# G-META+

**Muhammad Umair, Gian Marco Visani, Mert Erden**

**November 2020**

## 1 Introduction

Recent developments in Graph Neural Networks (GNN) and Meta Learning have led to the new field of meta learning on graphs. Parallel developments have caused a rise in the collection of high throughput data using Biological Interaction Networks. One state of the art Graph Meta Learning framework is G-Meta [1], which is a method of Model-Agnostic Meta-Learning for fast adaptation of deep networks. It works by creating subgraphs from a collection of graphs, and uses the local neighborhood information of these subgraphs to create a prototypical support embedding. This prototypical support embedding is created by averaging the individual subgraph embeddings. Predictions are then made via Probabilistic Nearest Neighbor (PrNN) classification between the prototypical support embedding and the subgraph embedding that we are using to make the prediction. We believe that we can augment this approach to include connectivity information between subgraphs, theoretically avoiding loss of information between subgraphs. Further, we believe that the PrNN classification can be replaced by a more effective technique borrowed from the metric learning literature.

Our approach aims to intelligently incorporate the global graph structure by exploiting the relationship between subgraphs into the existing G-Meta framework. This will generate higher quality support embeddings from the graphs that can then be used for different downstream meta-learning tasks. Additionally, we believe that a more sophisticated metric learning technique will increase the predictive power of the technique.

Overall, we aim to expand the G-Meta framework in two stages. We start by expanding the subgraphs of the support data by incorporating the shortest path between the support subgraphs that we generate. In the second stage, we will substitute G-Meta's existing prototypical network component [2] (which uses PrNN) with a Relation Network [3] component for label prediction. Similar to the Prototypical Networks, these Relation Networks will create support embeddings by adding all support embeddings. Instead of adopting the PrNN approach of prototypical networks, the Relation Network will train a relation function that learns to compare support data with query data.

# 2 Related Work

## 2.1 Few-shot meta-learning

Few-shot meta-learning is a general framework within which models are able to transfer knowledge from prior experience to new but related tasks, allowing them to make predictions with only a few examples. There are three recognized categories of meta-learning techniques: model-based [4], metric-based [2] [3] and optimization-based [5]. Most relevant to our work are prototypical networks [2], relation networks [3] and MAML [5].

## 2.2 Meta learning for graphs

Multiple approaches have been recently developed that apply meta-learning to graphs. Meta-GNN [6] uses gradient-based meta-learning for few-shot node classification within a single graph. As such, it may not be applied to transfer knowledge across different graphs, as is the case for our main task of interest concerned with transferring knowledge from non-human PPI networks to human PPI networks. Instead, GFL [7] and Meta-Graph [8] can meta-learn across multiple graphs. However, Meta-Graph only works on link-prediction, and both require labels to be shared across graphs. In contrast, our approach, which is based on G-Meta, can also meta-learn for different label sets across graphs.

## 2.3 G-Meta

G-Meta is currently the state-of-the-art graph meta learning framework. G-Meta represents every node with a local subgraph and uses these subgraphs to meta-train GNNs via MAML. The use of subgraph-level information as opposed to information from the whole graph is justified by the authors of G-Meta with a proof that evidence for a particular prediction can be found in the subgraph surrounding the target node or edge when using GNNs. However, some meaningful information may still be lost by only focusing on the local neighborhood. We take an intermediate approach by including connectivity information between local subgraphs within the G-Meta framework via the use of an attention mechanism. Our approach recovers useful global network structures lost by G-Meta without the drawbacks derived from using GNNs on the full networks. Furthermore, G-Meta makes predictions using a metric-learning component, specifically borrowing from prototypical networks. Instead, we use the analogous component from relation networks, which have been shown to outperform prototypical networks on several one-shot benchmarks [3].

# 3 Method

## 3.1 Meta Learning

Meta learning on a specific problem (e.g. link prediction) is formally defined on a dataset $\mathcal{D}$ partitioned into $\mathcal{D}_{train}$, $\mathcal{D}_{validate}$, $\mathcal{D}_{test}$, each containing a set of tasks $\tau$. Each task $\tau_i$ in this set can be divided into $\tau_i^{support}$ and $\tau_i^{query}$. $\tau_i^{support}$ contains a collection of $\kappa_{support}$ labeled data points for learning and, $\tau_i^{query}$ contains a collection of data points for evaluation.

In our case we define $\mathcal{D}_{train}$ on Non Human Graphs $G_1, \dots\ G_n$. We aim to use these graphs for Meta Learning on the problems of Link Prediction (Binary Classification) and Node classification (Multi-Label Classification). $\mathcal{D}_{train}$ contains tasks on non-human graphs, while $\mathcal{D}_{test}$ each contain a single task on a human graph. We note that in the general meta-learning formulation $\mathcal{D}_{test}$ may contain an arbitrary number of tasks, but in our case it only contains one. We plan on formulating $\mathcal{D}_{validate}$ on human networks as we want our predictions to have a degree of human network influence for these tasks.

## 3.2 G-Meta

G-Meta applies this meta-learning framework on the node classification and link prediction problems by meta-learning a GNNs on subgraphs. Specifically, G-Meta represents every node with a local subgraph corresponding to the neighborhood (up to h hops form the original node) and uses these subgraphs to meta-train GNNs via MAML. The GNN thus produces vector embeddings for each support and query subgraph.

Then, G-Meta makes predictions using a metric-learning component, specifically borrowing from prototypical networks ref. In particular, a prediction for query embedding $q$ is carried out as follows. For each label $l$, given $K$ support embeddings $s_l^1, \dots\ s_l^K$, G-Meta computes a prototypical support embedding $ps_l = 1/K \sum_{i=1}^{K} s_l^i$. Then, G-Meta computes a probability distribution for q over the labels based on the euclidean distance between q and the prototypical support embedding of each label: $\frac{exp(||q-sp_l||)}{\sum_l exp(||q-sp_l||)}$.

## 3.3 G-Meta+

We expand upon the work of G-Meta and expand on the framework by:

1. Expanding the subgraphs of support data by incorporating the shortest path between support subgraphs. In an attempt to learn about connectivity within the network.

2. Substitute the prototypical network component with a relation network component for label prediction.

To represent a given node, G-Meta uses the subgraph corresponding to the neighborhood surrounding the node. Subgraphs generated this way are used both as support and as query data, and are fed into a GNN. Our approach is to expand the support subgraphs via shortest paths between them. Specifically, for each support subgraph pair $(s_i, s_j)$ coming from the network $n_k$, we also include in $s_i$ and $s_j$ the shortest path in network $n_k$ that joins these subgraphs. Then we feed the subgraphs into the GNN in G-Meta. The primary hypothesis here is that the original subgraphs only capture local information, by allowing messages to propagate from the global graph structure. There has been a lot of work done on diffusion state embeddings that have shown that global structure has impact on the network. We assume that we can recover some of these properties using a shortest path technique.

The relation network component we aim to use works as follows. For each label $l$, given $K$ support embeddings $s_l^1, \ldots\ s_l^K$, we compute a prototypical support embedding $ps_l = \sum_{i=1}^{K} s_l^i$ (the original paper sums the vector instead of averaging them, but maybe we want to average?). Then, $ps_l$ and $q$ are concatenated and fed into a relation module $g_\phi$, parametrized by $\phi$, which outputs a relation score $r_{ql}$ between 0 and 1: $r_{ql} = g_\phi(C(ps_l, q))$. A prediction across labels can be made via argmax($r_{q_l}$ for $l$ in labels). MSE is used as loss function instead of cross entropy, since the prediction of relation scores can be seen as a regression problem. $g_\phi$ may be constructed arbitrarily, but with a sigmoid as final activation to ensure the output is between 0 and 1. We plan on using simple dense layers, and to fine-tune the architecture on validation data. Crucially, the optimization of $g_\phi$ will be included within the already present MAML framework.

# 4 Experimental Plan

We aim to benchmark our results relative to the original work of G-Meta to hopefully show that our new technique outperforms G-Meta. With that being said here is an overall outline of how we will be conducting our experiments.

1. As input we pass in all 1840 biological networks from the tree of life dataset, and a collection of DREAM Challenge and STRING DB human networks. These make up $D_{train}$, $D_{test}$ and $D_{validate}$ respectively.

2. We define 2 Meta Learning Tasks, namely Link Prediction (a binary classification task) and Node Classification (a multi-class classification task).

3. We modify the G-Meta technique with what we've described in the Methods Section.

4. We also run the same experiment on using the unmodified G-Meta for baseline results.

5. The output of steps 3 and 4 are a collection of predictions. We compare this to the ground truth to generate accuracy results.

6. We plan on evaluating the impact of our contributions independently as well as conjointly.

# References

[1] Kexin Huang and Marinka Zitnik. Graph meta learning via local subgraphs. *arXiv preprint arXiv:2006.07889*, 2020.

[2] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.

[3] 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] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks, 2015.

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

[6] Aravind Sankar, Xinyang Zhang, and Kevin Chen-Chuan Chang. Meta-gnn: Metagraph neural network for semi-supervised learning in attributed heterogeneous information networks. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 137–144, 2019.

[7] Huaxiu Yao, Chuxu Zhang, Ying Wei, Meng Jiang, Suhang Wang, Junzhou Huang, Nitesh V Chawla, and Zhenhui Li. Graph few-shot learning via knowledge transfer. *arXiv preprint arXiv:1910.03053*, 2019.

[8] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. Meta-graph based recommendation fusion over heterogeneous information networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 635–644, 2017.