Path-Augmented Graph Transformer Network

Benson Chen [1] **Regina Barzilay** [1] **Tommi Jaakkola** [1]

Abstract

Much of the recent work on learning molecular representations has been based on Graph Convolution Networks (GCN). These models rely on local aggregation operations and can therefore miss higher-order graph properties. To remedy this, we propose Path-Augmented Graph Transformer Networks (PAGTN) that are explicitly built on longer-range dependencies in graph-structured data. Specifically, we use path features in molecular graphs to create global attention layers. We compare our PAGTN model against the GCN model and show that our model consistently outperforms GCNs on molecular property prediction datasets including quantum chemistry (QM7, QM8, QM9), physical chemistry (ESOL, Lipophilictiy) and biochemistry (BACE, BBBP)[2].

1. Introduction

Graph Convolution Networks (GCN) have successfully been applied to molecular graph datasets (Duvenaud et al., 2015; Kearnes et al., 2016; Niepert et al., 2016; Jin et al., 2017). These "message-passing" algorithms exploit the feature locality of graphs through the usage of convolution operations (Gilmer et al., 2017). However, the convolution operator aggregates only local information, so long-range dependencies are naturally difficult for these models to learn. In molecular graphs, many informative structures are characterized by the paths between nodes. We propose the Path-Augmented Graph Transformer Network (PAGTN) model that utilizes these path features in global attention layers, resulting in a richer, more expressive model. Specifically, our model learns a better representation of the graph in the following ways:

Long-range dependencies In GCNs, long-range dependencies take many convolution layers to learn, because feature aggregation happen only within the immediate neighborhoods of each node. For large enough graphs, GCNs may fail to capture these long-range dependencies entirely. Our PAGTN model can more easily capture these dependencies because every node attends to all other nodes in the graph.

Substructures In graph problems, it is imperative for a model to pick up the important substructures in the graph. GCN models necessitate several layers to propagate information and learn these substructures. The advantage of our model is that this interaction can be learned within a single layer.

We test our PAGTN model against the GCN model on 7 benchmark moelcular property prediction tasks ranging from quantum chemistry (QM7, QM8, QM9), physical chemistry (ESOL, Lipophilictiy) and biochemistry (BACE, BBBP) (Wu et al., 2018). Each dataset focuses on a different property of the molecule, making composition of these datasets highly variable. Nevertheless, our model consistently shows improved performance against the GCN baseline, demonstrating that our model can learn more powerful representations.

2. Related Works

Transformer architectures have triumphed over traditional recurrent and convolution models in many natural language tasks such as machine translation (Vaswani et al., 2017). While recurrent and convolution models often incorporate a single attention layer at the top (Luong et al., 2015), it has been shown that using only these globally-connected self-attention layers learns a much more powerful model.

Attention models on graphs have been explored in previous works. Primarily, the Graph Attention Network (Veličković et al., 2017) and its variants (Gong & Cheng, 2018; Zhang et al., 2018; Monti et al., 2018) aggregates information within local neighborhoods by using attention. We emphasize that our model focuses on the global connectivity of the nodes. Moreover, our model does not use any complex attention mechanism across layers, but rather provides a simple framework using the path features that works well empirically. Another proposed model, Graph Transformer (Li et al., 2019), uses global attention layers, but that model

[1]Department of EECS, Massachusetts Institute of Technology, Cambridge, USA. Correspondence to: Benson Chen <bensonc@csail.mit.edu>.

Presented at the ICML 2019 Workshop on Learning and Reasoning with Graph-Structured Data Copyright 2019 by the author(s).

[2]Code to replicate our experiments is provided at https://github.com/benatorc/PA-Graph-Transformer

does not extend to graphs in which edge and path features are important.

3. Model

In this section, we first briefly overview the Transformer model. Then, we will go over our contributions, describing our variant of the Transformer model that uses path features to learn expressive representations of graphs.

3.1. Transformer

The Transformer model (Vaswani et al., 2017), in contrast to traditional recurrent or convolution architectures, consists of fully-connected attention layers. These models use multi-head self-attention, which confers more flexibility for the attention module. The attention layers are connected by position-wise feed-forward layers, with residual links and layer normalization present at each layer.

The transformer model itself has no direct notion of relative position, so it uses positional encodings in the form of sinusoidal functions. However, this form of positional encoding is not possible in graphs, because there is no longer a natural sequential ordering of the nodes. We introduce path features, which represent how two nodes are connected. These path features influence the attention module in the network, so that the node embeddings are globally aware. We first explain how we construct these path features, then how they are incorporated into the attention framework.

3.2. Path Features

We compute the path features between each node pair by taking the shortest path between them. Due to cycles on graphs, these shortest paths may not be unique. For molecular graphs, these cycles arise due to ring substructures on the graphs. Because the edge features are consistent within a single ring or cycle, multiple paths are almost always equivalent feature-wise; therefore, this approach is sensible for our model.

For efficiency, we truncate the path features between nodes up to a distance d apart. We make the assumption that as the distance between two nodes increases, the connectivity between the two nodes matter less. Therefore, this constraint puts a natural regularizer on the model. So while each node attend to all other nodes in the graph, that node only has rich edge features for a local neighborhood.

The path features between two nodes $i \rightarrow j$ is a concantenation of the following three components:

Edge features: are constructed by concatenating the individual bond features of the shortest path between $i \rightarrow j$. Let b_k be the bond features of the kth bond along the path, which includes the bond type, conjugacy and ring membership (whether or not that bond is in a ring) features. Then, the edge features are just the concatenation of the features: $[b_1; b_2; ...; b_n]$. Note that if $n > d$, we zero out these features, and if $n < d$, we pad the feature vector with zeros.

Distance: is a one-hot feature of the distance between two nodes $i \rightarrow j$, truncated by d.

Ring Membership: is a one-hot feature denoting whether the node i and node j are in the same ring. For molecular graphs, we find that it's also helpful to include one-hot features for specific rings such as five/six-membered aromatic rings. Note that this is distinct from the bond ring membership features which indicates whether a particular bond is part of a ring.

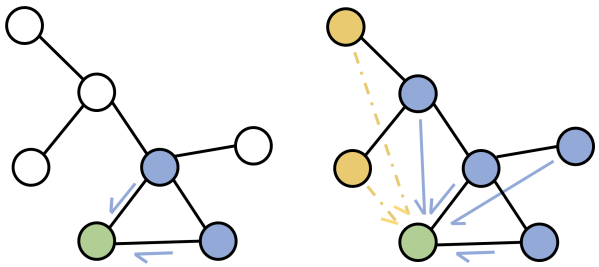

Figure 1. Illustration of graph propagation properties for GCN (left) and our PAGTN model (right). For the GCN, the source attention node (green) only attends to its immediate neighbors (blue). In the PAGTN, the source attention node (green) has connectivity information in the form of path features for its local neighborhood, $d = 2$, (blue), but also attends to all other nodes (yellow).

A comparison of the information propagation properties of the network layers is illustrated in Figure 1. In regular GCNs, only the direct neighborhood is impacted–which can require many layers of computation to learn from the graph. In our PAGTN model, every node is globally connected, which makes learning complex dependencies easier.

3.3. Additive Self-Attention

Although transformer models normally use scaled dot-product attention, we found in our experiments that an additive form of attention was easier to train and resulted in better performance. One way we deviate from standard self-attention modules is that we exclude the source node when computing attention for that node. The residual links at each layer grounds the learned embedding at each layer to be representative of the original input node.

Define $\mathbf{x} = (x_1, x_2, ..., x_n) \in \mathbb{R}^{n \times F_n}$ as a matrix of the input node features, where n is the number of nodes and F_n is the number of node features. Similarly, let $\mathbf{p} = (p_{1,1}, p_{1,2} ... p_{n,n}) \in \mathbb{R}^{n \times n \times F_p}$ be a matrix of the in-

put pairwise path features where F_p is the number of path features.

At each layer, we update the node features by computing a weighted average using learned attention weights. Let $\mathbf{h}^l = (h_1^l, h_2^l, ..., h_n^l) \in \mathbb{R}^{n \times F_m}$ represent the node features at layer l, where F_m is the number of model features. Note that the elements of $\mathbf{h}^0$ are the linearly transformed input features ($\mathbf{h^0} = W\mathbf{x^T}$). We compute $s_{i,j}^l$, the attention score of node $i \to j$, as:

$$s_{i,j}^l = W^{S_2}\Big[\text{LeakyReLU}\Big(W^{S_1}[h_i^{l-1}; h_j^{l-1}; p_{i,j}]\Big)\Big] \quad (1)$$

The attention probabilities $a_{i,j}$ are calculated as a softmax over the attention scores. As mentioned earlier, we exclude the source node itself when computing the attention probabilities.

$$\alpha_{i,j}^l = \text{softmax}(s_{i,j}^l) = \frac{\exp(s_{i,j}^l)}{\sum_{j' \neq i} \exp(s_{i,j'}^l)} \quad (2)$$

Using attention probabilities, we can compute a weighted average over the node features. Since we note the importance of path features in graphs, we define the output features to be a function of both node and path features. Here, σ is some non-linear function (we use ReLU for our experiments).

$$h_i^l = \sigma\big(W^{H_2} h_i^{l-1} + \sum_{j \neq i} \alpha_{i,j}^l W^{H_1}[h_j^{l-1}; p_{i,j}]\big) \quad (3)$$

As introduced in (Vaswani et al., 2017), multi-head attention can often benefit the model by allowing it more easily to attend to different aspects of the input data. If we split the attention into K heads, we can define the update rule for h_i^l as a function of the embeddings associated with invididual heads $h_i^{l,k}$:

$$h_i^l = \Big\|_k \sigma\big(W^{H_2,k} h_i^{l-1,k} + \sum_{j \neq i} \alpha_{i,j}^{l,k} W^{H_1,k}[h_j^{l-1,k}; p_{i,j}]\big) \quad (4)$$

Here, $\|$ is the concatenation operator. Empirically, we find that using multi-head attention helps on some tasks, but not on all tasks.

3.4. Molecule Embedding

Since we are interested in property prediction tasks for the molecule as a whole, we compute a molecule embedding h_M by aggregating the individual node embeddings. Here, we add a residual link to the input features, $\mathbf{x}$, of the network.

$$h_M = \sum_i \sigma\Big(W^M[h_i^L; x_i]\Big) \quad (5)$$

We choose the sum operator to aggregate the feature embeddings, which has higher expressive power than other classic operators (Xu et al., 2018). The target property is predicted using a 1-layer MLP with h_M as input.

4. Experiments

4.1. Experimental Setup

We test our model on 7 benchmark property prediction tasks, including quantum mechanics (QM7, QM8, QM9), physical chemistry (ESOL, Lipophilicity) and biochemistry (BACE, BBBP) (Wu et al., 2018).

We split each dataset into 10 different folds of 80:10:10 (train:validation:test) splits, and record the average performance over the folds using the appropriate measure for each dataset. Since these datasets feature markedly different properties, we tune the hyperparameters of the model for individual datasets.

4.2. Baselines

We compare our transformer to several baselines.

MolNet Molecule Net (Wu et al., 2018) tested many graph-based deep learning methods as well as more conventional methods on these property prediction datasets. We use their top performing model for each dataset.

GCN This is a traditional graph convolution model, and here we use a similar model to (Jin et al., 2017). We find that this model achieves very competitive results compared MolNet (which itself uses many different graph-based convolution models), and therefore is a fair baseline. GCN models can have a self-attention layer at the top, but we find empirically that this often hurts performance so we do not include this attention layer in our baseline.

PAGTN (Local) We include a variant of our PAGTN model, which does not attend to nodes for which there are no path features. That is, the model masks out nodes that are further than d from the source attention node. We include this baseline to show that global attention does indeed improve performance.

Our proposed model is dubbed the **PAGTN (Global)**, which attends globally to all nodes.

4.3. Property Prediction

The results of the property prediction tasks can be seen from Table 1. We first see that the GCN model is very comparable to those of MolNet (Wu et al., 2018). And compared to the

Table 1. Results comparing our PAGTN model to various baselines. The metrics used were MAE for the quantum mechanics datasets (QM7, QM8, QM9), RMSE for the physical chemistry datasets (ESOL, Lipophilicity), and AUC for the biochemistry datasets (BACE, BBBP). The bold numbers represent the model with the best performance.

DATA SET	# DATA	METRIC	MOLNET	GCN	PAGTN (LOCAL)	PAGTN (GLOBAL)
QM7	6,830	MAE	–[3]	52.4 ± 2.8	48.9 ± 3.4	**47.8 ± 3.0**
QM8	21,786	MAE	.0143	.0105 ± .0003	.0108 ± .0003	**.0102 ± .0003**
QM9	133,885	MAE	2.35	2.20 ± .03	2.10 ± .04	**2.07 ± .05**
ESOL	1,128	RMSE	.580	.587 ± .05	.592 ± .06	**.554 ± .06**
LIPOPHILICITY	4,200	RMSE	.655	.578 ± .05	.592 ± .05	**.572 ± .04**
BACE	1,513	AUC	.867	.878 ± .02	.876 ± .02	**.880 ± .01**
BBBP	2,039	AUC	.729	.907 ± .03	.898 ± .04	**.913 ± .03**

Table 2. Results comparing the GCN and the PAGTN (Global) models on a synthetic ring membership prediction task, which is to test whether or not two nodes are in the same ring on the graph. GCN does cannot always predict this property well, while the PAGTN can easily incorporate these features into the model.

MODEL	ACCURACY	AUC
GCN	91.6	96.5
PAGTN (GLOBAL)	97.8	99.8

GCN model, our PAGTN model achieves surperior performance in all 7 of these property prediction tasks, illustrating the broad representational power of the model. Furthermore, we see from the local PAGTN model that by attending globally rather than restricting to the local neighborhood, we always see an improvement in performance. This reveals that the global attention does indeed help the model.

4.4. Ring Membership

To help elucidate why the PAGTN formulation is better than that of GCN, we turn to a synthetic task. We note that certain properties such as ring membership can prove difficult for regular graph convolution networks. To test this observation, and to demonstrate the effectiveness of our PAGTN model, we create a synthetic dataset by choosing a subset of 5,769 molecules from the property prediction datasets that have at least 2 rings. For each molecule, we randomly choose 5 pairs of atoms that are in the same ring, and 5 pairs of atoms that are in different rings. For atoms in fused ring systems, we count two atoms in the same ring if they are in the smallest possible ring system.

From Table 2, we see that the GCN fails to perfectly predict ring membership. This is not surprising as the convolution operation has to learn to disambiguate features of nodes in same and different rings. These subtle but important graph features are imperative for models to fully capture the representation of the graph. Our PAGTN naturally solves this issue, since we can incorporate these features as a part of the network, whereas it is a lot more difficult to incorporate these features in the local convolution model. Note that the PAGTN still does not solve the problem perfectly, and this is due to the fact that in highly symmetrical graphs, multiple nodes are equivalent which leads to ambiguous ring membership as see from Figure 2.

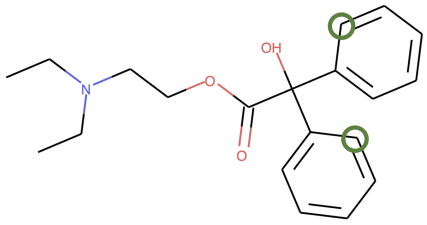

Figure 2. The two green-circled atoms are completely symmetric, so their output feature embeddings are equivalent. Since the ring membership prediction is made by aggregating pairwise node features, it is impossible to tell whether any other atom is in the same or different ring from these two atoms.

5. Conclusion

In this paper, we introduced the PAGTN model that exploits the connectivity structure of the data in its global attention mechanisms. Through the path features that we engineer into model's attention layers, our model better captures the complex structures of graphs compared to GCNs. On 7 different chemical property prediction tasks, we have shown that our PAGTN model can outperform traditional GCNs, and we hope that these global-attention models that incorporate path features will be used more frequently in works on molecular graphs moving forward.

[3]MolNet uses a stratified sampling of the data for QM7, whereas we use random sampling for this work.

References

Duvenaud, D. K., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. *CoRR*, abs/1509.09292, 2015.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. *CoRR*, abs/1704.01212, 2017.

Gong, L. and Cheng, Q. Adaptive edge features guided graph attention networks. *arXiv preprint arXiv:1809.02709*, 2018.

Jin, W., Coley, C. W., Barzilay, R., and Jaakkola, T. S. Predicting organic reaction outcomes with weisfeiler-lehman network. *CoRR*, abs/1709.04555, 2017.

Kearnes, S., McCloskey, K., Berndl, M., Pande, V., and Riley, P. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.

Li, Y., Liang, X., Hu, Z., Chen, Y., and Xing, E. P. Graph transformer, 2019. URL https://openreview.net/forum?id=HJei-2RcK7.

Luong, M.-T., Pham, H., and Manning, C. D. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

Monti, F., Shchur, O., Bojchevski, A., Litany, O., Günnemann, S., and Bronstein, M. M. Dual-primal graph convolutional networks. *CoRR*, abs/1806.00770, 2018. URL http://arxiv.org/abs/1806.00770.

Niepert, M., Ahmed, M., and Kutzkov, K. Learning convolutional neural networks for graphs. *CoRR*, abs/1605.05273, 2016.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

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

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? *CoRR*, abs/1810.00826, 2018.

Zhang, J., Shi, X., Xie, J., Ma, H., King, I., and Yeung, D. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. *CoRR*, abs/1803.07294, 2018.