning the classifier parameters θ on the support set of each task. We follow [4] and freeze the feature embedding model during fine-tuning, though the model can be fine-tuned as well [5]. We perform graph regularization on the classifiers which are learned for novel classes during fine-tuning, resulting in the objective below:

$$\sum_{(x_i,y_i)\in\mathcal{D}_s}\left[-\frac{x_i^T\theta_{y_i}}{||x_i||||\theta_{y_i}||} + \sum_{y'\in\mathcal{Y}}\exp\left(\frac{x_i^T\theta_{y_i}}{||x_i||||\theta_{y_i}||}\right)\right] + \lambda\mathcal{L}_{graph}(G,\theta). \tag{4}$$

We set the graph similarity to cosine similarity, $sim(\theta_i, \theta_j) = \theta_i^T\theta_j/||\theta_i||||\theta_j||$.

3 Experimental Results

For all ImageNet experiments use the associated WordNet [19] category hierarchy to define graph relationships between classes. Details of the experimental setup are given in Appendix **??**. On the synthetic dataset, we analyze the effect of graph regularizing few-shot methods.

3.1 Mini-ImageNet Experiment

We compare performance to few-shot baselines and graph embedded approach KGTN on the Mini-Imagenet experiment. We utilize S2M2$_R$, a strong baseline fine-tuning model. Table 1 shows graph regularization results on Mini-ImageNet compared to results from state-of-the-art models. When S2M2$_R$ is enhanced with graph regularization, it outperforms all other methods on both 1- and 5-shot tasks. As an additional baseline, we consider KGTN that also relies the WordNet hierarchy. To ensure that our improvements are not caused by the embedding function, we load the KGTN feature extractor with a checkpoint pretrained using S2M2$_R$. This demonstrates that even when controlling for improvements in the feature extractor, our simple graph regularization method still outperforms the complex graph-embedded model.

Table 1: Results on 1-shot and 5-shot classification on the Mini-ImageNet dataset. We report average accuracy over 600 randomly sampled episodes. We show graph-based models in the bottom section.

Model	**Backbone**	**1-shot**	**5-shot**
Qiao [21]	WRN 28-10	59.60 ± 0.41	73.74 ± 0.19
Baseline++ [4]	WRN 28-10	59.62 ± 0.81	78.80 ± 0.61
LEO (train+val) [22]	WRN 28-10	61.76 ± 0.08	77.59 ± 0.12
ProtoNet [25]	WRN 28-10	62.60 ± 0.20	79.97 ± 0.14
MatchingNet [29]	WRN 28-10	64.03 ± 0.20	76.32 ± 0.16
S2M2$_R$ [17]	WRN 28-10	64.93 ± 0.18	83.18 ± 0.11
SimpleShot [31]	WRN 28-10	65.87 ± 0.20	82.09 ± 0.14
KGTN [3]	WRN 28-10	65.71 ± 0.75	81.07 ± 0.50
S2M2$_R$ + Graph (Ours)	WRN 28-10	$\mathbf{66.93 \pm 0.65}$	$\mathbf{83.35 \pm 0.53}$

3.2 Graph Regularization is Model-agnostic

We augment ProtoNet [25], LEO [22], and S2M2$_R$ [17] approaches with graph regularization and evaluate effectiveness of our approach on the Mini-ImageNet dataset. These few-shot learning models are fundamentally different and vary in both optimization and training procedures. For example, ProtoNet and LEO both train episodically, while S2M2$_R$ train non-episodically. However, the flexibility of our graph regularization loss allows us to easily extend each method. Table 2 shows the results of graph enhanced few-shot baselines. Graph regularization consistently improves performance of few-shot baselines with larger gains in the 1-shot setup.

Table 2: Performance of graph-regularized few-shot baselines on the Mini-ImageNet dataset. We report average accuracy over 600 randomly sampled episodes.

Model	**Backbone**	**1-shot**	**5-shot**
ProtoNet [25]	ResNet-18	54.16 ± 0.82	73.68 ± 0.65
ProtoNet + Graph	ResNet-18	$\mathbf{55.47 \pm 0.73}$	$\mathbf{74.56 \pm 0.49}$
LEO (train) [22]	WRN 28-10	58.22 ± 0.09	74.46 ± 0.19
LEO + Graph	WRN 28-10	$\mathbf{60.93 \pm 0.19}$	$\mathbf{76.33 \pm 0.17}$
S2M2$_R$ [17]	WRN 28-10	64.93 ± 0.18	83.18 ± 0.11
S2M2$_R$ + Graph (Ours)	WRN 28-10	$\mathbf{66.93 \pm 0.65}$	$\mathbf{83.35 \pm 0.53}$

3.3 Large-Scale Few-Shot Classification

We next evaluate our graph regularization approach on the large-scale ImageNet-FS dataset, which includes 1000 classes. Notably, this task is more challenging because it requires choosing among all novel classes, an arguably more realistic evaluation procedure. We sample *K* images per category, repeat the experiments 5 times, and report mean accuracy with 95% confidence intervals. Results demonstrate that our graph regularization method boosts performance of the SGM baseline [9] by as

much as 6.7%. Remarkably, augmenting SGM with graph regularization outperforms all few-shot baselines, as well as models that benefit from class semantic information and label hierarchy KTCH and KGTN. We include further experimental details in Appendix **??**, and explore further ablations to justify design choices in Appendix **??**.

Table 3: Top-5 accuracy on the novel categories for the Imagenet-FS dataset. KTCH and KGTN are graph-based models. We report 95% confidence intervals over 5 experiment samplings.

Model	**Backbone**	**1-shot**	**2-shot**	**5-shot**
SGM [21]	ResNet-50	54.3	67.0	77.4
MatchingNet [29]	ResNet-50	53.5	63.5	72.7
ProtoNet [25]	ResNet-50	49.6	64.0	74.4
PMN [32]	ResNet-50	53.3	65.2	75.9
KTCH [16]	ResNet-50	58.1	67.3	77.6
KGTN [3]	ResNet-50	60.1	69.4	78.1
SGM + Graph (Ours)	ResNet-50	**61.09** $\pm$ **0.37**	**70.35** $\pm$ **0.17**	**78.61** $\pm$ **0.19**

3.4 Synthetic Experiment

To illustrate the intuitions of graph regularization, we devise a simple few-shot classification problem on a synthetic dataset. We first embed a balanced binary tree of height h using node2vec [7]. We set all leaf nodes as classes, and assign half as base and half as novel. For each task, we sample k support and q query examples from a Gaussian with mean centered at each class embedding and standard deviation σ. Given k support examples, the task is to predict the correct class for query examples among novel classes. In these experiments, we set $h \in \{4, 5, 6, 7\}$, $k \in \{1, 2, ..., 10\}$, $q = 50$, and $\sigma \in \{0.1, 0.2, 0.4\}$. The baseline model is a linear classifier layer with cross-entropy loss, and we apply graph regularization to this baseline. We learn using SGD with learning rate 0.1 for 100 iterations.

We first visualize the learned decision boundaries on identical tasks with and without graph regularization in Figure 1. In this task, the sampled support examples are far away from the query examples, particularly for the purple and green classes. The baseline model learns poor decision boundaries, resulting in many misclassified query examples. In contrast, much fewer query examples are misclassified when graph regularization is applied. Intuitively, graph regularization helps more when the support set is further away from the sampled data points, and thus generalization is harder.

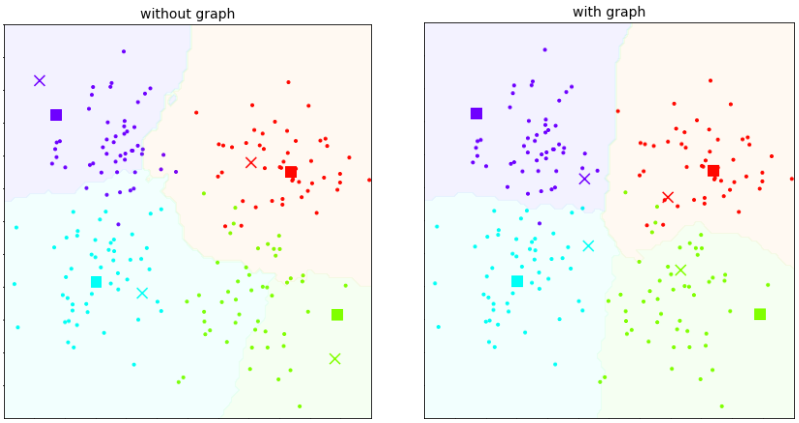

Figure 1: Synthetic experiment results. PCA visualization of learned classifiers for a single task without (left) and with graph regularization (right). Support examples are squares, query examples are dots, learned classifiers are crosses. Shaded regions show decision boundaries.

To measure the relation between few-shot task difficulty and performance, we adopt the hardness metric proposed by [5]. Intuitively, few-shot task hardness depends on the relative location of labeled

and unlabeled examples. If labeled examples are close to the unlabeled examples of the same class, then learned classifiers will result in good decision boundaries and accuracy will be high. Otherwise, accuracy will be low. Given a support set $\mathcal{D}_s$ and query set $\mathcal{D}_q$, the hardness Ω_ϕ is defined as the average log-odds of a query example being classified incorrectly:

$$\Omega_\phi(\mathcal{D}_q; \mathcal{D}_s) = \frac{1}{N_q} \sum_{(x,y)\in\mathcal{D}_q} \log \frac{1 - p(y|x)}{p(y|x)} \tag{5}$$

where $p(\cdot|x_i)$ is a softmax distribution over $sim(x_i, p_j) = -||x_i - p_j||_2^2$, the similarity scores between query examples x_i and the means of the support examples p_j from the j^{th} class in $\mathcal{D}_s$.

We show average loss with shaded 95% confidence intervals across shots in Figure 2, confirming our observations in real-world datasets that graph regularization improves the baseline model the most for tasks with lower shots. In particular, using our synthetic dataset, we can artificially create more difficult few-shot tasks by increasing h, tree heights, and increasing σ, the spread of sampled examples. We plot loss with respect to the proposed hardness metric of each task in Figure 2. The results demonstrate that graph regularization improves performance more on difficult tasks.

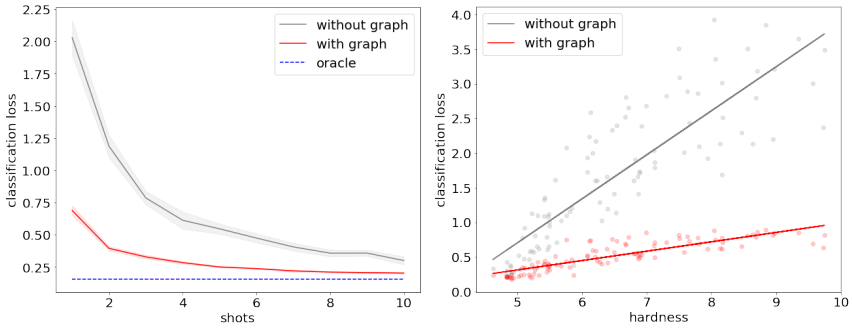

Figure 2: Quantified results of classification loss across shots (left) and task hardness metric (right). Each point is a sampled task. Red is with graph regularization and gray is without.

4 Conclusion

We have introduced a graph regularization method for incorporating label graph side-information into few-shot learning. It is simple and effective, outperforms state-of-the-art graph embedded models, and is a benchmark for any few-shot learning model that integrates label graphs. It is flexible, and can provide a performance boost to a wide range of few-shot learners. A direction of further research is to contribute the first unified experimental setting to compare a range of label graph few-shot methods.

References

[1] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[2] S. Bengio, Y. Bengio, J. Cloutier, and J. Gecsei. On the optimization of a synaptic learning rule. In *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, volume 2, 1992.

[3] 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.

[4] 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.

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

[6] 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.

[7] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[8] D. Hallac, J. Leskovec, and S. Boyd. Network lasso: Clustering and optimization in large graphs. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 387–396, 2015.

[9] 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.

[10] P. Kankuekul, A. Kawewong, S. Tangruamsub, and O. Hasegawa. Online incremental attribute-based zero-shot learning. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3657–3664. IEEE, 2012.

[11] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. Technical report, Stanford InfoLab, 1997.

[12] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):453–465, 2013.

[13] A. Li, T. Luo, Z. Lu, T. Xiang, and L. Wang. Large-scale few-shot learning: Knowledge transfer with class hierarchy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7212–7220, 2019.

[14] 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.

[15] 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.

[16] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019.

[17] 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.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[19] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11): 39–41, 1995.

[20] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[21] 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.

[22] 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.

[23] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR 2011*, pages 1481–1488. IEEE, 2011.

[24] J. Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

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

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] Y. Wang, W.-L. Chao, K. Q. Weinberger, and L. van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019.

[32] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7278–7286, 2018.

[33] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2018.