Graph Attention Networks with Positional Embeddings

Liheng Ma[1,2], Reihaneh Rabbany[1,2], Adriana Romero-Soriano[1,3]

`liheng.ma@mail.mcgill.ca, rrabba@cs.mcgill.ca, and adrianars@fb.com`

[1] SCS McGill University, Montreal, Canada
[2] Mila, Montreal, Canada
[3] Facebook AI Research, Montreal, Canada

Abstract. Graph Neural Networks (GNNs) are deep learning methods which provide the current state of the art performance in node classification tasks. GNNs often assume homophily – neighboring nodes having similar features and labels–, and therefore may not be at their full potential when dealing with non-homophilic graphs. In this work, we focus on addressing this limitation and enable Graph Attention Networks (GAT), a commonly used variant of GNNs, to explore the structural information within each graph locality. Inspired by the positional encoding in the Transformers, we propose a framework, termed Graph Attentional Networks with Positional Embeddings (GAT-POS), to enhance GATs with positional embeddings which capture structural and positional information of the nodes in the graph. In this framework, the positional embeddings are learned by a model predictive of the graph context, plugged into an enhanced GAT architecture, which is able to leverage both the positional and content information of each node. The model is trained jointly to optimize for the task of node classification as well as the task of predicting graph context. Experimental results show that GAT-POS reaches remarkable improvement compared to strong GNN baselines and recent structural embedding enhanced GNNs on non-homophilic graphs.

Keywords: Graph Neural Networks · Attention · Positional Embedding.

1 Introduction

The use of graph-structure data is ubiquitous in a wide range of applications, from social networks, to biological networks, telecommunication networks, 3D vision or physics simulations. The recent years have experienced a surge in graph representation learning methods, with Graph Neural Networks (GNNs) currently being at the forefront of many application domains. Recent advances in this direction are often categorized as spectral approaches and spatial approaches.

Spectral approaches define the convolution operator in the spectral domain and therefore capture the structural information of the graph [2]. However, such approaches require computationally intense operations and yield filters which may not be localized in the spatial domain. A number of works have been proposed to localize spectral filters and approximate them for computation efficiency [6, 10, 11]. However, spectral filters based on the eigenbasis of the graph Laplacian depend on the graph structure, and thus cannot be directly applied to new graph structures, which limits their performance in the inductive setting. Spatial approaches directly define a spatially localized operator on the

localities of the graph (i.e., neighborhoods), and as such are better suited to generalize to new graphs. Most spatial approaches aggregate node features within the localities, and then mix the aggregated features channel-wise through a linear transformation [3, 8, 29, 14, 20]. The spatial aggregation operator has been implemented as mean-pooling or max-pooling in [8], and sum-pooling in [29, 14, 20]. In order to increase the expressive power of spatial approaches, previous works have defined the aggregation operators with adaptive kernels in which the coefficients are parameterized as a function of node features. For instance, MoNets [18] define the adaptive filters based on a Gaussian mixture model formulation, and graph attention networks (GATs) [25] introduce a content-based attention mechanisms to parameterize the coefficients of the filter.

Among them, GATs [25] have become widely used and shown great performance in node classification [15, 23, 7]. However, the GATs' adaptive filter computation is based on node content exclusively and attention mechanisms cannot inherently capture the structural dependencies among entities at their input [24, 12], considering them as a set structure. Therefore, GAT's filters cannot fully explore the structural information of the graph either. This may put the GAT framework at a disadvantage when learning on non-homophilic graph datasets, in which edges merely indicate the interaction between two nodes instead of their similarity [19]. Compared to homophilic graphs in which edges indicate similarity between the connected nodes, non-homophilic graphs are more challenging and higher-level structural patterns might be required to learn the node labels. In sequence-based [24, 21, 5], tree-based [26] and image-based [4, 32] tasks, the lack of structural information leveraged by the attention mechanisms has been remedied by introducing handcrafted or learned positional encodings [24, 21, 26, 5, 4, 32], resulting in improved performances.

Inspired by these positional encodings, and to improve learning on non-homophilic graphs, we aim to enhance GATs with structural information. Therefore, we propose a framework, called *Graph Attention Networks with Positional Embeddings* (GAT-POS), which leverages both positional information and node content information in the attention computation. More precisely, we modify the graph attention layers to incorporate a positional embedding for each node, produced by an positional embedding model predictive of the graph context. Our GAT-POS model is trained end-to-end to optimize for a node classification task while simultaneously learning the node positional embeddings. Our results on non-homophilic datasets highlight the potential of the proposed approach, notably outperforming GNN baselines as well as the recently introduced Geom-GCN [19], a method tailored to perform well on non-homophilic datasets. Moreover, as a sanity check, we validate the proposed framework on standard homophilic datasets and demonstrate that GAT-POS can reach a comparable performance to GNN baselines. To summarize, the contributions of this paper are:

  – We propose GAT-POS, a novel framework which enhances GAT with positional embeddings for learning on non-homophilic graphs. The framework enhances the graph attentional layers to leverage both node content and positional information.
  – We develop a joint training scheme for the proposed GAT-POS to support end-to-end training and learn positional embeddings tuned for the supervised task.
  – We show experimentally that GAT-POS significantly outperforms other baselines on non-homophilic datasets.

2   Method

In this paper, we consider the semi-supervised node classification task, and follow the problem setting of GCNs [10] and GATs [25]. Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected and unweighted graph, with a set of N nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$. Each node $v \in \mathcal{V}$ is represented with a D-dimensional feature vector $\mathbf{x}_v \in \mathbb{R}^D$. Similarly, each node has an associated label represented with a one-hot vector $\mathbf{y}_v \in \{0, 1\}^C$, where C is the number of classes. The edges in $\mathcal{E}$ are represented as an adjacency matrix $\mathbf{A} = \{0, 1\}^{N \times N}$, in which $\mathbf{A}_{vu} = 1$ iff $(v, u) \in \mathcal{E}$. The neighborhood of a node is defined as $\mathcal{N}(v) = \{u | (v, u) \in \mathcal{E}\}$.

In the rest of the section, we will present our proposed GAT-POS framework, and detail the implementation of each one of its components.

2.1   The GAT-POS Model

Multiple previous works have been proposed to incorporate positional embeddings in the attention mechanisms for sequences and grids [24, 5, 21, 26, 32, 4]. However, the structure in a graph is more complicated and the positional encoding for these earlier works cannot be directly generalized to the graphs. Therefore, we propose our framework, Graph Attention Networks with Positional Embeddings (GAT-POS) to incorporate the positional embeddings in the attention mechanisms of GATs. In particular, we propose to learn the positional embeddings via an embedding model with an unsupervised objective $\mathcal{L}_u$, termed *positional embedding model*, which allows the positional embeddings to capture richer positional and structural information of the graph. We provide a detailed comparison with the previous works in Section 3.

In order to support end-to-end training, we propose a main-auxiliary architecture where the positional embedding model is plugged into GATs, inspired by the works of [27, 30]. With this architecture, the supervised task of the GATs enhanced with positional embeddings, and the unsupervised task of the positional embedding model are trained jointly. Consequently, besides supporting end-to-end training, the positional embedding model can learn embeddings not only predictive of the graph context, but also beneficial to the supervised task.

Fig. 1 provides an overview of the GAT-POS architecture, where each rectangle denotes the operation of a layer; the black arrows denote forward propagation and the green arrows denote backpropagation.

For the node classification tasks, the supervised loss function is the cross-entropy error over the set of labeled examples observed during training $\mathcal{Y}_L$, as follows,

$$\mathcal{L}_\mathcal{S}(\{\hat{\mathbf{y}}_v\}_{v \in \mathcal{Y}_L}, \{\mathbf{y}_v\}_{v \in \mathcal{Y}_L}) = - \sum_{v \in \mathcal{Y}_L} \mathbf{y}_v^\intercal \log \hat{\mathbf{y}}_v \tag{1}$$

where $\hat{\mathbf{y}}_v$ is node v's predicted label. The unsupervised task is designed to guide the positional embedding model to capture information beneficial to the supervised task.

In the following subsection, we will introduce a particular implementation of GAT-POS. However, it is worth noting that our framework is agnostic to the particular setup of the enhanced graph attentional layer, the architecture of the positional embedding

model, the choice of the unsupervised objective as well as the way the main and auxiliary architectures are connected.
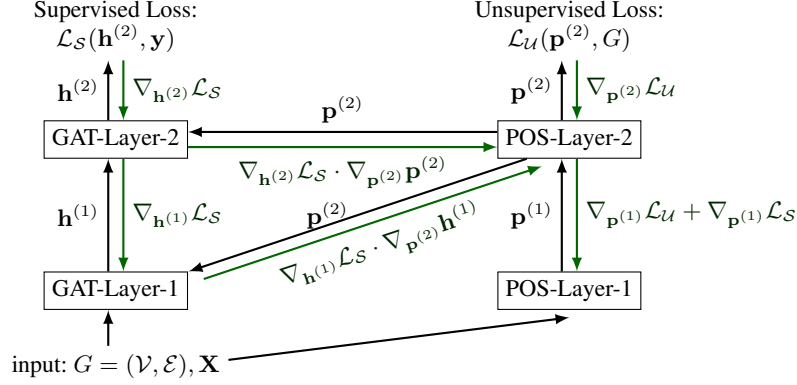

Fig. 1. A Demonstration of GAT-POS Architecture (the subscriptions are dropped for simplicity)

2.2 Positional Embedding Enhanced Graph Attentional Layer

We extend the graph attentional layer of [25] to leverage the node embeddings extracted from the GAT-POS positional embedding model when computing the attention coefficients. We modify the graph attentional layer to consider positional embeddings in the attention scores computation. In particular, our positional embedding enhanced graph attentional layer transforms a vector of node features $\mathbf{h}_v \in \mathbb{R}^F$ into a new vector of node features $\mathbf{h}'_v \in \mathbb{R}^{F'}$, where F and F' are the number of input and output features in each node, respectively. We start by computing the attention coefficients in the neighborhood of node v as follows,

$$\alpha_{vu}^k = \operatorname*{softmax}_{u \in \mathcal{N}_v \cup v} \left(\text{leakyrelu}(\mathbf{a}_k^\mathsf{T}[\mathbf{W}_k\mathbf{h}_v + \mathbf{U}_k\mathbf{p}_v \| \mathbf{W}_k\mathbf{h}_u + \mathbf{U}_k\mathbf{p}_u]) \right) \qquad (2)$$

where $\mathbf{W}_k$, $\mathbf{U}_k$ and $\mathbf{a}_k$ are the weights in the k-th attention head; $\mathbf{p}_v$ is the positional embedding for node v; $\|$ denotes the concatenation; and

$$\operatorname*{softmax}_{u \in \mathcal{U}}(e_{vu}) = \frac{\exp(e_{vu})}{\sum_{u' \in \mathcal{U}} \exp(e_{vu'})}$$

The attention coefficients, computed based on the input node features and structural information, are expected to exploit the structural and semantic information within the neighborhood. We subsequently update the features of node v by linear transforming the features of the nodes in the neighborhood with the obtained attention coefficients,

$$\mathbf{h}'_v = \begin{cases} \big\|_{k=1}^K \sigma\big(\sum_{u \in \mathcal{N}_v \cup v} \alpha_{vu}^k \cdot \mathbf{W}_k\mathbf{h}_u\big) & \text{, if at the hidden layers,} \\ \sigma\left(\frac{1}{K}\sum_{k=1}^K \sum_{u \in \mathcal{N}_v \cup v} \alpha_{vu}^k \cdot \mathbf{W}_k\mathbf{h}_u\right) & \text{, if at the output layer.} \end{cases} \qquad (3)$$

where σ denotes nonlinear activation function. Consequently, the features extracted by each of the positional embedding enhanced attentional layers should be able to explore the structural and semantic information in each neighborhood.

Finally, the enhanced GATs in GAT-POS is constructed by stacking multiple such graph attentional layers. In the first layer, $\mathbf{h}_v$ is set as $\mathbf{x}_v$ and we denote the output of the final layer as the predicted node label $\hat{\mathbf{y}}_v$.

2.3 Positional Embedding Model

We propose to learn the positional embeddings via an embedding model with an unsupervised task. In order to guide the positional embedding model to learn positional and structural information of the graph, we employ the unsupervised objective function utilized in many graph embedding models [22, 8, 30] based on the skip-gram model [17]. This objective function is a computationally efficient approximation to the cross-entropy error of predicting first-order and second-order proximities among nodes, via a negative sampling scheme. More precisely, we define

$$\mathcal{L}_{\mathcal{U}}(\{\mathbf{p}_v\}_{v\in\mathcal{V}}, G)) = \sum_{v\in\mathcal{V}} \sum_{u\in\mathcal{N}(v)} \left(-\log\sigma(\mathbf{p}_v{}^{\mathsf{T}}\mathbf{p}_u) - Q\cdot\mathbb{E}_{u'\sim P_n(v)}\log(\sigma(-\mathbf{p}_v{}^{\mathsf{T}}\mathbf{p}_{u'})) \right),$$
(4)

where $P_n(v)$ denotes the distribution negative sampling for node v; and Q is the number of negative samples per edge.

With the unsupervised objective, we utilize a positional embedding model with simple architecture, which is constructed by stacking multiple fully-connected layers. The t-th layer of the positional embedding model is computed as follows,

$$\mathbf{p}_v^t = \sigma(\mathbf{W}_{emb}^t \mathbf{p}_v^{t-1})$$
(5)

where $\mathbf{p}_v^t$ and $\mathbf{W}_{emb}^t$ are the learned positional representation for node v and the weight matrix at layer t, respectively; and σ denotes an arbitrary nonlinear activation. Even though more complicated embedding models may be used, embedding models with such simple architectures can still capture meaningful information of the graph structure according to previous works [17, 30, 22]. We only consider the transductive positional embedding model given the nature of the datasets used in our experiments. Thus, $\mathbf{p}_v^0$ is the learned initial positional embedding for node v and $\mathbf{p}_v$ is the corresponding output positional embedding from the final layer.

3 Related Works

Attention-based Models Attention mechanisms have been widely used in many sequence-based and vision-based tasks. Attention mechanisms were firstly proposed in the machine translation literature to overcome the information bottleneck problem of encoder-decoder RNN-based architectures [1, 16]. Similar models were designed to enhance image captioning architectures [28]. Since then, self-attention and attention-based models have been extensively utilized as feature extractors, which allow for variable-sized inputs [24, 21, 5, 26, 4, 32, 25]. Among those, Transformers [24] and GATs [25] are

closely related to our proposed framework. The former is a well-known language model based on attention mechanisms exclusively. Particularly, self-attention is utilized to learn word representations in a sentence, capturing their syntactic and semantic information. Note that self-attention assumes the words under the set structure, regardless their associated structural dependencies. Therefore, to explore the positional information of words in a sentence, a positional encoding was introduced in the Transformer and its variants. Following the Transformer, GATs learn node reprensentations in a graph with self-attention. More specifically, GATs mask out the interactions between unconnected nodes to somehow make use of the graph structure. This simple masking technique allows GATs to capture the co-occurrence of nodes in each neighborhood but, unfortunately, does not enable the model to fully explore the structure of the graph. Thus, we propose to learn positional embeddings to enhance the ability of GATs to fully explore the structural information of a graph. Consequently, our model, GAT-POS, can learn node representations which better capture the syntactic and semantic information in the graph.

Other Structural Embedding Enhanced GNNs As pointed by [19], GNNs struggle to fully explore the structural information of a graph, and this shortcoming limits their effective application to non-homophilic graphs, which require increased understanding of higher-level structural dependencies. To remedy this, position-aware Graph Neural Networks (PGNNs) [31] introduced the concept of position-aware node embeddings for pairwise node classification and link prediction. In order to capture the position or location information of a given target node, in addition to the architecture of vanilla GNNs, PGNN samples sets of anchor nodes, computes the distances of the target node to each anchor-set, and learns a non-linear aggregation scheme for distances over the anchor-sets. It is worth mentioning that the learned position-aware embeddings are permutation sensitive to the order of nodes in the anchor-sets. Moreover, to learn on non-homophilic graphs, [19] proposes a framework, termed Geom-GCN, in which GNNs are enhanced with latent space embeddings, capturing structural information of the graph. As opposed to GAT-POS, Geom-GCN does not support end-to-end training, and therefore their embeddings cannot be adaptively adjusted for different supervised tasks.Similar to GAT-POS, Geom-GCN also employs adaptive filters to extract features from each locality. However, the adaptive filters in Geom-GCN are merely conditioned on the latent space embeddings, which do not leverage node content information, and might be less effective than our GAT-POS' attention mechanisms based on both node content and positional information. Moreover, Geom-GCN also introduces an extended neighborhood that goes beyond the spatial neighborhood of each node, and which corresponds to the set of nodes close to the center node in the latent space. However, according to the experimental results of our paper, the extended neighborhood of the Geom-GCN is not beneficial when dealing with undirected graphs. Finally, a very recent concurrent work [13] enhances GNNs with distance encodings for GNNs. As opposed to Geom-GCN and GAT-POS, the distance encoding is built without any training process on the graph, capturing graph properties such as shortest-path-distance and generalized PageRank scores based on an encoding process.

Dataset	Homophilic Datasets			Non-homophilic Datasets		
	Cora	Citeseer	Pubmed	Chameleon	Squirrel	Actor
Homophily(β)	0.83	0.71	0.79	0.25	0.22	0.24
# Nodes	2708	3327	19717	2277	5201	7600
# Edges[*]	5429	4732	44338	36101	217073	33544
# Features	1433	3703	500	2325	2089	931
# Classes	7	6	3	5	5	5

Table 1. Summary of the datasets used in our experiments.

4 Experiments and Results

We have performed comparative evaluation experiments of GAT-POS on non-homophilic graph-structured datasets against standard GNNs (i.e., GCNs and GATs) and related embedding enhanced GNNs (i.e., Geom-GCN). Our proposed framework reaches remarkably improved performance compared to standard GNNs and outperforms Geom-GCNs. Besides, we also performed a sanity check test on homophilic datasets to validate that GAT-POS can reach comparable performance to standard GNNs.

4.1 Datasets

We utilize two Wikipedia page-page networks (i.e., Chameleon and Squirrel) and the Actor co-occurrence network (shortened as Actor) introduced in [19] as our non-homophilic datasets. We also include three widely used citation networks (i.e., Cora, Citeseer and Pubmed) [30, 10, 25] as representatives of homophilic datasets for our sanity check. Note that all graphs are converted to undirected graphs, following the widely utilized setup in other works [10, 25].

Table 1 summarizes basic statistics of each dataset, together with their homophily levels. The homophily level of a graph dataset can be demonstrated by a measure β introduced in [19], which computes the ratio of the nodes in each neighborhood sharing the same labels as the center node.

Following the evaluation setup in [19], for all graph datasets, we randomly split nodes of each class into 60%, 20%, and 20% for training, validation and testing, respectively. The final experimental results reported are the average classification accuracies of all models on the test sets over 10 random splits for all graph datasets. For each split, 10 runs of experiments are performed with different random initializations. We utilize the splits provided by [19]. Note that all datasets in our evaluation are under the transductive setting.

4.2 Methods in Comparison

In order to demonstrate that our framework can effectively enhance the performance of GATs on non-homophilic datasets, the standard GAT [25] is considered in the comparison. Furthermore, another commmonly used GNN, the GCN [10] is also included in our evaluation as a representative of spectral GNNs. We also include Geom-GCN [19] into our evaluation as a baseline of structural embedding enhanced GNNs. Specifically,

three variants of Geom-GCNs coupled with different embedding methods are considered, namely Geom-GCN with Isomap (Geom-GCN-I), Geom-GCN with Poincare embedding (Geom-GCN-P), and Geom-GCN with struc2vec embedding (Geom-GCN-S). Besides, we also include the variants of Geom-GCNs without their extended neighborhood scheme, considering only spatial neighborhoods in the graph, termed Geom-GCN-I-g, Geom-GCN-P-g and Geom-GCN-S-g respectively.

4.3 Experimental Setup

We utilize the hyperparameter settings of baseline models from [19] in our experiments since they have already undergone extensive hyperparameter search on each dataset's validation set. For our proposed GAT-POS, we perform a hyperparameter search on the number of hidden units, the initial learning rate, the weight decay, and the probability of dropout. For fairness, the hyperparameter search is performed on the same searching space as the models tuned in [19]. Specifically, the number of layers of GNN architectures is fixed to 2 and the Adam optimizer [9] is used to train all models.

For GAT-POS, according to the result of the hyperparameter search, we set the initial learning rate to 5e-3, weight decay to 5e-4, a dropout of $p = 0.5$ and the number of hidden units for the positional embedding model to 64. Besides, the activation functions of the positional embedding model and the enhanced GAT architecture are set to ReLU and ELU, respectively. The number of hidden units per attention head in the main architecture is 8 (Cora, Citeseer, Pubmed and Squirrel) and 32 (Chameleon and Actor). The number of attention heads for the hidden layer is 8 (Cora, Citeseer and Pubmed) and 16 (Chameleon, Squirrel and Actor). For all datasets, the number of attention heads in the output layer is 1 and the residual connections are employed.

4.4 Results

The results of our comparative evaluation experiments on non-homophilic and homophilic datasets are summarized in Tables 2 and 3, respectively.

| Method | Non-Homophilic Datasets | | |
	Chameleon	Squirrel	Actor
GCN	$65.22 \pm 2.22\%$	$45.44 \pm 1.27\%$	$28.30 \pm 0.73\%$
GAT	$63.88 \pm 2.42\%$	$41.19 \pm 3.38\%$	$28.49 \pm 1.06\%$
Geom-GCN-I	$57.35 \pm 1.85\%$	$31.92 \pm 1.04\%$	$29.14 \pm 1.15\%$
Geom-GCN-P	$60.68 \pm 1.97\%$	$35.39 \pm 1.21\%$	$31.92 \pm 0.95\%$
Geom-GCN-S	$57.89 \pm 1.65\%$	$35.74 \pm 1.47\%$	$30.12 \pm 0.92\%$
Geom-GCN-I-g	$61.97 \pm 2.01\%$	$39.91 \pm 1.77\%$	$32.98 \pm 0.78\%$
Geom-GCN-P-g	$60.31 \pm 2.07\%$	$37.14 \pm 1.19\%$	$31.60 \pm 1.06\%$
Geom-GCN-S-g	$63.86 \pm 1.78\%$	$42.73 \pm 2.16\%$	$31.96 \pm 1.97\%$
GAT-POS (ours)	$\mathbf{67.76 \pm 2.54}\%$	$\mathbf{52.90 \pm 1.55}\%$	$\mathbf{34.89 \pm 1.38}\%$

Table 2. The proposed positional embedding aware models outperforms their corresponding baselines in terms of accuracy on non-homophilic datasets and GAT-POS shows significantly better performance compared to Geom-GCN variations.

Non-homophilic Graph Datasets The experimental results demonstrate that our framework reaches better performance than baselines on all non-homophilic graphs. Compared to GATs, GAT-POS achieves a remarkable improvement on the three non-homophilic datasets. Our proposed framework also reaches a better performance compared to GCNs and Geom-GCNs. Unexpectedly, Geom-GCNs does not show significant improvement compared to standard GNNs on undirected graphs. This is against the previous experimental results on directed graphs. One possible reason is that the Geom-GCNs' extended neighborhood scheme cannot include extra useful neighbors in the aggregation but rather introduces redundant operations and parameters to train on undirected graphs. This is also demonstrated by the better results of the instantiations of Geom-GCNs with only spatial neighborhoods.

Homophilic Graph Datasets Recall that the evaluation of homophilic datasets is only performed as a sanity check, since GAT-POS has been tailored to exploit structural and positional information essential to non-homophilic tasks. As shown in the table, our proposed method reaches comparable results to standard GNNs. These observations are in line with our expectations on homophilic datasets since merely learning underlying group-invariances can already lead to good performance on graph datasets with high homophily.

4.5 Ablation Study

In this section, we present an ablation study to understand the contributions of the joint-training scheme. To that end, we include a variant of GAT-POS without joint-training. This variant pretrains the positional embedding model with the same unsupervised task and then freezes the learnt embeddings while training the enhanced GATs with the supervised objective. Our proposed implementation of the enhanced GAT is inspired the work applying attention mechanism on images [32]. Thus, we also introduce a variant following the architecture of Transformer from natural language processing domain, termed GAT-POS-Transformer. The difference between both architectures lies in that GAT-POS only considers the positional embeddings in the computation of attention

Method	Homophilic Datasets		
	Cora	**Citeseer**	**Pubmed**
GCN	$85.67 \pm 0.94\%$	$73.28 \pm 1.37\%$	$88.14 \pm 0.32\%$
GAT	$\mathbf{87.06 \pm 0.98}\%$	$74.79 \pm 1.89\%$	$87.51 \pm 0.43\%$
Geom-GCN-I	$84.79 \pm 2.04\%$	$78.84 \pm 1.51\%$	$89.73 \pm 0.54\%$
Geom-GCN-P	$84.68 \pm 1.59\%$	$73.77 \pm 1.59\%$	$88.15 \pm 0.57\%$
Geom-GCN-S	$85.25 \pm 1.46\%$	$74.42 \pm 2.52\%$	$84.80 \pm 0.62\%$
Geom-GCN-I-g	$85.81 \pm 1.50\%$	$\mathbf{80.05 \pm 1.59}\%$	$\mathbf{92.56 \pm 0.33}\%$
Geom-GCN-P-g	$86.48 \pm 1.49\%$	$75.74 \pm 1.58\%$	$88.49 \pm 0.56\%$
Geom-GCN-S-g	$86.50 \pm 1.43\%$	$75.91 \pm 2.26\%$	$88.55 \pm 0.53\%$
GAT-POS (ours)	$86.61 \pm 1.13\%$	$73.81 \pm 1.27\%$	$87.56 \pm 0.48\%$

Table 3. The proposed positional embedding aware models are on par with their corresponding baselines in terms of accuracy on homophilic datasets; in particular, GAT-POS's results are comparable to GATs.

coefficients but not in the neighborhood aggregation, while GAT-POS-Transformer directly injects the positional embeddings into the node features before feeding them to the attention module. Note that, GAT-POS-Transformer utilizes the architecture of the standard GATs since the input features have been enhanced by the positional embeddings prior to the attention computation. The results of the comparison are summarized in Table 4.

On Chameleon and Squirrel, both instantiations of GAT-POS with joint-training reach better results than without joint-training on average. However, it is worth noting that the joint-training also leads to relatively larger standard deviations. On Actor datasets, the difference of performances between the instantiations with and without joint-training is not notable. This might be because, on Actor dataset, the positional embedding is probably less affected by the supervised signals. Overall, enabling joint training of the main model and the positional embedding model is beneficial. Moreover, it is worth noting that the variants of GAT-POS without joint-training still outperform Geom-GCNs by a notable margin, which also highlights the advantage of having an attention mechanism incorporate semantic and structural information, compared to the weighting functions in Geom-GCNs, which only considers the structural information.

In the comparison between GAT-POS and GAT-POS-Transformer, GAT-POS reaches slightly better performance than GAT-POS-Transformer on average, but the difference is not significant. A deeper research on the architecture design of the enhanced GATs will be left for the future work.

5   Conclusion

We have presented a framework to enhance the GAT models with a positional embedding to explicitly leverage both the semantic information of each node and the structural information contained in the graph. In particular, we extended the standard GAT formulation by adding a positional embedding model, predictive of the graph context, and connecting it to the enhanced GAT model through the proposed attentional layer.Although we focused on extending the original GAT formulation, our proposed framework is compatible with most current graph deep learning based models, which leverage graph attentional layers. Moreover, it is worth mentioning that this framework is agnostic to the choice of positional embedding models as well as the particular setup of the graph attentional layers.

Experiments on a number of graph-structured datasets, especially those with more complicated structures and edges joining nodes beyond simple feature similarity, suggest that this framework can effectively enhance GATs by leveraging both graph structure and node attributes for node classification tasks, resulting in increased performances when compared to baselines as well as recently introduced methods. Finally,

	GAT-POS		GAT-POS-Transformer	
Joint-Training	✓	✗	✓	✗
Chameleon	$\mathbf{67.76 \pm 2.54}\%$	$65.75 \pm 1.81\%$	$\mathbf{65.55 \pm 2.38}\%$	$65.42 \pm 2.13\%$
Squirrel	$\mathbf{52.90 \pm 1.55}\%$	$50.63 \pm 1.29\%$	$\mathbf{51.62 \pm 1.84}\%$	$50.79 \pm 1.35\%$
Actor	$34.89 \pm 1.38\%$	$\mathbf{34.95 \pm 0.95}\%$	$\mathbf{34.97 \pm 1.27}\%$	$34.66 \pm 1.17\%$

Table 4. The Ablation Study of GAT-POS

through an ablation study, we have further emphasized the benefits of our proposed framework by showing that the performance improvements do not only come from the joint training of both parts of the model but rather from endowing GATs with node positional information.

There are several potential improvements to our proposed framework that could be addressed as future work. One is about improving the generalization ability of GAT-POS in the inductive setting. Even though GAT-POS can support the inductive setting by utilizing inductive embedding methods such as GraphSAGE, the bottleneck is still on the embedding methods, especially when the input node features are homogeneous. Another potential research direction is asynchronous training on supervised and unsupervised tasks with the overall architecture, which might be essential when learning on super large-scale graph.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR (2015)
2. Bruna, J., Zaremba, W., Szlam, A., Lecun, Y.: Spectral networks and locally connected networks on graphs. In: International Conference on Learning Representations (ICLR2014), CBLS, April 2014. pp. http–openreview (2014)
3. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1251–1258 (2017)
4. Cordonnier, J.B., Loukas, A., Jaggi, M.: On the relationship between self-attention and convolutional layers (2020)
5. Dai, Z., Yang, Z., Yang, Y., Carbonell, J.G., Le, Q., Salakhutdinov, R.: Transformer-xl: Attentive language models beyond a fixed-length context. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 2978–2988 (2019)
6. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in neural information processing systems. pp. 3844–3852 (2016)
7. Gao, H., Ji, S.: Graph u-nets. In: International Conference on Machine Learning. pp. 2083–2092 (2019)
8. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in neural information processing systems. pp. 1024–1034 (2017)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (Poster). OpenReview.net (2017)
11. Lanczos, C.: An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. United States Governm. Press Office Los Angeles, CA (1950)
12. Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., Teh, Y.W.: Set transformer: A framework for attention-based permutation-invariant neural networks. In: International Conference on Machine Learning. pp. 3744–3753 (2019)
13. Li, P., Wang, Y., Wang, H., Leskovec, J.: Distance encoding–design provably more powerful gnns for structural representation learning. arXiv preprint arXiv:2009.00142 (2020)
14. Li, Y., Tarlow, D., Brockschmidt, M., Zemel, R.: Gated graph sequence neural networks. arXiv preprint arXiv:1511.05493 (2015)

15. Liao, R., Zhao, Z., Urtasun, R., Zemel, R.S.: Lanczosnet: Multi-scale deep graph convolutional networks. In: 7th International Conference on Learning Representations, ICLR 2019 (2019)
16. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1412–1421 (2015)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
18. Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M.: Geometric deep learning on graphs and manifolds using mixture model cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5115–5124 (2017)
19. Pei, H., Wei, B., Chang, K.C.C., Lei, Y., Yang, B.: Geom-gcn: Geometric graph convolutional networks. arXiv preprint arXiv:2002.05287 (2020)
20. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Transactions on Neural Networks **20**(1), 61–80 (2008)
21. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). pp. 464–468 (2018)
22. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web. pp. 1067–1077 (2015)
23. Vashishth, S., Yadav, P., Bhandari, M., Talukdar, P.P.: Confidence-based graph convolutional networks for semi-supervised learning. In: AISTATS. Proceedings of Machine Learning Research, vol. 89, pp. 1792–1801. PMLR (2019)
24. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
25. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: ICLR (Poster). OpenReview.net (2018)
26. Wang, X., Tu, Z., Wang, L., Shi, S.: Self-attention with structural position representations. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 1403–1409 (2019)
27. Weston, J., Ratle, F., Collobert, R.: Deep learning via semi-supervised embedding. In: Proceedings of the 25th international conference on Machine learning. pp. 1168–1175 (2008)
28. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning. pp. 2048–2057 (2015)
29. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=ryGs6iA5Km
30. Yang, Z., Cohen, W.W., Salakhutdinov, R.: Revisiting semi-supervised learning with graph embeddings. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48. pp. 40–48. JMLR. org (2016)
31. You, J., Ying, R., Leskovec, J.: Position-aware graph neural networks. In: International Conference on Machine Learning. pp. 7134–7143 (2019)
32. Zhao, H., Jia, J., Koltun, V.: Exploring self-attention for image recognition. arXiv preprint arXiv:2004.13621 (2020)