Hierarchical Protein Function Prediction with Tail-GNNs

Stefan Spalević [1] **Petar Veličković** [2] **Jovana Kovačević** [1] **Mladen Nikolić** [1]

Abstract

Protein function prediction may be framed as predicting subgraphs (with certain closure properties) of a directed acyclic graph describing the hierarchy of protein functions. Graph neural networks (GNNs), with their built-in inductive bias for relational data, are hence naturally suited for this task. However, in contrast with most GNN applications, the graph is not related to the input, but to the *label* space. Accordingly, we propose **Tail-GNNs**, neural networks which naturally compose with the output space of any neural network for multi-task prediction, to provide relationally-reinforced labels. For protein function prediction, we combine a Tail-GNN with a dilated convolutional network which learns representations of the protein sequence, making significant improvement in F_1 score and demonstrating the ability of Tail-GNNs to learn useful representations of labels and exploit them in real-world problem solving.

1. Introduction

Knowing the function of a protein informs us on its biological role in the organism. With large numbers of genomes being sequenced every year, there is a rapidly growing number of newly discovered proteins. Protein function is most reliably determined in *wet lab* experiments, but current experimental methods are too slow for such quick income of novel proteins. Therefore, the development of tools for automated prediction of protein functions is necessary. Fast and accurate prediction of protein function is especially important in the context of human diseases since many of them are associated with specific protein functions.

The space of all known protein functions is defined by a directed acyclic graph known as the *Gene Ontology* (GO) (Ashburner et al., 2000), where each node represents one function and each edge encodes a hierarchical relationship between two functions, such as *is-a* or *part-of* (refer to Figure 2 for a visualisation). For every protein, its functions constitute a subgraph of GO, consistent in the sense that it is closed with respect to the predecessor relationship. GO contains thousands of nodes, with function subgraphs usually having dozens of nodes for each protein. Hence, the output of the protein function prediction problem is a *subgraph* of a hierarchically-structured graph.

This opens up a clear path of application for graph representation learning (Bronstein et al., 2017; Hamilton et al., 2017b; Battaglia et al., 2018), especially *graph neural networks* (GNNs) (Kipf & Welling, 2016; Veličković et al., 2017; Gilmer et al., 2017; Corso et al., 2020), given their natural inductive bias towards processing relational data.

One key aspect in which the protein function prediction task differs from most applications of graph representation learning, however, is in the fact that the graph is specified in the *label* space—that is, we are given a multilabel classification task in which we have known relational inductive biases over the individual labels (e.g. if protein X has function F, it must also have *all predecessor functions* of F under the closure constraint).

Driven by the requirement for a GNN to operate in the *label* space, we propose **Tail-GNN**, a graph neural network which learns representations of *labels*, introducing relational inductive biases into the *flat* label predictions of a feedforward neural network. Our results demonstrate that introducing this inductive bias provides significant gains on the protein function prediction task, paving the way to many other possible applications in the sciences (e.g., prediction of spatial phenomena over several correlated locations (Radosavljevic et al., 2010; Djuric et al., 2015), traffic state estimation (Djuric et al., 2011), and polypharmacy side effect prediction (Zitnik et al., 2018; Deac et al., 2019a)).

2. Tail-GNNs

In this section, we will describe an abstract model which takes advantage of a Tail-GNN, followed by an overview and intuition for the specific architectural choices we used for the protein prediction task. The entire setup from this section may be visualised in Figure 1.

Generally, we have a multi-label prediction task, from inputs

[1]Faculty of Mathematics, University of Belgrade [2]DeepMind. Correspondence to: Stefan Spalević <spalemon94@gmail.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020.

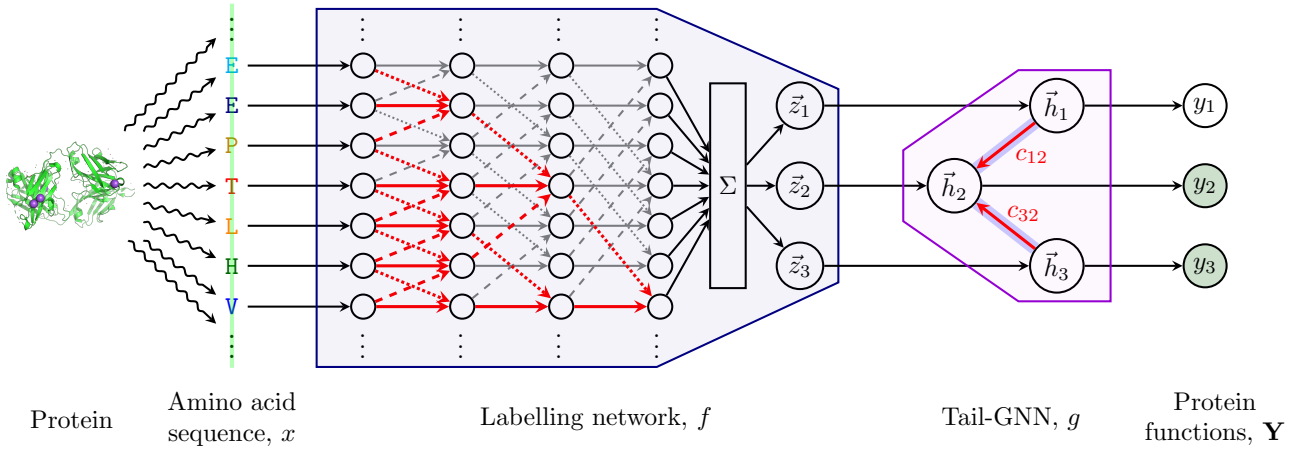

Figure 1. A high-level overview of the protein function modelling setup in this paper. Proteins are represented using their amino acid sequences (x), and are passed through the labelling network (f), to compute latent vectors for each label ($\vec{z}_i$). These latent vectors are passed to the Tail-GNN (g), which repeatedly aggregates their information along the edges of the gene ontology graph, computing an updated latent representation of each label ($\vec{h}_i$). Finally, a linear layer predicts the probability of the protein having the corresponding functions (y_i). The labelling network relies on *dilated convolutions* followed by global average pooling and reshaping. Note how dilated convolutions allow for an exponentially increasing receptive field at each amino acid.

$x \in \mathcal{X}$, to outputs $y_i \in \mathcal{Y}_i$, for each label $i \in \mathbb{L}$. We are also aware that there exist relations between labels, which we explicitly encode using a binary adjacency matrix $\mathbf{A} \in \mathbb{R}^{|\mathbb{L}|\times|\mathbb{L}|}$, such that $\mathbf{A}_{ij} = 1$ implies that the prediction for label j can be related[1] with the prediction for label i.

Our setup consists of a **labeller network**

$$f : \mathcal{X} \to (\mathcal{Z}_1 \times \mathcal{Z}_2 \times \cdots \times \mathcal{Z}_{|\mathbb{L}|}) \tag{1}$$

which attaches *latent vectors* $f(x) = \mathbf{Z} = \{\vec{z}_1, \ldots, \vec{z}_{|\mathbb{L}|}\}$, to each label i, for a given input x. Typically, these will be k-dimensional real-valued vectors, i.e. $\mathcal{Z}_i = \mathbb{R}^k$.

These labels are then provided to the **Tail-GNN** layer g, which is a node-level predictor; treating each label i as a node in a graph, $\vec{z}_i$ as its corresponding node features, and $\mathbf{A}$ as its corresponding adjacency matrix, it produces a prediction for each node:

$$g : \mathbb{R}^{|\mathbb{L}|\times k} \times \mathbb{R}^{|\mathbb{L}|\times|\mathbb{L}|} \to (\mathcal{Y}_1 \times \mathcal{Y}_2 \times \cdots \times \mathcal{Y}_{|\mathbb{L}|}) \tag{2}$$

That is, $g(f(x), \mathbf{A}) = g(\mathbf{Z}, \mathbf{A}) = \mathbf{Y} = (y_1, \ldots, y_{|\mathbb{L}|})$, provides the final predictions for the model in each label. As implied, the **Tail-GNN** is typically implemented within the graph neural network (Scarselli et al., 2008) framework, explicitly including the relational information.

Assuming f and g are differentiable w.r.t. their parameters, the entire system can be end-to-end optimised via gradient descent on the label errors w.r.t. ground-truth values.

[1]Note that different kinds of entries in $\mathbf{A}$ are also allowed, in case we would like to explicitly account for edge features.

In our specific case, the inputs x are protein sequences of one-hot encoded amino acids, and outputs y_i are binary labels indicating presence or absence of individual functions for those proteins.

Echoing the protein modelling results of Fast-Parapred (Deac et al., 2019b), we have used a deep *dilated* convolutional neural network for f (similarly as in ByteNet (Kalchbrenner et al., 2016) and WaveNet (Oord et al., 2016)). This architecture provides a parallelisable way of modelling amino-acid sequences without sacrificing performance compared to RNN encoders. This labelling network is *fully convolutional* (Springenberg et al., 2014): it predicts $|\mathbb{L}| \times k$ latent features for each amino acid, followed by global average pooling and reshaping the output to obtain a length-k vector for each label.

As we know that the gene ontology edges encode explicit containment relations between function labels, our Tail-GNN g is closely related to the GCN model (Kipf & Welling, 2016). At each step, we update latent features $\vec{h}_i$ in each label by aggregating neighbourhood features across edges:

$$\vec{h}'_i = \mathrm{ReLU}\left(\sum_{j\in\mathcal{N}_i} c_{ji}\mathbf{W}\vec{h}_j\right) \tag{3}$$

where $\mathcal{N}_i$ is the one-hop neighbourhood of label i in the GO, $\mathbf{W}$ is a shared weight matrix parametrising a linear transformation in each node, and c_{ji} is a coefficient of interaction from node j to node i, for which we attempt several variants: *sum-pooling* (Xu et al., 2018) ($c_{ji} = 1$), *mean-pooling* (Hamilton et al., 2017a) ($c_{ji} = \frac{1}{|\mathcal{N}_i|}$), and *graph attention*

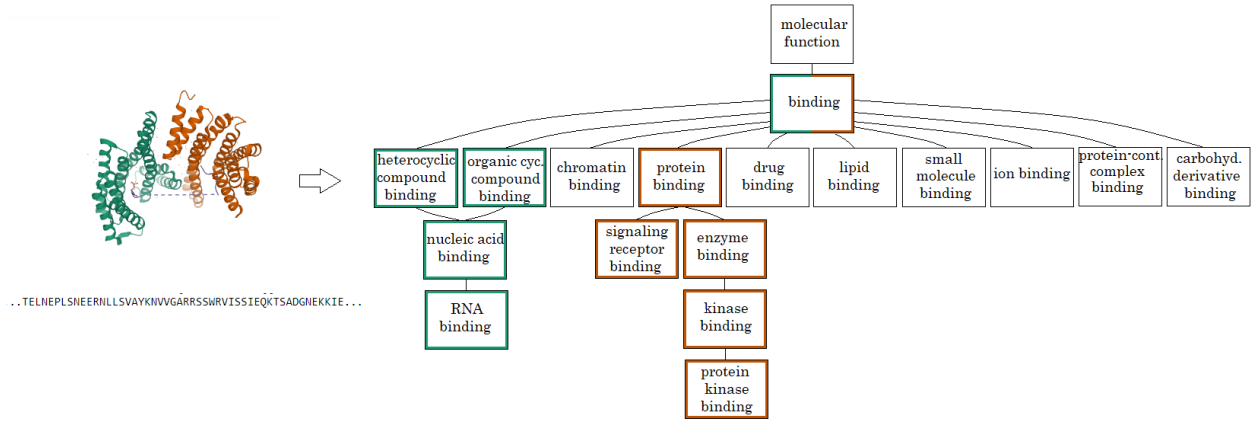

Figure 2. Representation of a function subgraph on a small subset of the ontology we leveraged. Assume that the input protein has three functions: *RNA binding*, *signaling receptor binding* and *protein kinase binding*. Its function subgraph contains all predecessors of these functions (e.g. *nucleic acid binding*, *enzyme binding*, *binding*). Note that, as we go deeper in the ontology, the functions associated with the nodes become more specialized.

($c_{ji} = a(\vec{h}_i, \vec{h}_j)$, where a is an *attention function* producing scalar coefficients). We use the same attention mechanism as used in GAT (Veličković et al., 2017).

Lastly, we also attempt to explicitly align with the containment inductive bias by leveraging *max-pooling*:

$$\vec{h}'_i = \mathrm{ReLU}\left(\max_{j \in \mathcal{N}_i} \mathbf{W}\vec{h}_j\right) \tag{4}$$

where $\max$ is performed elementwise.

The final layer of our network is a shared linear layer, followed by a logistic sigmoid activation. It takes the latent label representations produced by Tail-GNN and predicts a scalar value for each label, indicating the probability of the protein having the corresponding function. We optimise the entire network end-to-end using binary cross-entropy on the ground-truth functions.

It is interesting to note that, performing constrained relational computations in the label space, the operation of the Tail-GNN can be closely related to *conditional random fields* (CRFs) (Lafferty et al., 2001; Krähenbühl & Koltun, 2011; Cuong et al., 2014; Belanger & McCallum, 2016; Arnab et al., 2018). CRFs have been combined with GNNs in prior work (Ma et al., 2018; Gao et al., 2019), primarily as a means of strengthening the GNN prediction; in our work, we express all computations using GNNs alone, relying on the fact that, if optimal, Tail-GNNs could learn to specialise to the computations of the CRF through *neural execution* (Veličković et al., 2019), but will in principle have an opportunity to learn more *data-driven* rules for message passing between different labels.

Further, Tail-GNNs share some similarities with *gated propagation networks* (GPNs) (Liu et al., 2019), which leverage class relations to compute *class prototypes* for meta-learning (Snell et al., 2017). While both GPNs and Tail-GNNs perform GNN computations over a graph in the label space, the aim of GPNs is to compute structure-informed prototypes for a 1-NN classifier, while here we focus on multi-task predictions and directly produce outputs in an end-to-end differentiable fashion.

Beyond operating in the label space, GNNs have seen prior applications to protein function modelling through explicitly taking into account either the protein's residue contact map (Gligorijevic et al., 2019) or existing protein-protein interaction (PPI) networks. Especially, Hamilton et al. (2017a) provide the first study of explicitly running GNNs over PPI graphs in order to predict gene ontology signatures (Zitnik & Leskovec, 2017). However, as these models rely on an existence of either a reliable contact map or PPI graph, they cannot be reliably used to predict functions for novel proteins (for which these may not yet be known). Such information, if assumed available, may be explicitly included as a relational component within the labeller network.

3. Experimental Evaluation

3.1. Dataset

We used training sequences and functional annotations from CAFA3, a protein function prediction challenge (Zhou et al., 2019). The functional annotations were represented by functional terms of the hierarchical structure of the Gene Ontology (GO) (Ashburner et al., 2000)—the version released in April 2020. Out of the three large groups of functions represented in GO, we used the *Molecular Function Ontology* (MFO) which contains 11,113 terms. Function subgraphs for each protein were obtained by propagating functional

annotations to the root. We discarded obsolete nodes and functions occurring in less than 500 proteins in the original dataset, obtaining a reduced ontology with 123 nodes and 145 edges. Next, we eliminated proteins whose function subgraph contained only the root node (which is always active), as well as proteins longer than 1,000 amino acids.

All of the above constraints were devised with the aim of keeping the downstream task relevant, while at the same time simpler for the dilated convolutions to model—delegating most of the subsequent representational effort to the Tail-GNN. The final dataset contains 31,243 proteins, with an average sequence length of 431 amino acids. Average number of protein functions per protein is 7.

3.2. Training specifics

The dataset was randomly split into training/validation/test sets, with a rough proportion of 68:17:15 percent. We counted up the individual label occurrences within these datasets, observing that the split was appropriately stratified across all of them. The time of characterization of protein function was not taken into account since the aim was to examine whether GNN method is able to cope with structural labels.

The architectural hyperparameters were determined based on the validation set performance, using the F_1 score—a suitable measure for imbalanced label problems, which is also commonly used for evaluating models in CAFA challenges (Zhou et al., 2019). Via thorough hyperparameter sweeps, we decided on a labelling network of six dilated convolutional layers, with exponentially increasing dilation rate. Initially the individual amino acids are embedded into 16 features, and the individual layers compute $\{32, 64, 128, 256, 512, 512\}$ features each, mirroring the results of Deac et al. (2019b).

For predicting functions directly from the labelling network, we follow with a linear layer of 123 features and global average pooling across amino acid positions, predicting the probability of each function occurring.

When pairing with Tail-GNN, however, the linear layer computes $123k$ features, with k being the number of latent features computed per label (i.e. the dimensionality of the $\vec{z}_i$ vectors). We swept various small[2] values of k, finding $k = 9$ to perform optimally.

In addition, we concatenate five *spectral* features to each input node to the Tail-GNN, in the form of the five eigenvectors corresponding to the five largest eigenvalues of the graph Laplacian—inspired by the Graph Fourier Transform of Bruna et al. (2013).

[2]Further increasing k quickly leads to an increase in parameter count, leading to overfitting and memory issues.

Table 1. Values of F_1 score on our validation and test datasets for all considered architectures, aggregated over five random seeds.

Model	**Validation** F_1	**Test** F_1
Labelling network	0.582 ± 0.003	0.584 ± 0.003
Tail-GNN-mean	0.583 ± 0.006	0.586 ± 0.004
Tail-GNN-GAT	0.582 ± 0.004	0.587 ± 0.005
Tail-GNN-max	0.581 ± 0.002	0.585 ± 0.004
Tail-GNN-sum	$\mathbf{0.596} \pm 0.003$	$\mathbf{0.600} \pm 0.003$
Tail-GNN-sum (no spectral fts.)	0.587 ± 0.007	0.590 ± 0.008

For each choice of Tail-GNN aggregation, we evaluated one and two GNN layers of 16 features each, followed by a linear classifier for protein functions. We also assessed performance without incorporating the spectral features.

All models are optimising the binary cross-entropy on the function predictions using the Adam SGD optimiser (Kingma & Ba, 2014) (with learning rate 0.001 and batch size of 32), incorporating class weights to account for any imbalance. We train for 200 epochs with early stopping on the validation F_1, with a patience of 20 epochs.

3.3. Results

We evaluate the recovered optimised models across five random seeds. Results are given in Table 1; the *labelling network* is the baseline dilated convolutional network without leveraging GNNs. Additionally, we provide results across a variety of Tail-GNN configurations. Our results are consistent with the top-10 performance metrics in the CAFA3 challenge (Zhou et al., 2019) but the direct comparison was not possible since we use a reduced ontology.

Our results demonstrate a significant performance gain associated with appending Tail-GNN to the labelling network, specifically, when using the *sum* aggregator. While less aligned to the containment relation than maximisation, summation is also more "forgiving" with respect to any labelling mistakes: if Tail-GNN-max had learnt to perfectly implement containment, any mistakenly labelled leaves would cause large chunks of the ontology to be misclassified.

Further, we discover a performance gain associated with including the Laplacian eigenvectors: including them as node features, and a low-frequency indicator of global graph features, further improves the results of the Tail-GNN-sum.

While much of our analysis was centered around the protein function prediction task, we conclude by noting that the way Tail-GNNs are defined is task-agnostic, and could easily see application in other areas of the sciences (as discussed in the Introduction), with minimal modification to the setup.

References

Arnab, A., Zheng, S., Jayasumana, S., Romera-Paredes, B., Larsson, M., Kirillov, A., Savchynskyy, B., Rother, C., Kahl, F., and Torr, P. H. S. Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction. *IEEE Signal Processing Magazine*, 35(1):37–52, 2018.

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

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Belanger, D. and McCallum, A. Structured prediction energy networks. In *ICML*, 2016.

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4): 18–42, 2017.

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

Corso, G., Cavalleri, L., Beaini, D., Liò, P., and Veličković, P. Principal neighbourhood aggregation for graph nets. *arXiv preprint arXiv:2004.05718*, 2020.

Cuong, N. V., Ye, N., Lee, W. S., and Chieu, H. L. Conditional random field with high-order dependencies for sequence labeling and segmentation. *Journal of Machine Learning Research*, 15:981–1009, 2014.

Deac, A., Huang, Y.-H., Veličković, P., Liò, P., and Tang, J. Drug-drug adverse effect prediction with graph co-attention. *arXiv preprint arXiv:1905.00534*, 2019a.

Deac, A., Veličković, P., and Sormanni, P. Attentive cross-modal paratope prediction. *Journal of Computational Biology*, 26(6):536–545, 2019b.

Djuric, N., Radosavljevic, V., Coric, V., and Vucetic, S. Travel speed forecasting by means of continuous conditional random fields. *Transportation Research Record: Journal of the Transportation Research Board*, 2263:131–139, 12 2011. doi: 10.3141/2263-15.

Djuric, N., Radosavljevic, V., Obradovic, Z., and Vucetic, S. Gaussian conditional random fields for aggregation of operational aerosol retrievals. *IEEE Geoscience and Remote Sensing Letters*, 12(4):761–765, 2015.

Gao, H., Pei, J., and Huang, H. Conditional random field enhanced graph convolutional neural networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 276–284, 2019.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1263–1272. JMLR. org, 2017.

Gligorijevic, V., Renfrew, P. D., Kosciolek, T., Leman, J. K., Cho, K., Vatanen, T., Berenberg, D., Taylor, B. C., Fisk, I. M., Xavier, R. J., et al. Structure-based function prediction using graph convolutional networks. *bioRxiv*, pp. 786236, 2019.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pp. 1024–1034, 2017a.

Hamilton, W. L., Ying, R., and Leskovec, J. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017b.

Kalchbrenner, N., Espeholt, L., Simonyan, K., Oord, A. v. d., Graves, A., and Kavukcuoglu, K. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Krähenbühl, P. and Koltun, V. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011.

Lafferty, J., Mccallum, A., and Pereira, F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. pp. 282–289, 01 2001.

Liu, L., Zhou, T., Long, G., Jiang, J., and Zhang, C. Learning to propagate for graph meta-learning. In *Advances in Neural Information Processing Systems*, pp. 1037–1048, 2019.

Ma, T., Xiao, C., Shang, J., and Sun, J. Cgnf: Conditional graph neural fields. 2018.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Radosavljevic, V., Vucetic, S., and Obradovic, Z. Continuous conditional random fields for regression in remote sensing. volume 215, pp. 809–814, 01 2010. doi: 10.3233/978-1-60750-606-5-809.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

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

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Veličković, P., Ying, R., Padovano, M., Hadsell, R., and Blundell, C. Neural execution of graph algorithms. *arXiv preprint arXiv:1910.10593*, 2019.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Zhou, N., Jiang, Y., Bergquist, T. R., Lee, A. J., Kacsoh, B. Z., Crocker, A. W., Lewis, K. A., Georghiou, G., Nguyen, H. N., Hamid, M. N., et al. The cafa challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome biology*, 20(1):1–23, 2019.

Zitnik, M. and Leskovec, J. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33 (14):i190–i198, 2017.

Zitnik, M., Agrawal, M., and Leskovec, J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, 2018.