Meta-Graph: Few shot Link Prediction via Meta-Learning

Avishek Joey Bose [*]
McGill University, Mila

Ankit Jain
Uber AI

Piero Molino
Uber AI

William L. Hamilton
McGill University, Mila

Abstract

Fast adaptation to new data is one key facet of human intelligence and is an unexplored problem on graph-structured data. Few-Shot Link Prediction is a challenging task representative of real world data with evolving sub-graphs or entirely new graphs with shared structure. In this work, we present a meta-learning approach to Few Shot Link-Prediction. We further introduce Meta-Graph, a meta-learning algorithm which in addition to the global parameters learns a Graph Signature function that exploits structural information of a graph compared to other graphs from the same distribution for even faster adaptation and better convergence than vanilla Meta-Learning.

1 Introduction

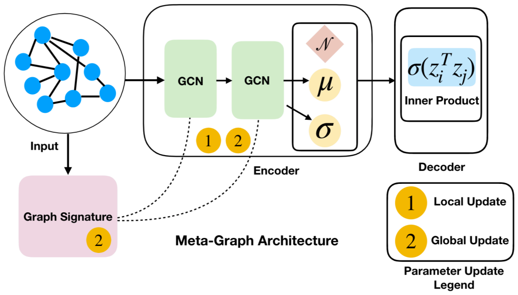
One of the hallmarks of human intelligence is *fast adaptation*, i.e., the ability to learn and adapt to novel tasks when presented with minimal evidence. Gradient based meta-learning approaches [2, 13] attempt to achieve fast adaptation by learning a set of global parameters that are shared across tasks and that can then be used as a good initialization for new, related tasks. These meta-learning approaches have achieved state-of-the-art results across numerous fast adaptation tasks in recent years, e.g., for reinforcement learning [6] and few-shot image classification [6, 13].

However, while meta-learning algorithms have successfully been deployed in traditional deep learning domains such as images [14], or text [10], graph-structured data has received little attention. Meta-learning for graph structured data represents a practical setting for many real-world problems where only a limited set of sub-graphs from a larger graph are available and retraining a model is computationally expensive. Furthermore, when only a sparse subset of edges are observable—e.g., in a new social network—meta-learning can be a viable solution for effective recommendation provided there are training graphs that share structural similarities with the new sparse one.

Present Work. In this work, we adapt the classical gradient-based meta-learning formulation for few shot classification to the graph domain. Specifically, we consider a distribution over graphs as the distribution over tasks from which a global set of parameters is learned. We consider the challenging task of predicting missing links for each graph (i.e., link prediction) when only a small fraction of the edges can be observed. To further bootstrap fast adaptation to new graphs we also introduce a graph signature that utilizes the similarity, from the perspective of meta-learning, between the new graph and the previous graphs seen during training. We experimentally validate our approach on two standard graph datasets. We find that our meta-learning based approach successfully achieves fast adaptation, while also converging to better overall solutions in many experimental settings.

[*] Work done during Uber AI Internship. Correspondence to joey.bose@mail.mcgill.ca

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.



```
Algorithm 1: Meta-Graph for Few Shot Link Prediction
Result: Global Parameters $\theta$, Graph Signature $\psi$
Initialize learning rates: $\alpha, \epsilon$
Sample a mini-batch of Graphs, $\mathcal{G}_{batch}$ from $p(\mathcal{G})$;
for each Graph $\mathcal{G} \in \mathcal{G}_{batch}$ do
    $\mathcal{E} = \mathcal{E}_{\mathcal{G}_{train}} \cup \mathcal{E}_{\mathcal{G}_{val}} \cup \mathcal{E}_{\mathcal{G}_{test}}$ // Split Edges in to train, val, and test
    Initialize: $\phi_0 \leftarrow \theta$ ;
    for $k$ in $1, ..., K$ do
        $\beta, \gamma = \text{stopgrad}(\psi(\mathcal{G}, \mathcal{E}_{\mathcal{G}_{train}}))$ // Compute Graph Signature
        $\mathcal{L}_{train} = \mathbb{E}_q[\log p(A|Z)] - KL[q(Z|\mathcal{E}_{\mathcal{G}_{itrain}}, \beta, \gamma)||p(z)]$
        Update $\phi_k \leftarrow \phi_{k-1} - \alpha \nabla_\phi \mathcal{L}_{train}$
    end
    Initialize: $\theta \leftarrow \phi_K$
    $\beta, \gamma = \psi(\mathcal{G}, \mathcal{E}_{\mathcal{G}_{val}})$ // Compute Graph Signature with val edges
    $\mathcal{L}_{val} = \mathbb{E}_q[\log p(A|Z)] - KL[q(Z|\mathcal{E}_{\mathcal{G}_{val}}, \beta, \gamma)||p(z)]$
    Update $\theta \leftarrow \theta - \epsilon \nabla_\theta \mathcal{L}_{val}$
    Update $\psi \leftarrow \psi - \epsilon \nabla_\psi \mathcal{L}_{val}$
end
```

Figure 1: **Left:** Meta-Graph Architecture. **Right:** Meta-Graph Algorithm for link-prediction on sparse graphs with parameters updates to θ, ϕ, ψ

2 Background

Variational Graph Autoencoder. One of the most prominent approaches to unsupervised learning is the Variational Autoencoder (VAE) [7]. For data structured as a graph, an analogous Variational Graph Autoencoder (VGAE) has been proposed [9]. Formally, given a graph $\mathcal{G} = (V, \mathcal{E})$, with $N = |V|$ nodes, a weight matrix W, an adjacency matrix A and node feature matrix $X \in \mathbb{R}^{N \times D}$, the VGAE learns both an inference model that effectively encodes each node into an embedding vector as well as a generative model that scores the likelihood of an edge existing between pairs of nodes. The parameters of the inference or recognition network are shared across all nodes in $\mathcal{G}$, effectively *amortizing* the inference process needed to define the approximate posterior, $q(z|X, A) = \prod_{i=1}^{N} q(z_i|X, A)$ where $q(z|X, A) = \mathcal{N}(z_i|\mu_i, \text{diag}(\sigma_i^2))$. Conversely, the generative network models is defined by $p(A|Z) = \prod_{i=1}^{N} \prod_{j=1}^{N} p(A_{i,j}|z_i, z_j)$, that is the likelihood of an edge existing between the node pairs. In much the same vein, the overall loss function optimizes for the variational lower bound given by:

$$\mathcal{L}_G = \mathbb{E}_q[\log p(A|Z)] - KL[q(Z|X, A)||p(z)] \tag{1}$$

Optimizing, this lower bound effectively maximizes the log likelihood of the data as the $KL-$divergence is positive by definition.

Meta-Learning. Humans have a remarkable ability to perform new tasks having only been briefly exposed to them. Much of this is attributed to the fact that skills that were learned in previous experiences can be reused, and thus not relearned from scratch, to bootstrap the learning process in a new task. In meta-learning or learning to learn [4, 3, 17, 15], the objective is to learn from prior experiences to form inductive biases for fast adaptation to unseen tasks. While there are many approaches to meta-learning, in this work we focus on a class of approaches known as gradient-based meta-learning, where stochastic gradient descent is used to backpropagate through the learning process itself. Let $\mathcal{D}$ be a dataset which defines a distribution over tasks $\mathcal{T}$ with some shared structure and, each task $\mathcal{T}_j$ defines a distribution over datapoints, x_j, which is to be optimized by the meta-learner to produce task specific parameters. In the few-shot learning setting, the meta-learner observes up to N samples from each task —i.e. $x_{j_1}, ..., x_{j_N} \sim p(\mathcal{T}_j)$ with the goal of finding a set of shared parameters, θ, over tasks. These global parameters are optimized such that when a few gradient descent steps are taken from the initialization, θ, given a small sample of points from $\mathcal{T}_j$ there is good generalization performance on held out samples also from the same task—i.e., $x_{j_{N+1}}, ..., x_{j_{N+m}} \sim p(\mathcal{T}_{j_i})$. That is, starting from the global parameters θ the meta-learning algorithm produces a set of local parameters, ϕ_j tailored to $\mathcal{T}_j$, through fast adaptation. This approach to gradient based meta-learning is known as Model Agonistic Meta-Learning (MAML) [6], and it's overall objective is defined as:

$$\mathcal{L}(\theta) = \frac{1}{J} \sum_{j \in J} \Big[\sum_{m \in M} - \log p(x_{j_{N+m}}|\theta - \alpha \nabla_\theta \frac{1}{N} \sum_{n \in N} - \log p(x_{j_n}|\theta)) \Big]. \tag{2}$$

Here, the local parameters are $\phi_j = \theta - \alpha \nabla_\theta \frac{1}{N} \sum_{n \in N} - \log p(x_{j_n}|\theta)$ with α as the learning rate for the specific task.

3 Method

We consider the problem of link prediction in a sparse graph drawn from a distribution of graphs as the setting for our meta learning problem.

Meta-Graph. We introduce Meta-Graph (Figure 1), a novel meta-learning algorithm that uses an encoding of the current graph as means to modulate the parameters of the recognition network in a VGAE model. Specifically, Meta-Graph learns global and local parameters, θ, ϕ, that are based on the VGAE model comprising of k hidden GCN [8] layers as the recognition network and a dot-product decoder as the generative model. To exploit the shared structural and node feature similarities between graphs, we also define a Graph Signature (GS) function, ψ, which learns a graph embedding γ and a bias β for each GCN layer in the recognition network given a sampled graph, $\mathcal{G}_i \sim p(\mathcal{G})$. Intuitively, the role of ψ is to encode the structural properties within a graph across the distribution over graphs given a new $\mathcal{G}_i \sim p(\mathcal{G})$ it can thus inform the weight updates for even faster adaption. Similar to the recognition network we define ψ using k-GCN layers followed by a small MLP with a non-linear activation such that the output is bounded —i.e. $\gamma, \beta \in [-1, 1]$. Inspired by [5], we use feature-wise linear modulation [16] for each GCN layer in the recognition network as follows:

$$\beta_k, \gamma_k, = \psi(\mathcal{G})$$
$$h_k = \sum_{i \to j \in \mathcal{E}} (\gamma_k \odot W h_{k-1} + \beta_k).$$

To enforce the GS to learn useful modulating parameters γ and β over the entire distribution, we update ψ, only in the outer loop. During meta-training, we use ψ as a deterministic encoding for $\mathcal{G}$, and update local parameters ϕ with a few steps of gradient descent, while ensuring that gradients are never used to update ψ itself. Defining ψ in the outer loop allows us to compute parameters that influence the parameters of the recognition prior to a single gradient step, enabling faster adaptation than vanilla meta-learning which does not explicitly leverage structural and other features of previously observed graphs. Fig. 1 Left shows the architecture of Meta-Graph while Fig. 1 Right gives the exact algorithm used to update global and local parameters, θ, ϕ and the graph signature function ψ.

Connection to MAML. The Meta-Graph algorithm has important differences from MAML and other conventional meta-learning domains. Meta-Graph assumes a distribution over graphs rather than specific tasks. Secondly, examples in our dataset are edges in a sparse graph which can be non-i.i.d. unlike in supervised classification or regression. The most significant difference, is however the addition of the Graph Signature function and its explicit role in influencing the local parameters during meta-training. To be more precise, new test graphs which are similar to training graphs, for the purposes of meta-learning, start at a significantly better initialization point due to feature-wise modulation of the recognition network parameters.

4 Experiments

Datasets. We demonstrate the ability of Meta-Graph on the Protein Protein Interaction (PPI) [18] and FIRSTMM DB [11] datasets taken from the biological and robotics domains respectively. The PPI datasets consists of human protein-protein interaction networks corresponding to different tissues, from which 20 graphs are taken as training, 2 for validation and another 2 for testing. The FIRSTMM DB contains a set of graphs corresponding to 3d point cloud data and categories of various household objects for semantic and graph-based object category prediction and has 33 training graphs and 4 graph each for validation and testing. For both datasets, we perform link prediction by training on a subset (i.e., a percentage) of the edges and then attempting to predict the unseen edges (with 20% of the held-out edges used for validation).

Results. We evaluate Meta-Graph against multiple baselines for both final convergence as well as fast adaptation using the AUC classification accuracy for predicting real vs. randomly sampled negative edges. Table 1 shows convergence results for both datasets for different training edge

	PPI			FirstMM DB		
% Edges	10%	20%	30%	10%	20%	30%
Meta-Graph	**0.795/0.813**	**0.833/0.839**	**0.845/0.846**	**0.782/0.715**	**0.786/0.718**	0.783/0.712
MAML	0.770/0.785	0.815/0.825	0.828/0.834	0.776/0.712	0.782/0.713	**0.793/0.733**
Random	0.578/0.530	0.651/0.590	0.697/0.639	0.742/0.677	0.732/0.665	0.720/0.649
No Fintune	0.738/0.757	0.786/0.803	0.801/0.820	0.740/0.692	0.710/0.646	0.734/0.687
Finetune	0.752/0.759	0.8010/0.817	0.821/0.835	0.752/0.701	0.735/0.690	0.723/0.672
Adamic	0.540/0.540	0.623/0.622	0.697/0.700	0.504/0.504	0.519/0.519	0.544/0.543
Deepwalk	0.664/0.641	0.673/0.669	0.694/0.701	0.487/0.492	0.473/0.525	0.510/0.604

Table 1: Convergence AUC/AP results with various fractions of training edges

	PPI			FirstMM DB		
% Edges	10%	20%	30%	10%	20%	30%
Meta-Graph	**0.795/0.812**	**0.824/0.832**	**0.847/0.849**	**0.773/0.713**	**0.767/0.715**	**0.737/0.667**
MAML	0.728/0.710	0.809/0.816	0.804/0.806	0.763/0.702	0.750/0.672	0.624/ 0.590
No Fintune	0.600/0.547	0.697/0.668	0.717/0.671	0.708/0.644	0.680/0.603	0.709/0.643
Finetune	0.582/0.546	0.727/0.733	0.774/0.788	0.705/0.655	0.695/0.646	0.704/0.633

Table 2: 5-gradient update AUC/AP results with various fractions of training edges

splits. Specifically, when testing for model convergence we adapt to new test graphs until learning converges as determined by performance on the validation set of edges. In addition, we also report in Table 2 results in the fast adaptation setting where each approach has 5-gradient steps to quickly adapt to the new graph. We compare Meta-Graph against a number of classical link prediction baselines: Adamic-Adar [1], DeepWalk [12] and a GCN model with random weights to understand the natural expressive power of the base VGAE model. We also report a NoFinetuning and Finetuning baselines. The former trains a single set of VGAE parameters for each graph, as a result, each model is independent of the other graphs in the dataset. For finetuning the graphs are observed in a sequential order and the weights are finetuned starting from the previous graph in the sequence. Finally, we also compare with a meta-learning baseline which does not include GS, which we call MAML as there are both global and local parameters. For fair comparison we tune all learning rates and meta-learning specific hyperparameters like the number of local updates using Bayesian Optimization with Thompson sampling on a validation set of graphs.

As shown in Table 1 we find that Meta-Graph outperforms all baselines for PPI in terms of final convergence by a significant margin for all training edge splits. A similar result is observed in Table 2 for FIRSTMM DB for 10% and 20% of edges, while for 30% MAML (which itself is a meta-learning algorithm) marginally outperforms Meta-Graph. As meta-learning approaches are purpose built for fast adaptation we find that both Meta-Graph and MAML achieve large performance gains from just 5 gradient updates compared to all other baseline on both PPI and FIRSTMM DB. Furthermore, the addition of GS in Meta-Graph further boosts performance by learning a better initialization point relative to MAML and is superior in all settings.

5 Conclusion

We consider the problem of meta-learning where each task is link prediction in a sparse graph drawn from a distribution over graphs. We introduce a new meta-learning algorithm for sparse graph data that learns an additional graph signature function that modulates the parameters of a recognition network during fast adaptation. Empirically, we observe significant gains on the PPI and FIRSTMM DB datasets when compared to conventional link prediction approaches such as DeepWalk and Adamic-Adar. A similar result is also observed on one popular approach to gradient based meta-learning in MAML for both final convergence but critically fast adaptation in few gradient steps. While, the GS uses an element-wise modulation of the GCN weights it is not the only possible choice. Extending Meta-Graph with different mechanisms for parameter modulation and gaining deeper insight over the learned parameters in GS is a fruitful direction for future work.

Acknowledgements

The authors would like to thank Thang Bui, Maxime Wabartha, Nadeem Ward, Sebastien Lachapelle, and Zhaocheng Zhu for helpful feedback on earlier drafts of this work. In addition, the authors would like to thank the Uber AI team including other interns that helped shape earlier versions of this idea. Joey Bose is supported by the IVADO PhD fellowship and this work was done as part of his internship at Uber AI.

References

[1] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

[2] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pages 3981–3989, 2016.

[3] Samy Bengio, Yoshua Bengio, Jocelyn Cloutier, and Jan Gecsei. On the optimization of a synaptic learning rule. In *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, pages 6–8. Univ. of Texas, 1992.

[4] Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. *Learning a synaptic learning rule*. Université de Montréal, Département d'informatique et de recherche fi, 1990.

[5] Marc Brockschmidt. Gnn-film: Graph neural networks with feature-wise linear modulation. *arXiv preprint arXiv:1906.12192*, 2019.

[6] 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.

[7] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[8] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[9] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[10] Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. Personalizing dialogue agents via meta-learning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 5454–5459, 2019.

[11] Marion Neumann, Plinio Moreno, Laura Antanas, Roman Garnett, and Kristian Kersting. Graph kernels for object category prediction in task-dependent robot grasping. In *Online Proceedings of the Eleventh Workshop on Mining and Learning with Graphs*, pages 0–6, 2013.

[12] Bryan Perozzi, Rami Al-Rfou, and Steven 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. ACM, 2014.

[13] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

[14] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016.

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

[16] Florian Strub, Mathieu Seurin, Ethan Perez, Harm De Vries, Jérémie Mary, Philippe Preux, and Aaron CourvilleOlivier Pietquin. Visual reasoning with multi-hop feature modulation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–800, 2018.

[17] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

[18] Marinka Zitnik and Jure Leskovec. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14):i190–i198, 2017.