In the name of God, the compassionate, the merciful

**Abstract**

Graph embedding learning is a fundamental and crucial task while dealing with a wide range of data. The encoder-decoder architecture like Unet has been successfully applied on many image pixel-wised tasks. Unet model is a deep neural network whose capability has seen it find widespread use in image segmentation tasks. However, it is limited in this regard by seeking non-graph structured data. Although some pooling and upsampling operation was introduced to adjust this architecture for graph data, they connive the importance of structural information. Existing methods for Graph pooling predominantly rely on down sampling just based on node embedding, without utilizing graph topology.

We present a novel model called GIUNet for the representation learning on graph data. This proposed graph unets structure is based on graph isomorphic convolution while using a comprehensive pqPooling layer. We also apply link prediction at each level to achieve more expressive node embedding to resolve dangling node problem. Ablation studies indicates that utilizing GIUnet model will make a remarkable result in compare to state-of-the-art work on several benchmark datasets.

# 1 Introduction

­­­­Graphs are pervasive data structures that model the pair-wise interactions between entities. Learning with graphs encounters unique challenges, including their combinatorial nature and the scalability bottleneck, compared with Euclidean data (e.g., images, videos, speech signals, and natural languages). With the remarkable success of deep learning for the latter data types, recent years have witnessed increasing interest in generalizing in neural network for graph-structured data [graph convolutional network with eigenpooling]. Convolutional neural networks (CNNs) (LeCun et al., 2012) have demonstrated great capability in various challenging artificial intelligence tasks, especially in fields of computer vision (He et al., 2017; Huang et al., 2017) and natural language processing (Bahdanau et al., 2015). Graph convolutional networks (GCNs) (Kipf & Welling,2017) generalize convolutional neural networks (CNNs) (Le-Cun et al., 1995) on graph-structured data, To learn the graph representations. The “graph convolution” operation applies the same linear transformation to all the neighbors of a node followed by a nonlinear activation function. The stream of research on this topic is usually under the name of “Graph Neural Networks” [the graph neural network model] which typically involves transforming, propagating and aggregating node features across the graph. Among them, most researches focuses on node-level representation learning [18,22,35]. While others investigate learning graph-level representation. Each representation stand from different perspective; methods focusing on node representation learning have brought improvement to tasks such as node classification [ 14 –16 , 18 , 22, 35 ] and link prediction [ 35 ] and those methods working on graph-level representation learning, have mainly facilitated graph classification.

Images can be considered as special cases of graphs, in which nodes lie on regular 2D lattices. This special structure enables the use of convolution and pooling operations on images. Based on this relationship, node classification and embedding tasks have a natural correspondence with pixel-wise prediction tasks such as image segmentation (Noh et al., 2015; Gao & Ji, 2017; J ́egou et al., 2017). Encoder-decoder architectures like the U-Net (Ronneberger et al., 2015) are state-of-the-art methods for these tasks. It is thus highly interesting to develop U-Net-like architectures for graph data. In addition to convolutions, pooling and up-sampling operations are essential building blocks in these architectures. Graph-Unet for first time propose these blocks to empower the capacity of Unet architecture on graph structured data [g-u-net]. The mentioned paper was a pioneer in this field of study, but it neglects some significant aspects. This model does not use topological information for choosing influential nodes and results in missing graph-level representation. Furthermore, in the Graph-Unet model, using simple GCN layer in each level, leads to inexpressive embedding cause of not learning to distinguish certain simple graph structures [GIN].

To cover these aspects, we proposed GIUnet structure which introduces novel general pooling (pqPool), promising basis unpooling (pastUnpool) and beneficiaries from Graph isomorphic convolution which is the most expressive convolution among all class of GNNs. Experimental outcomes on graph classification task, demonstrates outstanding result in compare to state-of-the-art models.

# 2 Related Work

# 3 Proposed Method

Our proposed Graph Isomorphic Unet architecture consists of three components: pqPool layer and pastUnpool layer, Edge Augmentation Layer and GIN layer.

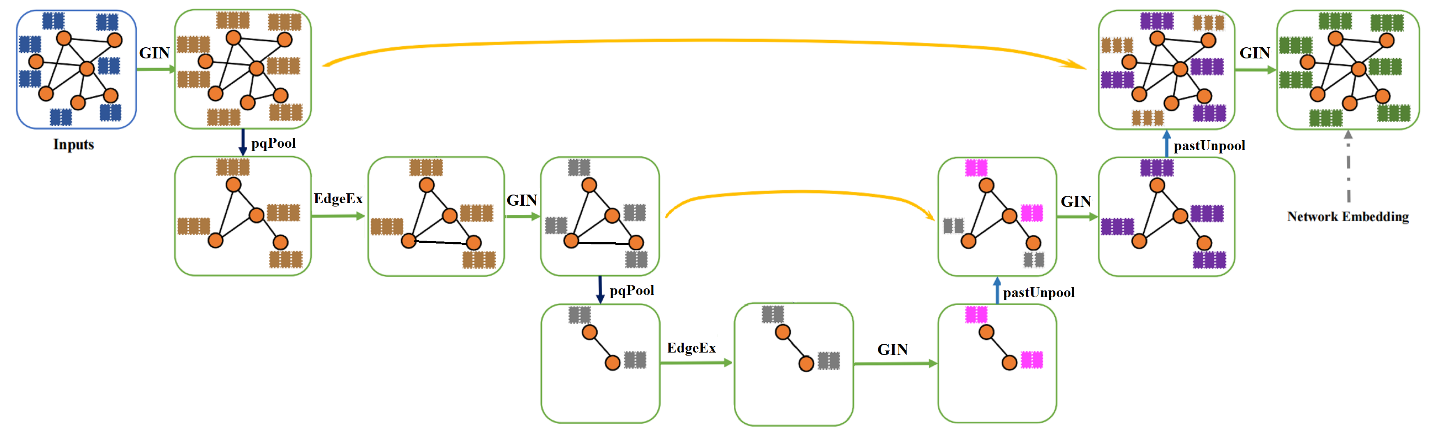
At the first of this section, we will explain the Graph Unet structure and its vantage among different architecture. As follow, we will clarify the proposed “pqPool” layer and “pastUnpool” layer and later on, to settle “Dangle node problem” edge expansion, “edgeEx” layer is set up and at last Graph isomorphism network, “GIN” layer as convolution layer is demonstrated. Later on this chapter, we will deep into detail of each mentioned components.

## 3-1 Graph Unet architecture

The encoder-decoder structure shows significant result on pixel-based classification. The Unet precedes in prediction field since, its capability in encoding high-level features while preserving local spatial properties. Similar to pixel-wise prediction task [gong 2014], graph classification tasks aim to make a prediction for each super node. With using our pqPool and pastUnpool layer, the Innovative Graph Unet architecture will be proposed.

This architecture contains of two major part; Encoder and Decoder. Encoder aims to convert nodes into low-dimensional expressive embedding. The encoder is built by several stacked encoding block, each of which contains a pqPool layer followed by EdgeEx layer and a GIN layer. pqPool layer’s role is to contract graph size so it can encode higher order feature, meanwhile EdgeEx try to face sparsity of adjacency matrix problem, after pooling operation. In this block, GIN layer is responsible for aggregating information of first order neighbors.

In decoder part an equal number of stacked decoding block is utilized. Each decoder block is composed of pastUnpool layer and a GIN layer; the pastUnpool layer restores the graph into its higher resolution structure. There are skip-connection between corresponding blocks of encoder and decoder blocks at each level, which transmit spatial information to decoders and aid to initialize of unpooled nodes for better representation. Figure 1 provides an illustration of a sample Graph Unet with two encoder and decoder blocks.



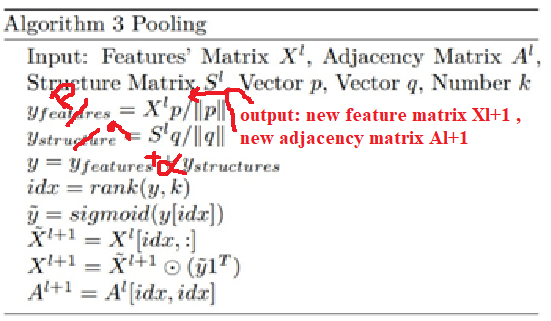
## 3-2 pqPool Layer

Pooling layer is one of crucial building block of a deep neural networks. The pooling layer executes the down-sampling on the feature map coming from the previous layer, produces an embedding with a condense resolution and enlarges receptive fields. This layer drastically diminishes the spatial dimension of input and rises to better generalization and performance (yu & kultun, 2016).

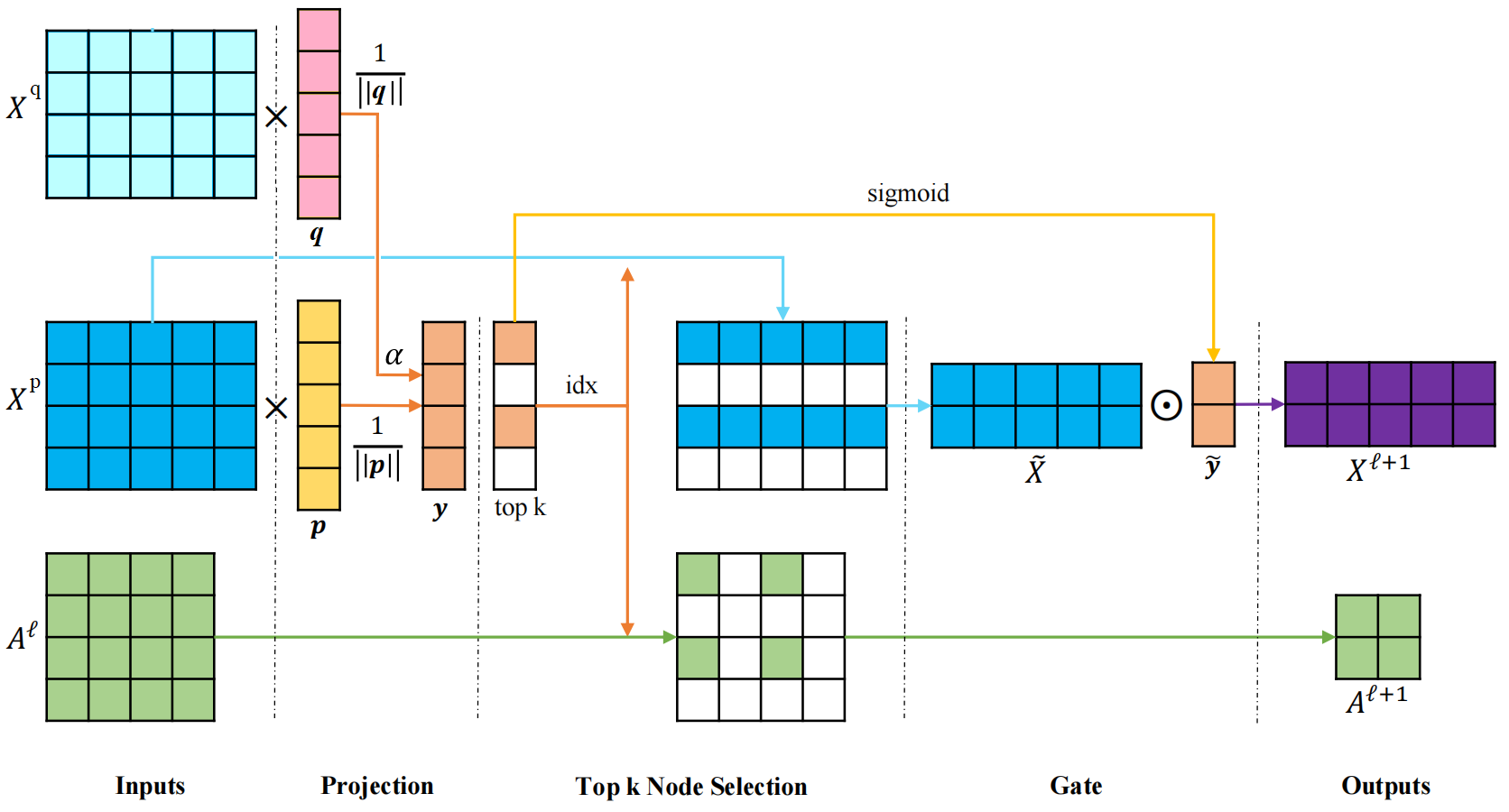
Applying Pooling operation in Graph-structure is in company with challenges due to lack of spatial order. To resolve this issue, node embbedings and graph topology could be exerted. Former work concentrated on the canonical idea of finding top k important nodes just based on their embbedings and does not consider the structure of graph, so the topological information is discarded. Our proposed layer simultaneously considers both node features and graph patterns. To aim this goal, pqPooling layer is proposed. This method composed of two part; P part learns node importance from embeddings while Q part tries to capture structural information. For P part we employ a trainable projection vector P by projecting all node features. We can perform k-max pooling based on summation of P and Q part. It is obvious that the selection is based on 1D footprint of each node, so the connectivity in the new graph is consistent across nodes.

For computation in P part, given a node i with its feature vector xi, the scalar projection of xi on p is ypi = xip/‖p‖. Also in Q part, we use similar approach to involve graph topology. Here – with using two proposed pooling- we got a structural feature matrix. In detail, given a node i with structural feature vector , the scaler projection of it on q is as same as p formula : yqi = xiq/‖q‖. To preserve as much information as possible from the original graph, we select nodes with the largest scalar projection values on y which is computed from weighted summation of ypi and yqi to down sample the graph. Also for adjusting feature vector based on their importance for next layer, we multiply , sigmoid of y, into embedding of chosen node. All of these operation could be written algorithm 1:

[اصلاح نامگذاری Xq  و Xp]



where k is the number of nodes selected in the new graph. rank(y, k) is the operation of node ranking, which returns indices of the k-largest values in y. The idx returned by rank(y, k) contains the indices of nodes selected for the new graph. A(idx, idx) and X(idx, :) perform the row and/or column extraction to form the adjacency matrix and the feature matrix for the new graph. y(idx) extracts values in y with indices idx followed by a sigmoid operation. 1C ∈ RC is a vector of size C with all components being 1, and represents the element-wise matrix multiplication. Figure 2 show the structure of proposed pooling:



As mentioned above, to prepare structural feature matrix we propose two approaches; first one is based on scores obtained from different centralities and in second perspective, we utilize Eigen vectors to exploit graph structure.

### 3-2-1 pqCentralityPool

Throughout this section to achieve the structural features, we introduce a new feature extraction based on different centrality measure to endow various aspect of topological information.

To deep in details, for each node we represent a feature vector based on 6 distinctive centrality score including, closeness, degree, subgraph, harmonic, adamic-adar and second order centrality metrics.

After computing feature matrix Xq, the linear combination of Xq in a learnable vector Q, which decide the precedence of metrics, will result to structural score for each node.

### 3-2-2 pqSpectralPool

In order enable down-sampling on graph data, many studies have been released. Recently, graph signal processing has been extensively incorporated into this area because of notable improving [Graph Signal Processing: Overview, Challenges, and Applications]. Based on this approach, lots of impressive method for graph sampling, have been proposed. Utilizing Spectral clustering get stunning result in this field. [active semi supervised learning… ]

In this paper [active semi supervised learning… ], authors proposed a method to get the smoothest signal which contains cardinal information without noisy behaviors. To reach this, they choose nodes with most energy as sample from whole graph in eigen vector correspond to least eigen value. In a situation, where the residual of eigen values are too small, choosing just one dimension is not enough representative and leads to lots of information loss. One rational idea to prevent this loss and utilize as most information as possible, is to use m-least eigen dimension to choose most influential nodes. To deploy this idea, first we compute singular value decomposition on Laplacian matrix computed from adjacency matrix. Then we choose eigen vectors correspond to m less absolute eigen values. We use resulted matrix as Xq in our proposed algorithm.

## 3-3 PastUnpool Layer

Upsampling operations are as substantial as pooling in Encoder-Decoder architecture such as Unet. While in Encoder block, pooling constructs an abstract vision, in Decoder, pastUnpool endeavors to restitute the original resolutions.

PastUnpool layer performs the inverse operation of pqPool layer and the restore the graph primary structure. To attain this objective, we maintain the original location of nodes which was eliminated in corresponding pqPool layer and use this information to revert nodes to its position. Formally, we proposed the layer-wise propagation rule of graph unpooling layer as

Where contains indices of eliminated nodes in the corresponding pqPool layer that reduces the graph size from N nodes to k nodes. are the feature matrix of the current graph and are the initially feature matrix for the new graph. In contrast to g-U-Nets, this initialization will lead to more expressive embedding with less effort, since it begins from better start point representation.

## 3-4 EdgeEx Layer

In our proposed pqPool layer we choose some central nodes to contract the initial graph. As result of graph down sampling, some edges will be eliminated an so some dangling node will be isolated and fallen. To prevent this scenario, we suggest to consolid information flow with edge augmentation using link prediction methods. This operation could be done in 2 different order; first we can predict edges as preprocessing on initial graph, but in the case of large graph it would face high time complexity problem, second way to implement this idea is to add a link prediction block for each batch after pooling layer. Although applying this method on each batch reduces the computation cost, but it limits our option for link augmentation algorithms due to complete structure necessity and as result, lots of link prediction methods are not effective. To improve structural information flow as much as possible in each batch of graph, we employ Adamic-Adar method to maintain and amplify neighborhood.

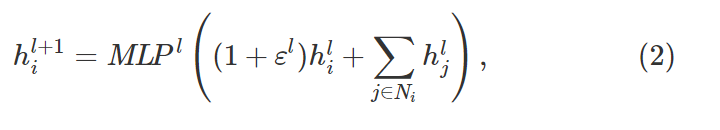
## 3-5 GIN Layer

Graph neural networks (GNNs) have enjoyed tremendous success when applying to graph-structured data. GNN is a special neural network architecture with the equivalent functionality as CNN in deep learning models; it aims to extract the representation of each node or graph for various task in Machine Learning field.

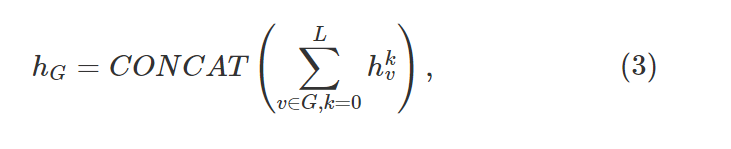
The GNNs leverage the graph structure and iteratively update the node representation from the node’s neighborhood in a fashion of convolutional operation to obtain the final feature embedding of the nodes or the graphs. To explore the deeper and more extensive information of the node receiving domain multiple graphical convolution layer are usually stacked together to achieve more abstract vision. [Enhanced Graph Isomorphism Network for

Molecular ADMET Properties Prediction]

Graph Isomorphism Network (GIN) is one of the most potential GNN variations, and its discriminative/representational power is equal to the power of the Weisfeiler-Lehman (WL) graph isomorphism test. GIN replaces fundamental mean aggregation function of GCN with the sum aggregator as following formula:



where  is a learnable parameter, MLP is a Multi-layer Perceptron. The (2) shows that in GIN, each neighbor contributes equally to the update of the central node. Furthermore, GIN concatenates the information of the nodes’ representation across all layers of the model for the final representation according to the following formula:



Where v, G are the node and graph, respectively. CONCAT(⋅) denotes concatenate function.

It has been demonstrated theoretically and experimentally that GIN has more discriminative or representational power of graph structures than previous GNN models [GIN].

# 5 Experiment

# Conclusion

In this work we introduce a novel pqPool and pastUnpool layers in GIUnet structure for node embedding which covers structural and featural aspects. The pqPool layer implements both local and global k-max pooling operation on graph data and leads to extract more abstract feature and improve general overview of graph. By learning two trainable projection vector p and q, we apperceive the importance of each dimension in both node representation space and structural space. After that based on weighted summation of scalar projection values the supremacy of nodes is determined. Proposed pastUnpool layer applies the reverse stream and avails the topology of graph in each level to recover structure and improves it, by using past correspond node embedding as initialization to speed up convergence. By employing the EdgeEx layer we dissolve dangling node problem with link prediction techniques. Experimental results approved that, Applying GIN in graph-unet structure will leads to superior performance to state-of-the-art results.