Learning Conjoint Attentions for Graph Neural Nets

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper, we present Conjoint Attentions (CAs), a series of novel learning-to-attend strategies for graph neural networks (GNNs). Different from previous graph attentions computing weights for feature aggregation solely bestowing layer-wise node features propagated inside GNN, CAs are capable of adaptively leveraging heterogeneous learnable factors either inside or outside of GNNs to compute attention scores. The significant node features that are reckoned by the conjoint criteria are therefore more likely to be propagated in the GNN. Given new conjoint attention strategies, we then propose Graph Conjoint Attention Networks (CATs) that can learn representations embedded with significant latent features deemed by the conjoint attentions. Besides, we theoretically validate the discriminative capacity of CATs. CATs utilizing the proposed conjoint attention strategies have been extensively tested in prevalent benchmarking datasets and comprehensively compared with state-of-the-art baselines. The obtained outstanding performance demonstrates the effectiveness of the proposed conjoint attentions.

1 Introduction

Graph neural networks (GNNs) have achieved a terrific success in learning in graph structured data. Amongst these extraordinary GNNs, attention-based GNNs [26] have drawn much interest lately, and have been applied to solve various real-world problems, such as node classification [15, 26], image segmentation [28], and social recommendations [24].

Empirical attention mechanisms adopted by GNNs aim to leverage the node features to compute the normalized correlations between pairs of nodes that are observed to connect. Treating the normalized correlations (attention scores) as the relative weights between node pairs, attention-based GNN may utilize them to perform a weighted sum of node features which are ultimately propagated to higher layers. Compared with other GNNs, especially those graph convolutional networks [1, 15, 17] which aggregate node features following different predefined strategies, attention-based GNNs provide a dynamical way for feature aggregation, which allows more correlated features from neighboring nodes to be propagated in the multi-layer neural architecture. Representations implanted with multi-layer correlated features are consequently learned by attention-based GNNs and can be used for various downstream tasks.

Though effective, empirical graph attentions always confront the following problems when aggregating node features. First, the computation of attention scores relies solely on the layer-wise node features inside the neural nets, other possible enlightenment which may lead to more appropriate attention scores is overlooked. Second, the empirical attention heavily leaning on the node features may cause over-fitting in the training stage of neural nets [27]. The predictive power of prevalent

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

attention-based GNNs is consequently limited, and most previous graph attention mechanisms are incapable of flexibly learning effective attention scores when heterogeneous relevance is given.

To overcome the mentioned challenges, in this paper, we propose a class of generic graph attention mechanisms, dubbed Conjoint Attentions (CAs). Given CAs, we construct Graph Conjoint Attention Networks (CATs) for different downstream analytical tasks. Different from previous graph attentions, CAs are able to flexibly compute the attention scores not only relying on the layer-wise node features, but also leveraging the information possibly outside of the neural networks. As a result, CATs are able to learn representations from the features that are considered to be significant by various criteria, so that the corresponding predictive power is improved. The main contributions of the paper are summarized as follows.

- We propose Conjoint Attentions (CAs) for GNNs. Different from prevalent graph attentions entirely relying on node features, CAs are able to flexibly utilize heterogeneous learnable factors to compute more appropriate attention scores for node feature aggregation. The learning performance of CA-based GNNs is thereby improved.
- For the first time, we theoretically analyze the expressive power of graph attention layers considering heterogeneous factors for node feature aggregation. And the discriminant capacity of such attention layers, i.e., CA layers is validated.
- Given CA layers, we build Graph Conjoint Attention Networks (CATs) for various learning tasks. The proposed CATs have been extensively tested on prevalent benchmarking datasets and have compared with a number of state-of-the-art baselines. The outstanding results demonstrate the effectiveness of the proposed attention mechanisms.

2 Related works

To effectively learn low-dimensional representations in graph structured data, many graph neural networks have been proposed. According to the ways through which GNNs define the layer-wise operators for feature aggregation, GNNs can generally be categorized as spectral, and spatial ones [31].

Spectral GNNs-The layer-wise function for feature aggregation in spectral GNNs is defined according to the spectral representation of the graph. For example, Spectral CNN [3] constructs the convolution layer based on the eigen-decomposition of graph Laplacian in the Fourier domain. However, such layer is computationally demanding. Aiming to reduce such computational burden, several approaches adopting the convolution operators which are based on simplified/approximate spectral graph theory are proposed. First, parameterized filters with smooth coefficients are introduced for Spectral CNN to allow it to consider spatially localized nodes in the graph [13]. Chebyshev expansion is then introduced in [7] to approximate graph Laplacian rather than directly perform eigen-decomposition of it. Finally, the graph convolution filter is further simplified by only considering first, or higher order of connected neighbors [15, 30], so as to further make convolution layer computationally efficient.

Spatial GNNs-In contrast, spatial GNNs define the convolution operators for feature aggregation directly making use of local structural properties of the central node. The key of spatial GNNs is consequently how to design an appropriate function for aggregating the effect brought by the features of candidate neighbors selected according to a proper sampling strategy. To achieve this, it sometimes requires to learn a weight matrix according to node degree [8], utilize the power of transition matrix to preserve the neighbor importance [1, 4, 16, 17, 34], extract the normalized neighbors [21], or sample a fixed number of neighbors [12, 36].

As representative spatial GNNs, attention-based GNNs [11, 26] has shown a promising performance in various graph learning tasks. What makes them effective in learning graph representations is they adopt the attention mechanism, which has been successfully used in machine reading and translation [5, 19], and video processing [32], to compute the node-feature-based attention weights (attention scores) between a central node and its one-hop neighbors (including the central node itself). Then, attention-based GNNs utilize the attention scores to obtain a weighted aggregation of node features

which are propagated to the next layer. As a result, those neighbors possessing similar features may impact more on the center node, and meaningful representations can be inferred by GATs. Having investigated previous efforts to graph neural networks, we observe that the computation of empirical graph attentions heavily relies on layer-wise node features, while other enlightenment, e.g., structural properties hidden in the graph data, is overlooked. This motivate us to propose novel attention mechanisms to alleviate the limitations of prevalent attention-based graph neural networks.

3 Graph conjoint attention networks

In this section, we elaborate the proposed conjoint attention mechanisms, which are the cornerstones for building layers of novel attention-based graph neural networks. Mathematical preliminaries and notations used in the paper are firstly illustrated. Then, how to construct neural layers utilizing various conjoint attention mechanisms are introduced. Given the formulated conjoint attention layers, we finally construct graph conjoint attention networks (CATs).

3.1 Notations and preliminaries

Throughout this paper, we assume a graph $G = \{V, E\}$ containing N nodes, $|E|$ edges, where V and E respectively represent the node and edge set. We use $\mathbf{A} \in \{0,1\}^{N\times N}$ and $\mathbf{X} \in \mathbb{R}^{N\times D}$ to represent graph adjacency matrix and node feature matrix, respectively. $\mathcal{N}_i$ denotes the union of node i and its one-hop neighbors. $\mathbf{W}^l$ and $\{\mathbf{h}_i^l\}_{i=1,...N}$ denote the weight matrix and features of node i at lth layer of CATs, respectively, and $\mathbf{h}^0$ is set to be the input feature, i.e., $\mathbf{X}$. For the nodes in $\mathcal{N}_i$, their possible feature vectors form a multiset $M_i = (S_i, \mu_i)$, where $S_i = \{s_1, ...s_n\}$ is the ground set of M_i which contains the distinct elements existing in M_i, and $\mu_i : S_i \to \mathbb{N}^\star$ is the multiplicity function indicating the frequency of occurrence of each distinct s in M_i.

3.2 Enlightenment for conjoint attentions

As aforementioned, the propose conjoint attentions are able to make use of information either inside or outside of neural architectures to compute novel attention scores. Here, we propose a simple but effective way for CAs to capture the enlightenment outside of GNNs. Let $\mathbf{C}_{ij}$ be the particular latent knowledge between ith and jth node in the graph. It can be obtained through the following generic generating function:

$$\mathbf{C}_{ij} = \underset{\phi(\mathbf{C})_{ij}}{\arg\min}\, \Psi(\phi(\mathbf{C})_{ij}, \mathbf{Y}_{ij}), \tag{1}$$

where $\Psi(\cdot)$ represents a distance function and $\phi(\cdot)$ stands for a operator transforming $\mathbf{C}$ to the same dimensionality of $\mathbf{Y}$. Given the generic generating function in Eq. (1), it is known that many effective paradigms for latent feature learning can be used for the subsequent computation of conjoint attentions, if the prior feature matrix $\mathbf{Y}$ is appropriately provided. In this paper, we consider two different generation processes that are simply based on the graph adjacency $\mathbf{A}$. Let $\Psi(\cdot)$ be the euclidean distance, when $\phi(\mathbf{C})_{ij} \doteq \mathbf{V}\mathbf{V}_{ij}^T$, we have:

$$\mathbf{C}_{ij} = \underset{\mathbf{V}\mathbf{V}_{ij}^T}{\arg\min}(\mathbf{A}_{ij} - \mathbf{V}\mathbf{V}_{ij}^T)^2, \tag{2}$$

where we use a proper low-dimensional matrix $\mathbf{V}$ to approximate $\mathbf{C}$ to reduce the computational burden. As it shows in Eq. (2), $\mathbf{C}_{ij}$ attempts to acquire the local information around node i and j based on matrix factorization (MF). If $\phi(\mathbf{C})_{ij} \doteq \sum_j \mathbf{V}\mathbf{V}_{ij}^T\mathbf{A}_{ij}$, we have:

$$\mathbf{C}_{ij} = \underset{\mathbf{V}\mathbf{V}_{ij}^T}{\arg\min}(\mathbf{A}_{ij} - \sum_j \mathbf{V}\mathbf{V}_{ij}^T\mathbf{A}_{ij})^2. \tag{3}$$

As what shows in Eq. (3), $\mathbf{C}_{ij}$ is the coefficient of self-expressiveness [10] (SC) which may describe the relation between node i and j from a global perspective. It is known that both aforementioned properties are never considered by empirical graph attention mechanisms. We believe considering either of them in the conjoint attentions may lead better attention scores for feature aggregation. Note that other types of $\mathbf{C}$ may also be feasible for the proposed attention mechanisms, as long as they are able to capture some meaningful property which is not possessed by the node features in GNN.

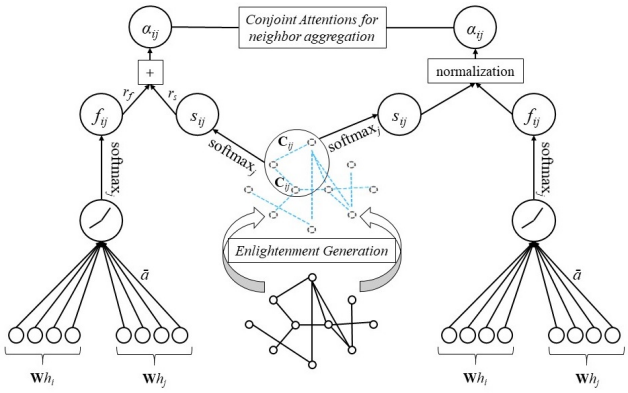

Figure 1: Graphical illustration of the conjoint attention layer used in CATs. Left: CA mechanism using *Implicit direction* strategy (CAT-I). Right: CA mechanism using *Explicit direction* strategy (CAT-E). Both two mechanisms consider learnable external enlightenment.

3.3 Conjoint attention layer

Having obtained a proper $\mathbf{C}$, we are now presenting the conjoint attention layer, which is the core module for building CATs and will be used in our experiments. Different from the attention layers utilized by other GNNs, the attention layer proposed here adopts novel attention mechanisms, i.e., conjoint attentions, which may automatically identify the relative significance between input features and other information that is used for computing attention scores. Appropriate attention scores between central node and its neighbors can be obtained and meaningful representations can be inferred by CATs.

Given a set of node features $\{\mathbf{h}_i^l\}_{i=1,\dots N}$, each $\mathbf{h}_i^l \in R^{D^l}$, the conjoint attention layer is to map them into D^{l+1} dimensional space $\{\mathbf{h}_i^{l+1}\}_{i=1,\dots N}$, according to the correlations node features and other possible enlightenment. The contextual correlation between two connected nodes, say v_i and v_j is firstly obtained. To do so, we directly adopt the feature-based attention mechanism utilized by previous graph attention networks (GAT) [26]:

$$f_{ij} = \frac{\exp(\text{LeakyReLU}(\tilde{\mathbf{a}}^T(\mathbf{W}^l\mathbf{h}_i^l \parallel \mathbf{W}^l\mathbf{h}_j^l)))}{\sum_{k\in\mathcal{N}_i}\exp(\text{LeakyReLU}(\tilde{\mathbf{a}}^T(\mathbf{W}^l\mathbf{h}_i^l \parallel \mathbf{W}^l\mathbf{h}_k^l)))}, \tag{4}$$

where $\tilde{\mathbf{a}} \in \mathbb{R}^{2D^{l+1}}$ is a vector of parameters of the feedforward layer, $\parallel$ stands for the concatenation function, and $\mathbf{W}^l$ is $D^{l+1} \times D^l$ variable matrix for feature mapping. Given Eq. (4), the proposed CA layer capture the feature correlations between connected nodes (first-order neighbors) by computing the similarities w.r.t. node features mapped to next layer.

As what has been mentioned, determining the attention scores solely based on node features inside a GNN may lead it to overlook other inspiring information hidden in the graph. To overcome this issue, the proposed CA layer attempts to learn the additional latent features as described in Section 3.2, and utilize novel paradigms to appropriately compute the attention scores. Given any learnable parameter $\mathbf{C}_{ij}$ between two nodes, CA layer additionally obtain a supplementary correlation as follows:

$$s_{ij} = \frac{\exp(\mathbf{C}_{ij})}{\sum_{k\in\mathcal{N}_i}\exp(\mathbf{C}_{ik})}. \tag{5}$$

Given f_{ij} and s_{ij}, we propose two different strategies to compute the conjoint attention scores. The first mechanism is dubbed *Implicit direction*. It aims at computing the attention scores whose relative significance between supplementary and feature correlations can be automatically acquired. To do so, each CA layer introduces two learnable parameters, g_f and g_s to determine the relative significance between feature and supplementary correlations and it can be obtained as follows:

$$r_f = \frac{\exp(g_f)}{\exp(g_s)+\exp(g_f)}, r_s = \frac{\exp(g_s)}{\exp(g_s)+\exp(g_f)}, \tag{6}$$

where r_s or r_f represents the normalized significance related to different types of learnable features. Given them, CAT is able to compute the attention score based on *Implicit direction* strategy:

$$\alpha_{ij} = \frac{r_f \cdot f_{ij} + r_s \cdot s_{ij}}{\sum_{k \in \mathcal{N}_i} [r_f \cdot f_{ik} + r_s \cdot s_{ik}]} = r_f \cdot f_{ij} + r_s \cdot s_{ij}. \tag{7}$$

Given the attention mechanism shown in Eq. (7), α_{ij} attempts to capture the weighted mean attention in terms of various node-node correlations, which are ones inside/outside the GNN. Compared with the attention mechanism solely based on features of one-hop neighbors, α_{ij} computed by Eq. (7) may be softly adjusted according to the implicit impact brought by the supplementary enlightenment. Moreover, the relative significance r can also be automatically inferred as it is involved into the back propagation process. More smooth and appropriate attention scores can thereby be computed by the CA layer for learning meaningful representations.

To enhance the impact of external enlightenment, the CA layer has another strategy, named as *Explicit direction*, to compute attention scores between neighbors. Given f_{ij} and s_{ij}, the attention scores obtained via *Explicit direction* strategy is defined as follows:

$$\alpha_{ij} = \frac{f_{ij} \cdot s_{ij}}{\sum_{k \in \mathcal{N}_i} f_{ik} \cdot s_{ik}}. \tag{8}$$

Compared with Eq. (7), s_{ij} explicitly influences the magnitude of f_{ij}, so that those node pairs which are irrelevant in terms of $\mathbf{C}_{ij}$ are impossible to assign with high attention weights. Utilizing *Explicit direction* strategy, the CA layer becomes more external-enlightenment dependent when performing message passing to the higher layers in the neural architecture.

Having obtained the conjoint attention scores, the CA layer is now able to compute a linear combination of features corresponding to each node and its neighbors as the output, which will be either propagated to the higher layer, or be used as the final representations for subsequent learning tasks. The mentioned output features can be computed as follows:

$$\mathbf{h}_i^{l+1} = (\alpha_{ii} + \epsilon \cdot \frac{1}{|\mathcal{N}_i|})\mathbf{W}\mathbf{h}_i^l + \sum_{j \in \mathcal{N}_i, j \neq i} \alpha_{ij} \mathbf{W}\mathbf{h}_j^l, \tag{9}$$

where $\epsilon \in (0, 1)$ is a learnable parameter which may improve the expressive capability of the proposed CA layer.

3.4 Construction of graph conjoint attention networks (CATs)

In Fig. 1, the conjoint attention layer utilizing the attention mechanisms proposed in this paper are graphically illustrated. We are now able to construct Graph Conjoint Attention Networks (CATs) using a particular number of CA layers proposed. In practice, we also adopt the multi-head attention strategy [25] to stabilize the learning process. CATs may either concatenate the node features from multiple hidden layers as the input for next layer, or compute the average of node features obtained by multiple units of output layers as the final node representations. The details on how to implement multi-head attention in graph neural networks can be checked in [26].

4 Theoretical analysis

Study on the expressive power of various GNNs has drawn much attention in the recent. It concerns whether a given GNN can distinguish different structures where vertices possessing various vectorized features. It has been found that what the neighborhood aggregation functions of all message-passing GNNs aim at are analogous to what 1-dimensional Weisfeiler-Lehman test (1-WL test), which is injective and iteratively operated in Weisfeiler-Lehman algorithm [29, 33, 37], does. As a result, all message-passing GNNs are as most powerful as 1-WL test [33]. The theoretical validation of the expressive power of a given GNN thereby lies in whether those adopted aggreation/readout functions are homogeneous to 1-WL test.

Having the effective framework evaluating the expressive power of a given message-passing GNN, one may naturally be interested in whether the proposed CA layers can distinguish all different graph structures as 1-WL test does. To do so, we firstly show that neighborhood aggregation function (Eq. (9)) without the term for improving expressive capability (i.e., $\epsilon \cdot \frac{1}{|\mathcal{N}_i|}\mathbf{W}\mathbf{h}_i^l$ in Eq. (9)) still fails to discriminate some graph structures possessing certain topological properties. Then, by integrating the term of improving expressive capability, all the proposed CA layers are able to distinguish all those graph structures that previously cannot be discriminated.

For the function of neighborhood aggregation solely utilizing the strategy shown in Eq. (7), we have the following theorem pointing out the conditions under which the aggregation function fails to distinguish different structures.

Theorem 1 *Assume the feature space $\mathcal{X}$ is countable and the aggregation function using the weights computed by Eq. (7) is represented as $h(c, X) = \sum_{x \in X} \alpha_{cx} g(x)$, where c is the feature of center node, $X \in \mathcal{X}$ is a multiset containing the feature vectors from nodes in $\mathcal{N}_i$, $g(\cdot)$ is a function for mapping input feature X, and α_{cx} is the weight between $g(c)$ and $g(x)$. For all g and the strategy in Eq. (7), $h(c_1, X_1) = h(c_2, X_2)$ if and only if $c_1 = c_2$, $X_1 = \{S, \mu_1\}$, $X_2 = \{S, \mu_2\}$, and $\sum_{y=x, y \in X_1} f_{c_1 y} - \sum_{y=x, y \in X_2} f_{c_2 y} = q[\sum_{y=x, y \in X_2} s_{c_2 y} - \sum_{y=x, y \in X_1} s_{c_1 y}]$, for $q = \frac{r_s}{r_f}$ and $x \in S$. In other words, h will map different multisets into the same embedding iff the multisets have same central node feature, same underlying set, and the difference in feature-based scores is proportional ($\frac{r_s}{r_f}$) to the opposite of that in weights corresponding to supplementary enlightenment.*

We leave the proof of all the theorems and corollaries in the appendix. For the aggregation function utilizing the strategy shown in Eq. (8), we have the following theorem indicating the structures which cannot be correctly distinguished.

Theorem 2 *Under the same assumptions shown in Theorem 1, for all g and the strategy in Eq. (8), $h(c_1, X_1) = h(c_2, X_2)$ if and only if $c_1 = c_2$, $X_1 = \{S, \mu_1\}$, $X_2 = \{S, \mu_2\}$, and $q \cdot \sum_{y=x, y \in X_1} \phi(\mathbf{C}_{c_1 x}) = \sum_{y=x, y \in X_2} \phi(\mathbf{C}_{c_2 y})$, for $q > 0$ and $x \in S$, where $\phi(\cdot)$ is an function for mapping values to $\mathbb{R}^+$. In other words, h will map different multisets into the same embedding iff the multisets have same central node feature, same node features whose corresponding mapped supplementary enlightenment are proportional.*

Theorems 1 and 2 indicate that the CA layers may still fail to distinguish some structures, if they exclude the improving term shown in Eq. (9). However, GNNs utilizing Eqs. (7) or (8) can still be more expressively powerful than classical GATs. As node features and external enlightenment are heterogeneous, intuitively, structures satisfying the stated conditions should be infrequent. This may well explain why those GNNs concerning including external factors, e.g., some structural properties into the computation of weights for feature aggregation may experimentally perform better than GATs. However, when distinct multisets with corresponding properties meet the conditions mentioned in Theorems 1 and 2, the attention mechanisms solely based on Eqs. (7) or (8) cannot correctly distinguish such multisets. Thus, GNNs only utilizing Eqs. (7) or (8) as the feature aggregation function fail to reach the upper bound of expressive power of all message-passing GNNs, i.e., 1-WL test.

However, we are able to readily improve the expressive power of CATs to meet the condition of 1-WL test by slightly modifying the aggregation function as Eq. (9) shows. Then, the newly obtained conjoint attention scores can be used to aggregate the node features passed to the higher layers. Next, we prove that the proposed conjoint attention mechanisms (Eqs. (7)-(9)) reach the upper bound of message-passing GNNs via showing they can distinguish those structures possessing the properties mentioned in Theorems 1 and 2.

Corollary 1 *Let $\mathcal{T}$ be the attention-based aggregator shown in Eq. (9) that consider one of the strategies in Eq. (7) or (8) and operates on a multiset $H \in \mathcal{H}$, where $\mathcal{H}$ is a node feature space mapped from the countable input feature space $\mathcal{X}$. A $\mathcal{H}$ exists so that utilizing attention-based*

aggregator shown in Eq. (9), $\mathcal{T}$ can distinguish all different multisets in aggregation that it previously cannot discriminate.

Based on the performed analysis, the expressive power of CATs is theoretically stronger than state-of-the-art attention-based GNNs, e.g., GATs [26].

Remarks: It should be noted that, all the theoretical validations presented in this paper assume the feature space is countable. Previous works have proved that GNNs utilizing a simplex function for feature aggregation only can pass the 1-WL test in countable feature space [6]. To ensure a GNN is also injective when the feature space is uncountable, heterogeneous functions for feature aggregation have to be used in each layer of the graph neural architecture. Thus, the expressive power of the proposed CATs can be further improved in uncountable feature space when the conjoint attention mechanisms are appropriately combined with other types of functions for feature aggregation.

5 Experiments and analysis

In this section, we evaluate the proposed Graph Conjoint Attention Networks against a variety of prevalent baselines, on widely used network datasets.

5.1 Experimental set-up

Baselines for comparison-To validate the effectiveness of the proposed CATs, we compare them with a number of state-of-the-art baselines, including Arma filter GNN (ARMA) [2], Simplified Graph convolutional networks (SGC) [30], Personalized Pagerank GNN (APPNP) [16], Graph attention networks (GAT) [26], Jumping Knowledge Networks (JKNet) [34], Graph convolutional networks (GCN) [15], GraphSAGE [12], and mixture model CNN (MoNet) [20]. Based on the experimental results previously reported, these baselines may represent the most advanced techniques for learning in graph structured data.

Testing datasets-Five widely-used network datasets, which are Cora, Citeseer, Pubmed [18, 22], CoauthorCS [23], and OGB-Arxiv [14], are used in our experiments. Cora, Citeseer, and Pubmed are three classical network datasets for validating the effectiveness of GNNs. However, it is recently found that these three datasets sometimes may not effectively validate the predictive power of different graph learning approaches, due to the relatively small data size and data leakage [14, 23]. Thus, more massive datasets having better data quality have been proposed to evaluate the performance of different approaches [9, 14]. In our experiment, we additionally use CoauthorCS and OGB-Arxiv as testing datasets. The details of all benchmarking sets can be checked in the appendix.

Evaluation and experimental settings-Two learning tasks, semi-supervised node classification and semi-supervised node clustering are considered in our experiments. For the training paradigms of both two learning tasks, we closely follow the experimental scenarios established in the related works [14, 15, 26, 35]. For the testing phase of different approaches, we use the test splits that are publicly available for classification tasks, and all nodes for clustering tasks. The effectiveness of all methods is validated through evaluating the classified nodes using $Accuracy$. In the training stage, we construct the two-layer network structure (i.e., one hidden layer possessed) for all the baselines and different versions of CATs. In each set of testing data, all approaches are run ten times to obtain the statistically steady performance. As for other details related to experimental settings, we leave them in appendix.

5.2 Results on node classification

The results on semi-supervised node classification are summarized in Table 1. As the table shows, CATs utilizing different attention strategies generally perform better than any other baseline in all the testing datasets. Specifically, CAT utilizing *Implicit direction* (CAT-I) may perform better than all the compared baselines in all the five datasets. CAT utilizing *Explicit direction* (CAT-E) is better than other compared baselines in four datasets out of five, except the case of CoauthorCS. In that dataset, CAT-E may rank the second-best when compared with other baselines.

Table 1: Average $Accuracy$ on semi-supervised node classification. Bold fonts mean CAT obtains a better performance than any other baseline.

	Cora	**Cite**	**Pubmed**	**CoauthorCS**	**OGB-Arxiv**
MoNet	81.96 ± 0.50	64.22 ± 0.16	79.78 ± 0.33	91.96 ± 0.75	47.71 ± 0.27
GCN	81.42 ± 0.19	71.60 ± 0.73	79.66 ± 0.39	91.54 ± 0.43	71.78 ± 0.16
GraphSAGE	81.12 ± 0.41	71.06 ± 0.64	79.04 ± 0.62	93.06 ± 0.80	69.07 ± 0.27
JKNet	78.34 ± 0.02	65.88 ± 0.01	79.88 ± 0.01	89.62 ± 0.01	64.91 ± 0.01
GAT	83.84 ± 0.61	70.36 ± 0.42	81.50 ± 0.47	92.80 ± 0.41	72.39 ± 0.07
APPNP	82.80 ± 0.32	72.38 ± 0.50	82.62 ± 0.37	89.16 ± 0.65	63.16 ± 0.54
SGC	81.90 ± 0.01	71.40 ± 0.01	82.42 ± 0.04	93.60 ± 0.01	61.06 ± 0.09
ARMA	80.06 ± 0.57	70.00 ± 0.66	76.46 ± 0.58	86.28 ± 0.75	68.77 ± 0.17
CAT-I-MF	$\mathbf{85.38 \pm 0.16}$	$\mathbf{73.22 \pm 0.19}$	$\mathbf{83.90 \pm 0.24}$	$\mathbf{93.74 \pm 0.14}$	$\mathbf{72.89 \pm 0.06}$
CAT-I-SC	$\mathbf{85.50 \pm 0.22}$	$\mathbf{73.18 \pm 0.22}$	$\mathbf{84.28 \pm 0.20}$	$\mathbf{93.70 \pm 0.11}$	$\mathbf{72.85 \pm 0.04}$
CAT-E-MF	$\mathbf{85.56 \pm 0.19}$	$\mathbf{73.24 \pm 0.21}$	$\mathbf{83.60 \pm 0.17}$	93.40 ± 0.12	$\mathbf{72.81 \pm 0.09}$
CAT-E-SC	$\mathbf{85.40 \pm 0.36}$	$\mathbf{73.02 \pm 0.24}$	$\mathbf{84.02 \pm 0.24}$	93.30 ± 0.11	$\mathbf{72.83 \pm 0.11}$

Table 2: Average $Accuracy$ on semi-supervised node clustering. Bold fonts mean CAT obtains a better performance than any other baseline.

	Cora	**Cite**	**Pubmed**	**CoauthorCS**	**OGB-Arxiv**
MoNet	79.42 ± 0.86	63.07 ± 0.11	79.39 ± 0.61	88.75 ± 0.54	53.08 ± 0.15
GCN	74.25 ± 0.13	63.36 ± 0.87	77.83 ± 0.75	89.74 ± 0.53	75.02 ± 0.07
GraphSAGE	78.46 ± 0.56	69.00 ± 0.17	79.52 ± 1.13	90.16 ± 0.53	73.50 ± 0.13
JKNet	75.95 ± 0.01	65.12 ± 0.03	79.52 ± 0.01	86.66 ± 0.01	71.28 ± 0.01
GAT	81.39 ± 0.18	69.20 ± 0.28	80.88 ± 0.33	90.09 ± 0.15	76.04 ± 0.38
APPNP	79.93 ± 0.82	70.55 ± 0.85	82.81 ± 0.32	85.93 ± 0.39	69.73 ± 0.67
SGC	79.38 ± 0.02	69.71 ± 0.02	81.64 ± 0.01	90.13 ± 0.01	71.09 ± 0.37
ARMA	77.70 ± 0.99	68.38 ± 0.87	77.29 ± 1.11	84.72 ± 0.29	69.24 ± 0.12
CAT-I-MF	$\mathbf{82.17 \pm 0.11}$	$\mathbf{71.15 \pm 0.12}$	82.77 ± 0.07	$\mathbf{90.26 \pm 0.22}$	$\mathbf{77.72 \pm 0.07}$
CAT-I-SC	$\mathbf{82.26 \pm 0.13}$	$\mathbf{71.17 \pm 0.15}$	$\mathbf{82.86 \pm 0.07}$	$\mathbf{90.29 \pm 0.21}$	$\mathbf{77.01 \pm 0.16}$
CAT-E-MF	$\mathbf{81.98 \pm 0.19}$	$\mathbf{71.21 \pm 0.12}$	82.40 ± 0.08	89.66 ± 0.22	$\mathbf{76.93 \pm 0.16}$
CAT-E-SC	$\mathbf{82.01 \pm 0.24}$	$\mathbf{71.11 \pm 0.24}$	82.61 ± 0.14	89.72 ± 0.15	$\mathbf{76.98 \pm 0.08}$

5.3 Results on node clustering

Node clustering can be more challenging as all the nodes containing various potential structures in the graph are used in the testing phase. CATs still performs robustly when compared with other baselines in this challenging task. As Table 2 shows, the *Implicit direction* strategy utilized by CAT may still ensure the proposed neural architecture to outperform other compared baselines in all the datasets. As for CAT utilizing *Explicit direction*, it can rank the best in three datasets out of five. While in the rest two datasets, Pubmed and CoauthorCS, the performance of CAT-E is very close to the best. Based on the robust performance shown in Tables 1 and 2, CAT is seen as one of the most effective GNNs for various graph learning tasks.

5.4 Ablation study and sensitivity test

To further investigate whether the proposed CA mechanisms are effective in improving the predictive power of CATs, we compared the performance on different attention scores computed using various factors. The corresponding results are summarized in Fig. 2. As the figure shows, for both classification and clustering tasks, the proposed CA mechanisms perform statistically better than others whose attention scores are computed using node features, or different factors learned from the graph. It is also observed that the consideration of structural attentions (SC and MF) may also improve the performance of attention-based GNNs. Such results may experimentally validate the theoretical analysis in Section 4.

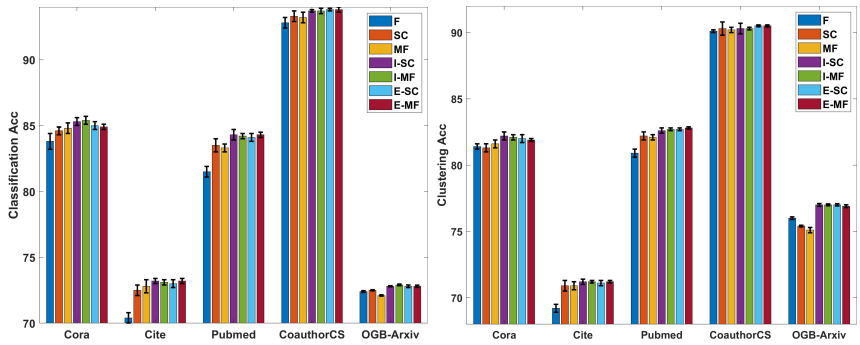

Figure 2: Performance comparison on computing attention scores using different factors

As external enlightenment ($\mathbf{C}$) is learnable, we allow the learning of $\mathbf{C}$ to be simultaneously done with the training of CATs. Thus, the loss function of CATs becomes $L = L_{predict} + \lambda L_{ext}$, where $L_{predict}$ is the task-specific loss (e.g., cross entropy loss for node classification), L_{ext} can be the sum of all possible items shown in either Eq. (2) or (3), and λ is a balancing parameter for controlling the relative significance of external enlightenment learning. To investigate the model sensitivity against λ, we set $\lambda = [10^{-4}, 10^{-2}, 10^{-1}, 1, 5, 10, 20, 50, 100]$ and run CATs in all datasets to test their performance on different settings of λ. The results have been plotted in Fig. 3. As depicted, all the versions of CATs may perform robustly under a wide range settings of λ. For simplicity, we set $\lambda = 0.01$ for CATs in our experiments.

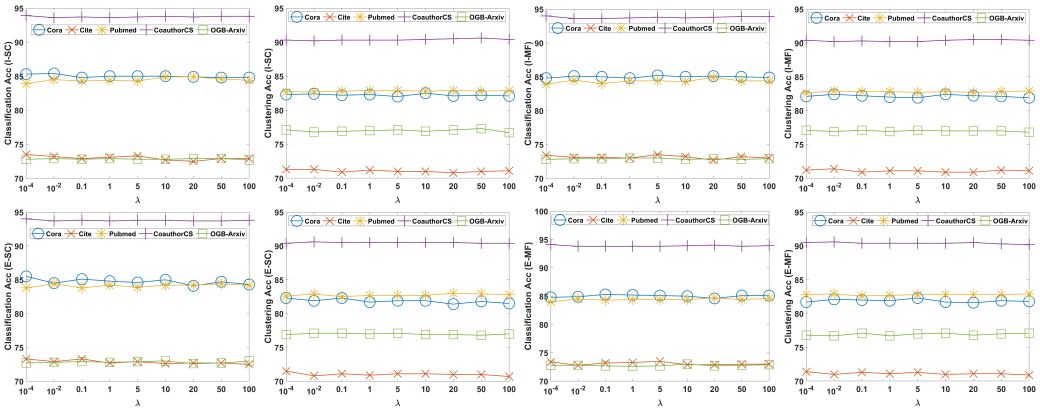

Figure 3: Sensitivity test on λ

6 Conclusion

In this paper, we propose a class of novel attention strategies, dubbed Conjoint Attentions (CAs) to construct attention-based GNNs (CATs). Different from empirical graph attentions, CAs can flexibly make use of both node features and other learnable factors perhaps outside of GNN to compute appropriate weights for feature aggregation. Besides, the expressive power of CATs is theoretically validated. The effectiveness of CATs has been verified via being compared with a number of prevalent approaches in different learning tasks. In future, we will further improve the effectiveness of CATs by considering different properties hidden in the network data and explore CATs' applicability by using them in multi-view and heterogeneous network data.

References

[1] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In *Advances in neural information processing systems*, pages 1993–2001, 2016.

[2] Filippo Maria Bianchi, Daniele Grattarola, Lorenzo Livi, and Cesare Alippi. Graph neural networks with convolutional arma filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[3] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and deep locally connected networks on graphs. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[4] Julian Busch, Jiaxing Pi, and Thomas Seidl. Pushnet: Efficient and adaptive neural message passing. *arXiv preprint arXiv:2003.02228*, 2020.

[5] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561, 2016.

[6] Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. *arXiv preprint arXiv:2004.05718*, 2020.

[7] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[8] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[9] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020.

[10] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781, 2013.

[11] Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, et al. Hyperbolic attention networks. In *International Conference on Learning Representations*, 2018.

[12] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.

[13] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[14] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020.

[15] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[16] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank.

[17] Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. In *Advances in Neural Information Processing Systems*, pages 13354–13366, 2019.

[18] Qing Lu and Lise Getoor. Link-based classification. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, pages 496–503, 2003.

[19] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, 2015.

[20] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017.

[21] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023, 2016.

[22] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

[23] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

[24] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. Session-based social recommendation via dynamic graph attention networks. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 555–563, 2019.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[26] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[27] Guangtao Wang, Rex Ying, Jing Huang, and Jure Leskovec. Improving graph attention networks with large margin-based constraints. *arXiv preprint arXiv:1910.11945*, 2019.

[28] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10296–10305, 2019.

[29] B Weisfeiler and A Leman. The reduction of a graph to canonical form and the algebgra which appears therein. *NTI, Series*, 2, 1968.

[30] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019.

[31] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[32] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[33] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[34] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, pages 5453–5462. PMLR, 2018.

[35] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pages 40–48. PMLR, 2016.

[36] Kai Zhang, Yaokang Zhu, Jun Wang, and Jie Zhang. Adaptive structural fingerprints for graph attention networks. In *International Conference on Learning Representations*, 2020.

[37] Shuo Zhang and Lei Xie. Improving attention mechanism in graph neural networks via cardinality preservation. In *IJCAI: proceedings of the conference*, volume 2020, page 1395. NIH Public Access, 2020.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes]
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]