Response to the reviewers

We thank the reviewers for their critical assessment of our work. In the following we address their concerns point by point.

Reviewer 1

Reviewer Point P 1.1 — There are lots of symbols in this paper. Some symbols are reused and confusing, such as s denotes sub-layers or edge features.

Reply: We apologize for the confusing use of symbols. To clarify the symbol usage, we have checked the manuscript and unified the usage of symbol. We avoid the problem of reusing symbols, so that each symbol only represents one meaning after the revision. We summarize the frequently-used symbols in Table 1 in the revised manuscript. We quote the table below:

Category	Symbol	Meaning
Graph Structure	$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	The simple undirected input graph with the vertex set $\mathcal{V}$ and the edge set $\mathcal{E}$.
	v_x	The x-th vertex of the input graph.
	$e_{x,y}$	The edge pointing from v_x to v_y of the input graph.
	$\mathcal{N}(v_x)$	The adjacency set of v_x in the input graph.
	$\bar{d}$	The average degree of the input graph.
GNN Definition	L	The number of GNN layers.
	K	The number of heads in a GNN layer.
	ϕ^l	The messaging function of the GNN layer l.
	Σ^l	The aggregation function of the GNN layer l.
	γ^l	The vertex updating function of the GNN layer l.
	$\phi^{l,i}$ / $\Sigma^{l,i}$ / $\gamma^{l,i}$	The messaging/aggregation/updating function of the i-th sub-layer of the GNN layer l.
	$\boldsymbol{W}^l, \boldsymbol{W}^{(k)}/\boldsymbol{b}, \boldsymbol{a}$	The matrices/vectors represented by the blue characters are the weight matrices/vectors that need to be learned in the GNN.
Vector	$\boldsymbol{v}_x$	The feature vector of the vertex v_x.
	$\boldsymbol{e}_{x,y}$	The feature vector of the edge $e_{x,y}$.
	$\boldsymbol{h}_x^l$	The input hidden vector of the graph neuron corresponding to v_x in the GNN layer l.
	$\boldsymbol{h}_x^{l+1}$	The output hidden vector of the graph neuron corresponding to v_x in the GNN layer l.
	$\boldsymbol{m}_{x,y}^l$	The message vector of the edge $e_{x,y}$ outputted by ϕ^l of the GNN layer l.
	$\boldsymbol{s}_x^l$	The aggregated vector of the vertex v_x outputted by Σ^l of the GNN layer l.
	$\boldsymbol{h}_x^{l,i}$ / $\boldsymbol{m}_{x,y}^{l,i}$ / $\boldsymbol{s}_x^{l,i}$	The hidden/message/aggregated vector of the vertex v_x outputted by $\gamma^{l,i}$/$\phi^{l,i}$/$\Sigma^{l,i}$ of the i-th sub-layer of the GNN layer l.
	d_{in}^l, d_{out}^l	The dimension of the input/output hidden vectors of the GNN layer l.
	$dim(\boldsymbol{x})$	The dimension of a vector $\boldsymbol{x}$.

In the revised manuscript, we use $e_{x,y}$ to represent an edge and use $\boldsymbol{e}_{x,y}$ to represent its input feature vector. The input feature vectors of all edges are same for all GNN layers. We use $\boldsymbol{s}$ to represent aggregated vectors outputted by the aggregation function Σ in graph neurons. For every

vertex v_x, we use $\boldsymbol{s}_x^l$ to denote its aggregated vector in the GNN layer l. If the GNN layer l has sub-layers, $\boldsymbol{s}_x^{l,i}$ represents its aggregated vector in the i-th sub-layer.

To clarify the concept of *sub-layers* in a GNN layer, we have added more description on it in the revised manuscript. We first introduce the concept of sub-layer in Section 2.2 "Graph Neuron and Message-passing Model" as:

> Some complex GNNs like GAT [6] and GaAN [7] use more than one message passing phase in each GNN layer. We regard every message passing phase in a GNN layer as a *sub-layer*. We will give out more details on sub-layers when we introduce GAT.

We then use GAT as an example to elaborate on the concept of sub-layers in Section 2.3 "Representative GNNs" as:

> Each GAT layer consists of a vertex pre-processing phase and two sub-layers (i.e., message-passing phases).
>
> The vertex pre-processing phase calculates the attention vector $\hat{\boldsymbol{h}}_x^l$ for every vertex v_x by $\hat{\boldsymbol{h}}_x = \|_{k=1}^K \boldsymbol{W}_{(k)}^l \boldsymbol{h}_x^l$. We denote the attention sub-vector generated by the k-th head as $\hat{\boldsymbol{h}}_x[k] = \boldsymbol{W}_{(k)}^l \boldsymbol{h}_x^l$.
>
> The first sub-layer of GAT (defined in Equation 1) uses the attention vectors to emit the attention weight vector $\boldsymbol{m}_{y,x}^{l,0}$ for every edge $e_{y,x}$ and aggregates the attention weight vectors for every vertex v_x to get the weight sum vector $\boldsymbol{h}_x^{l,0}$.
>
> $$\begin{aligned} \boldsymbol{m}_{y,x}^{l,0} &= \phi^{l,0}(\boldsymbol{h}_y^l, \boldsymbol{h}_x^l, \boldsymbol{e}_{y,x}, \hat{\boldsymbol{h}}_y, \hat{\boldsymbol{h}}_x) = \|_{k=1}^K \exp(LeakyReLU(\boldsymbol{a}^T[\hat{\boldsymbol{h}}_y[k] \parallel \hat{\boldsymbol{h}}_x[k]])), \\ \boldsymbol{s}_x^{l,0} &= \Sigma_{v_y \in \mathcal{N}(v_x)} \boldsymbol{m}_{y,x}^{l,0}, \\ \boldsymbol{h}_x^{l,0} &= \gamma^{l,0}(\boldsymbol{h}_x^l, \boldsymbol{s}_x^{l,0}) = \boldsymbol{s}_x^{l,0}. \end{aligned} \tag{1}$$
>
> The second sub-layer of GAT (defined in Equation 2) uses the weight sum vectors to normalize the attention weights for every edge and aggregates the attention vectors $\hat{\boldsymbol{h}}_y^l$ with the normalized weights. The aggregated attention vectors $\boldsymbol{s}_x^{l,1}$ are transformed by an activation function δ and are outputted as the hidden vectors of the current layer $\boldsymbol{h}_x^{l+1}$.
>
> $$\begin{aligned} \boldsymbol{m}_{y,x}^{l,1} &= \phi^{l,1}(\boldsymbol{h}_y^{l,0}, \boldsymbol{h}_x^{l,0}, \boldsymbol{e}_{y,x}, \hat{\boldsymbol{h}}_y, \hat{\boldsymbol{h}}_x) = \|_{k=1}^K \frac{\exp(LeakyReLU(\boldsymbol{a}^T[\hat{\boldsymbol{h}}_y[k] \parallel \hat{\boldsymbol{h}}_x[k]]))}{\boldsymbol{h}_x^{l,0}[k]} \hat{\boldsymbol{h}}_y[k], \\ \boldsymbol{s}_x^{l,1} &= \Sigma_{v_y \in \mathcal{N}(v_x)} \boldsymbol{m}_{y,x}^{l,1}, \\ \boldsymbol{h}_x^{l+1} = \boldsymbol{h}_x^{l,1} &= \gamma^{l,1}(\boldsymbol{h}^{l,0}, \boldsymbol{s}_x^{l,1}) = \delta(\boldsymbol{s}_x^{l,1}). \end{aligned} \tag{2}$$

Reviewer Point P 1.2 — Some typical applications of GNNs should be included, such as video object segmentation [ref1], human-object interaction [ref2] and human-parsing [ref3].[1] Zero-shot video object segmentation via attentive graph neural networks,iccv 2019 [2] Learning human-object interactions by graph parsing neural networks, eccv 2018. [3] Hierarchical human parsing with typed part-relation reasoning, cvpr 2020.

Reply: Thank you for pointing out our shortcomings. Computer vision is indeed another important application area of graph neural networks. We have added the mentioned references in the INTRODUCTION section in the revised manuscript. We quote the related sentence below:

> The powerful expression ability makes GNNs achieve good accuracy in not only graph analytical tasks [8, 9, 10] (like node classification and link prediction) but also computer vision tasks (like human-object interaction [11], human parsing [12], and video object segmentation [13]).

Reviewer Point P 1.3 — There are some grammar errors and typos:
- 'Take the demo GNN in Figure 1(a) as the example.'
- 'to calculate the output hidden vector h^{l+1} of the current layer l, i.e., $h^{l+1} = \gamma^l(h^l, s^l)$ The end-to-end training requires. . . '
- 'Implementing it with the specially optimized basic operators on the GPU is a potential optimization'
- The sentences in the experimental section should be unified.

Reply:

Thank you for pointing them out. We have proofread our revised manuscript carefully to eliminate grammar errors and typos. We have also unified the tenses of the sentences in Section 3 "Evaluation Design" and Section 4 "Evaluation Results and Analysis". We use the past tense to describe experimental methods, results and what they indicate. We only use the present tense in the sentences that Figure/Table X are the subjects of the sentences.

Reviewer Point P 1.4 — Figures 6 and 7 should be adjusted. The figures and fonts are too small.

Reply:

We appreciate your comment. We have enlarged Figure 6 and Figure 7 in the revised manuscript. Besides them, we have also enlarged fonts in other figures in the manuscript, to make sure that font sizes in figures are no less than the font size of figure captions.

Reviewer Point P 1.5 — In my view, computation efficiency is to describe the testing or validation process. Except for reporting and analyzing the training times, it is meaningful to discuss the inference time. This is also an important point of view for deep learning researchers to be concerned about.

Reply:

Reviewer 2

Reviewer Point P 2.1 — This is the first point of Reviewer 2. With some more words foo bar foo bar ...

Reply: Our reply to it with reference to one of our points above using the LaTeX's