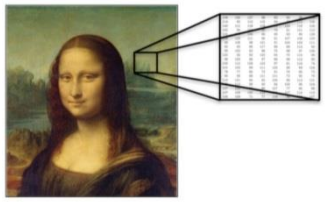
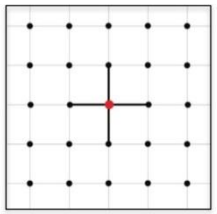
Graph U-Net

Anonymous authors
Paper under double-blind review
(Under review as a conference paper at ICLR 2019)

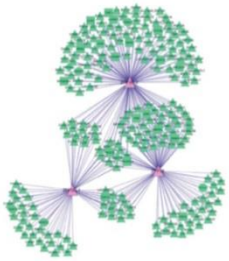
Image and Graph





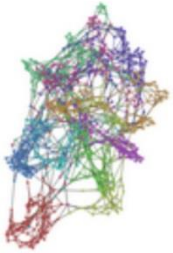
2D grids

Social networks

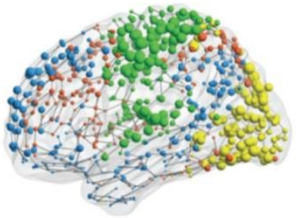


Regulatory networks

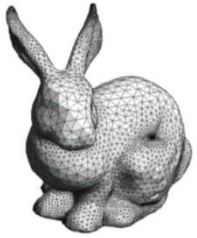
=



Graphs/
Networks



Functional networks



3D shapes

- Images have grid-like structures. Elements on feature maps have locality and order information, which enables the application of convolutional operations and pooling operations.

Summary

- Encoder-decoder architectures like the U-Net are state-of-the-art methods for pixel-wise prediction tasks. This work is to develop U-Net-like architectures for graph data.
- In U-Net, pooling and up-sampling operations are essential building blocks in these architectures, which is challenging to implement on graph data. The authors proposed graph pooling (gPool) and unpooling (gUnpool) operations.

GCN

- The layer-wise forward-propagation operation of GCNs is defined as,

$$X_{\ell+1} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}X_{\ell}W_{\ell})$$

  - X_l is the feature matrix of layer *l*
  - $\hat{A}$ = *A + I* is used to add self-loops in the input adjacency matrix *A*
  - $\hat{D}$ diagonal node degree matrix for normalization.
  - W_l is a trainable weight matrix that applies a linear transformation to feature vectors

GRAPH U-NET ARCHITECTURE

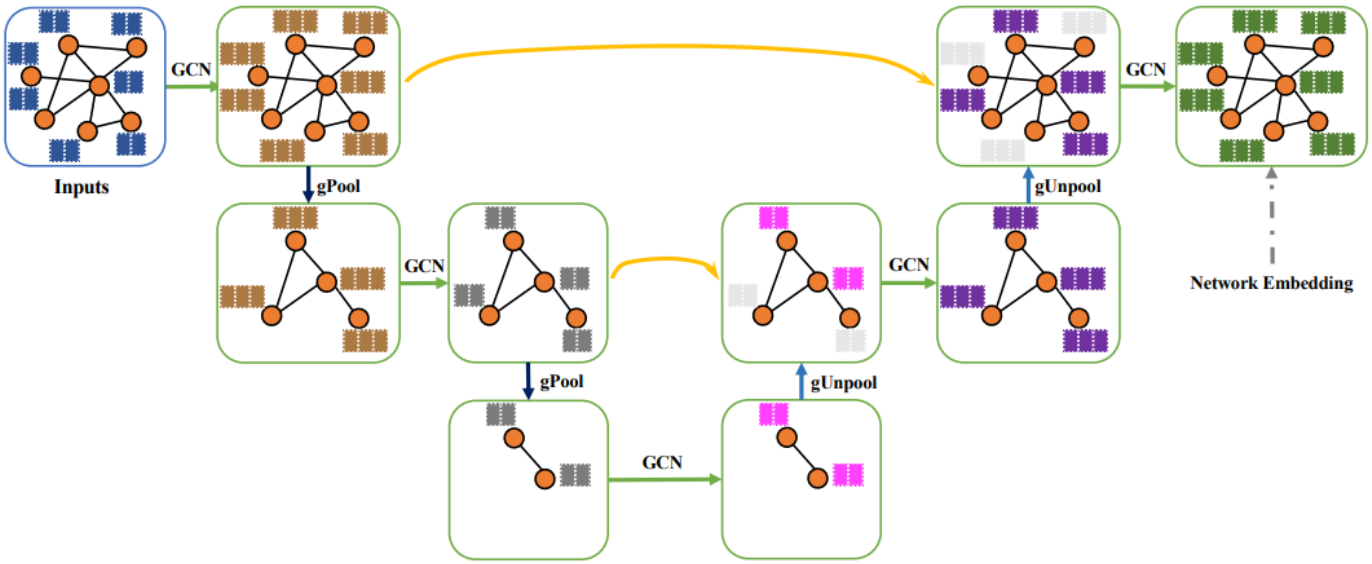


Figure 3: An illustration of the proposed graph U-Net (g-U-Net). In this example, each node in the input graph has two features. The input feature vectors are transformed into low-dimensional representations using a GCN layer. After that, we stack two encoder blocks, each of which contains a gPool layer and a GCN layer. In the decoder part, there are also two decoder blocks. Each block consists of a gUnpool layer and a GCN layer. For blocks in the same level, encoder block uses skip connection to fuse the low-level spatial features from the encoder block. The output feature vectors of nodes in the last layer are network embedding, which can be used for various tasks such as node classification and link prediction.

GRAPH POOLING LAYER

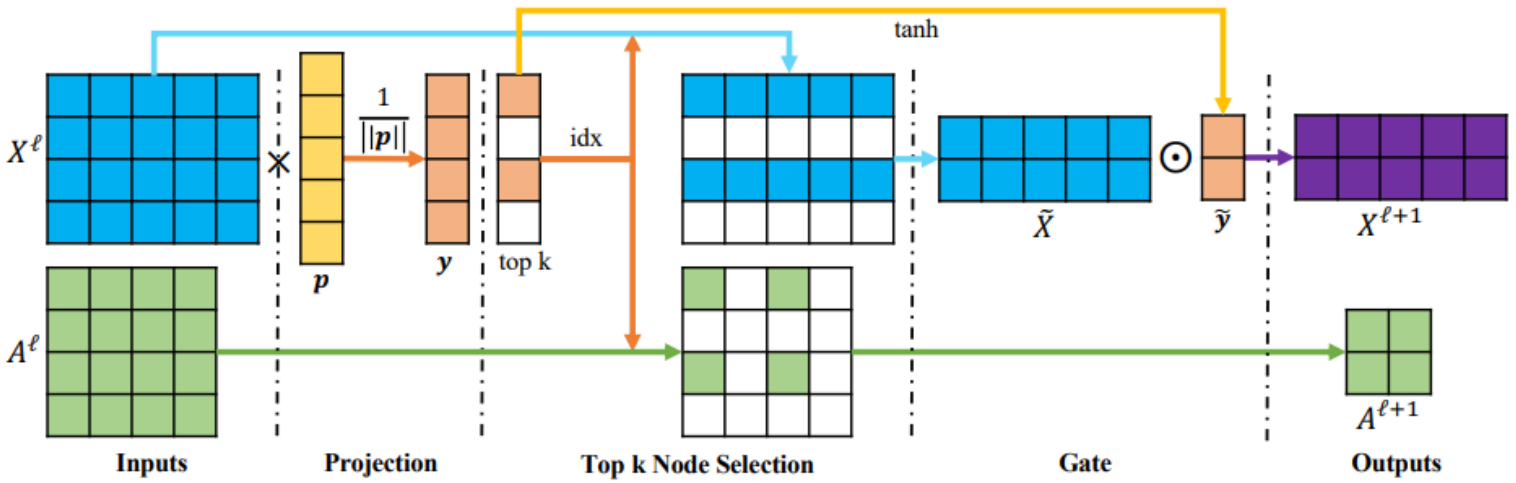
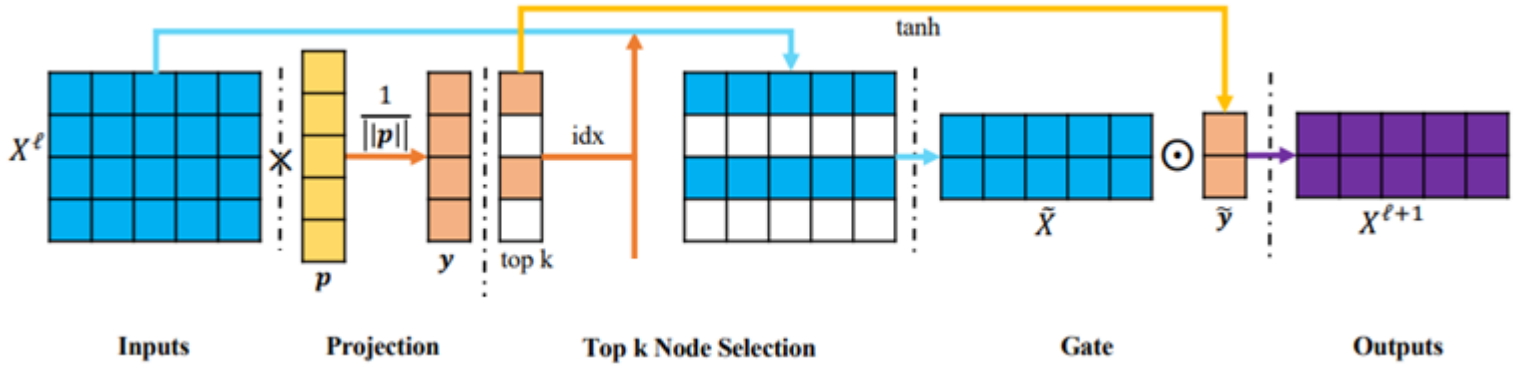


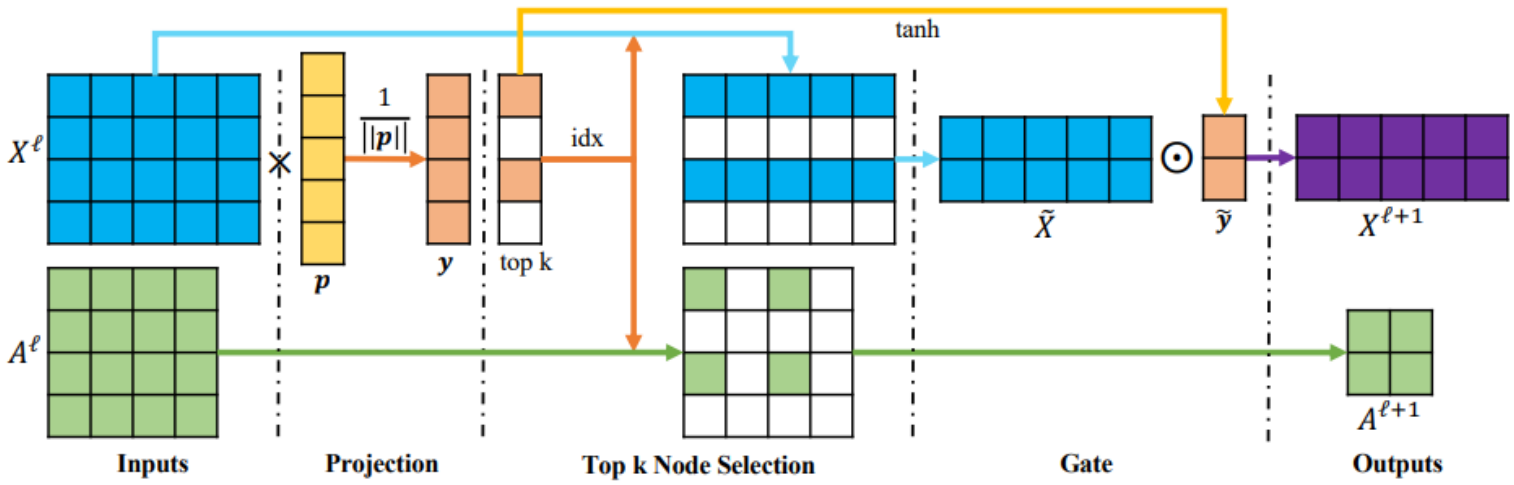
Figure 1: An illustration of the proposed graph pooling layer with $k = 2$. $\times$ and $\odot$ denote matrix multiplication and element-wise product, respectively. We consider a graph with 4 nodes, and each node has 5 features. By processing this graph, we obtain the adjacency matrix $A^\ell \in \mathbb{R}^{4\times4}$ and the input feature matrix $X^\ell \in \mathbb{R}^{4\times5}$ of layer ℓ. In the projection stage, $\mathbf{p} \in \mathbb{R}^5$ is a trainable projection vector. By matrix multiplication and $\tanh(\cdot)$, we obtain $\mathbf{y}$ that are scores estimating scalar projection values of each node to the projection vector. By using $k = 2$, we select two nodes with the highest scores and record their indices in the top-k-node selection stage. We use the indices to extract the corresponding nodes to form a new graph, resulting in the pooled feature map $\tilde{X}^\ell$ and new corresponding adjacency matrix $A^{\ell+1}$. At the gate stage, we perform element-wise multiplication between $\tilde{X}^\ell$ and the selected node scores vector $\tilde{\mathbf{y}}$, resulting in $X^{\ell+1}$. This graph pooling layer outputs $A^{\ell+1}$ and $X^{\ell+1}$.

GRAPH POOLING LAYER



1. Projection: $\mathbf{y} = X^\ell \mathbf{p}^\ell / \|\mathbf{p}^\ell\|$, project all node features to 1D
2. Top k Node selection: $\text{idx} = \text{rank}(\mathbf{y}, k), \quad \tilde{X}^\ell = X^\ell(\text{idx}, :)$ reduce feature map dimension
3. Gate: $\tilde{\mathbf{y}} = \tanh(\mathbf{y}(\text{idx})), \quad X^{\ell+1} = \tilde{X}^\ell \odot (\tilde{\mathbf{y}} \mathbf{1}_C^T)$, control information flow and make **p** trainable

GRAPH POOLING LAYER



- Top k Node Selection: $A^{\ell+1} = A^{\ell}(\text{idx}, \text{idx}),$
- *Graph connectivity augmentation: to aggregate information from its first-order neighboring nodes*

$$A^2 = A^\ell A^\ell, \quad A^{\ell+1} = A^2(\text{idx}, \text{idx})$$

GRAPH U-NET ARCHITECTURE

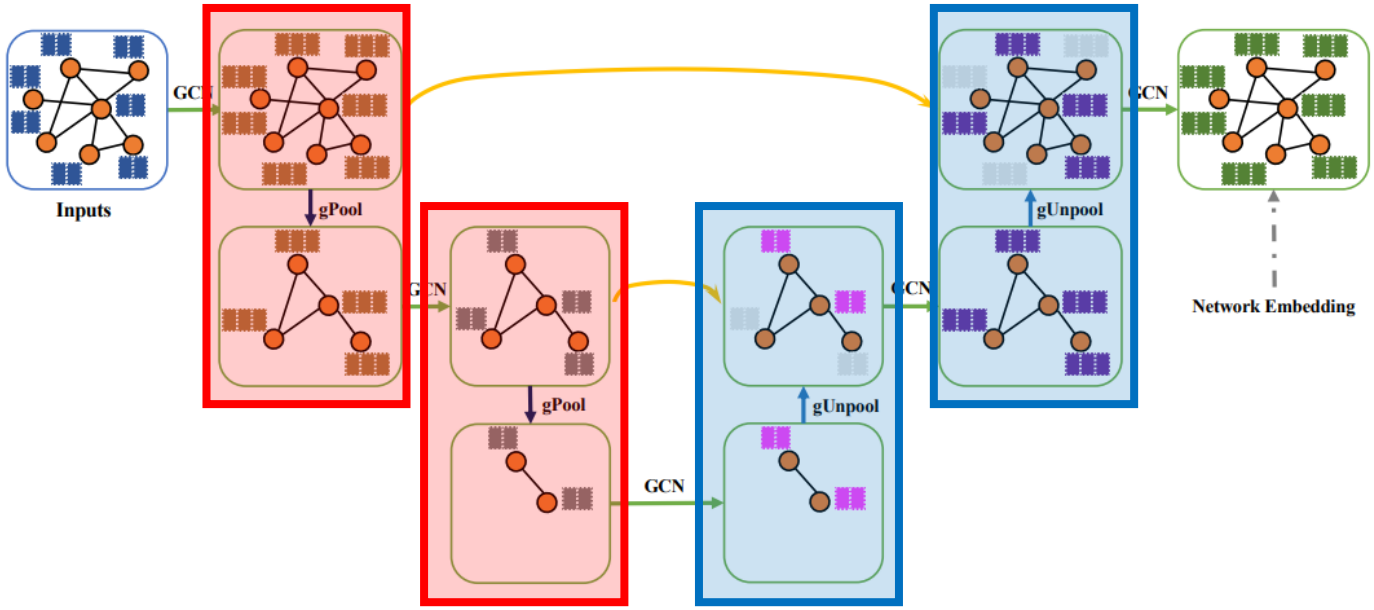
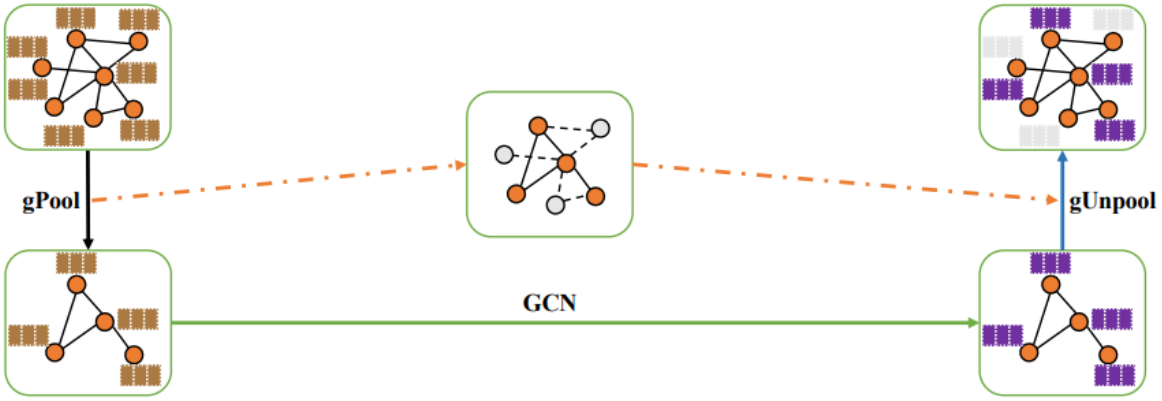


Figure 3: An illustration of the proposed graph U-Net (g-U-Net). In this example, each node in the input graph has two features. The input feature vectors are transformed into low-dimensional representations using a GCN layer. After that, we stack two encoder blocks, each of which contains a gPool layer and a GCN layer. In the decoder part, there are also two decoder blocks. Each block consists of a gUnpool layer and a GCN layer. For blocks in the same level, encoder block uses skip connection to fuse the low-level spatial features from the encoder block. The output feature vectors of nodes in the last layer are network embedding, which can be used for various tasks such as node classification and link prediction.

GRAPH UNPOOLING LAYER



$$X^{\ell+1} = \text{distribute}(0_{N\times C}, X^{\ell}, \text{idx})$$

- In $X^{\ell+1}$, row vectors with indices in idx are updated by row vectors in X^{ℓ} , while other row vectors remain zero.

GRAPH U-NET ARCHITECTURE

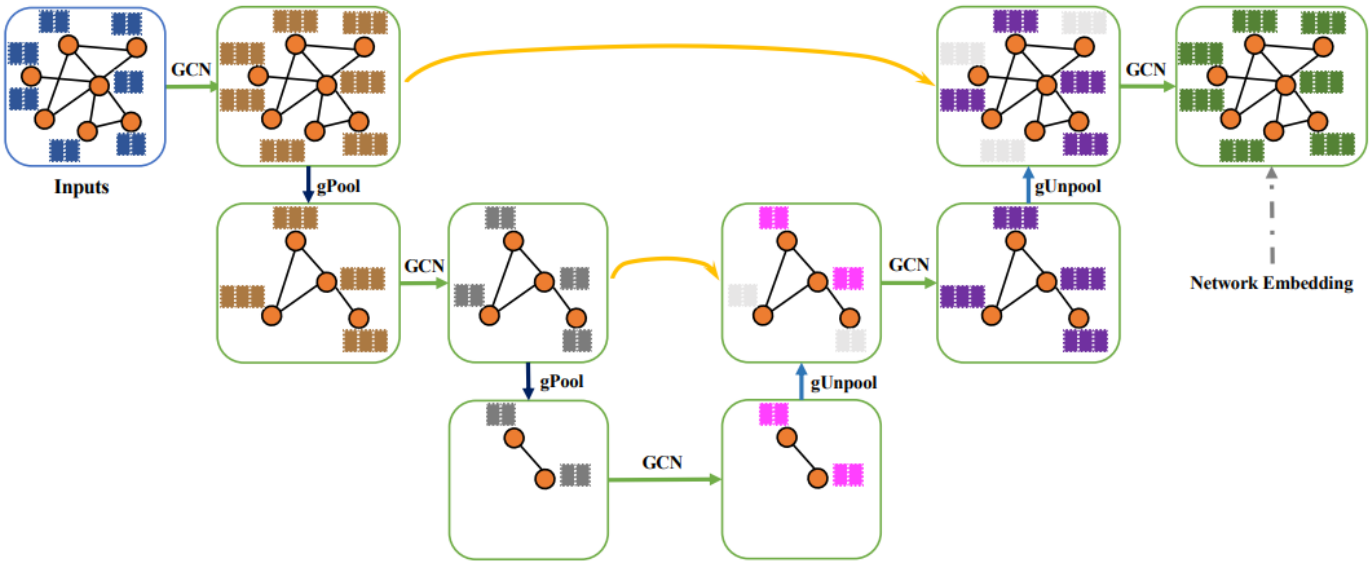


Figure 3: An illustration of the proposed graph U-Net (g-U-Net). In this example, each node in the input graph has two features. The input feature vectors are transformed into low-dimensional representations using a GCN layer. After that, we stack two encoder blocks, each of which contains a gPool layer and a GCN layer. In the decoder part, there are also two decoder blocks. Each block consists of a gUnpool layer and a GCN layer. For blocks in the same level, encoder block uses skip connection to fuse the low-level spatial features from the encoder block. The output feature vectors of nodes in the last layer are network embedding, which can be used for various tasks such as node classification and link prediction.

Results

Table 2: Results of transductive learning experiments in terms of node classification accuracies on Cora, Citeseer, and Pubmed datasets. g-U-Net denotes our proposed graph U-Net model.

Models	**Cora**	**Citeseer**	**Pubmed**
DeepWalk (Perozzi et al., 2014)	67.2%	43.2%	65.3%
Planetoid (Yang et al., 2016)	75.7%	64.7%	77.2%
Chebyshev (Defferrard et al., 2016)	81.2%	69.8%	74.4%
GCN (Kipf & Welling, 2017)	81.5%	70.3%	79.0%
GAT (Veličković et al., 2017)	$83.0 \pm 0.7\%$	$72.5 \pm 0.7\%$	$79.0 \pm 0.3\%$
g-U-Net (Ours)	$\mathbf{84.4 \pm 0.6\%}$	$\mathbf{73.2 \pm 0.5\%}$	$\mathbf{79.6 \pm 0.2\%}$

Table 3: Comparison of g-U-Nets with and without gPool or gUnpool layers in terms of node classification accuracy on Cora, Citeseer, and Pubmed datasets.

Models	**Cora**	**Citeseer**	**Pubmed**
g-U-Net without gPool or gUnpool	$82.1 \pm 0.6\%$	$71.6 \pm 0.5\%$	$79.1 \pm 0.2\%$
g-U-Net (Ours)	$\mathbf{84.4 \pm 0.6\%}$	$\mathbf{73.2 \pm 0.5\%}$	$\mathbf{79.6 \pm 0.2\%}$

Results

Table 4: Comparison of g-U-Nets with and without graph connectivity augmentation in terms of node classification accuracy on Cora, Citeseer, and Pubmed datasets.

Models	**Cora**	**Citeseer**	**Pubmed**
g-U-Net without augmentation	$83.7 \pm 0.7\%$	$72.5 \pm 0.6\%$	$79.0 \pm 0.3\%$
g-U-Net (Ours)	$\mathbf{84.4 \pm 0.6\%}$	$\mathbf{73.2 \pm 0.5\%}$	$\mathbf{79.6 \pm 0.2\%}$

Table 5: Comparison of different network depths in terms of node classification accuracy on Cora, Citeseer, and Pubmed datasets. Based on g-U-Net, we try different network depths in terms of the number of blocks in encoder and decoder parts.

Depth	**Cora**	**Citeseer**	**Pubmed**
2	$82.6 \pm 0.6\%$	$71.8 \pm 0.5\%$	$79.1 \pm 0.3\%$
3	$83.8 \pm 0.7\%$	$72.7 \pm 0.7\%$	$79.4 \pm 0.4\%$
4	$\mathbf{84.4 \pm 0.6\%}$	$\mathbf{73.2 \pm 0.5\%}$	$\mathbf{79.6 \pm 0.2\%}$
5	$84.1 \pm 0.5\%$	$72.8 \pm 0.6\%$	$79.5 \pm 0.3\%$

Conclusions

- The authors proposed gPool and gUnpool layers in g-U-Net networks for network embedding, which samples a subset of important nodes to enable high-level feature encoding and receptive field enlargement.
- Based on gPool and gUnpool layers, the authors proposed the graph U-Net (g-U-Net) architecture which uses a similar encoder-decoder architecture as regular U-Net on image data.
- Experimental results demonstrate that our g-U-Net achieves performance improvements as compared to other GNNs on transductive learning tasks