Exploration of GraphGPS For Graph Representation Learning

Prabina Pokharel
p2pokharel@ucsd.edu

Gal Mishne
gmishne@ucsd.edu

Yusu Wang
yusuwang@ucsd.edu

Abstract

Graph-based architectures have been popularized in recent years because of the surge in graph-based data. Current architectures do not perform well on graph data with large amounts of layers, referred to as long-range interaction. They face issues like oversmoothing and oversquashing, which leads to model learning only short-ranged signals from training data and not generalizing well on test data. A model needs to generalize to long-range interactions so that we can use information from neighboring nodes to classify a given node. In this paper, we will conduct a comparative analysis of baseline Graph Neural Networks (GNNs) such as Graph Convolutional Networks (GCN), Graph Isomorphism Networks (GIN), and Graph Attention Networks (GAT) against the transformer-based model GraphGPS. We will evaluate these models across Cora, IMDB, and Enzymes datasets; and PascalVOC-SP dataset from the Long Range Graph Benchmark. Our research aims to analyze the accuracy of these models to provide insights into their performance ability and discuss whether they overcome the oversmoothing and oversquashing issues.

Code: https://github.com/prabina-p/GNN-Long-Range-Interactions

1 Introduction

1.1 Ubiquity of Graph Data: Real-World Context

Many real-world data are commonly formatted in graph format. For example, many social apps, like Instagram, have a social network where nodes are people, edges are relationships (mother, friend, co-worker, etc.), and attributes are information pertinent to the node (height, age, etc.). In the current age where graph data is ubiquitous, there is a growing concern that current graph-based architectures fail to capture the complexity present in those data.

1.2 Complexity and Challenges in Graph Structures

This distinction in performance becomes more apparent when we face extensive and intricate layers of information in these graphs. Our graph data may include diverse attributes such as one layer for user characteristics, another for hobbies, and another for geographical location, among others. These multi-layered structures introduce the notion of long-range interaction, where preserving these connections between layers becomes an important goal, so we can leverage it for machine-learning tasks like node and graph classification.

Given these intricacies, current state-of-the-art architectures face significant challenges with graph data that has long-range interactions. Issues such as oversmoothing and oversquashing arise, leading to ineffective learning.

1.2.1 Oversmoothing and its Implications

Oversmoothing (Oono and Suzuki 2020) occurs when the message-passing formalism (i.e. framework used to analyze interactions between nodes), softens out the distances between neighboring nodes excessively. Consequently, node embeddings become too similar, which leads to hindering classification in later phases. Even nodes that do not experience oversmoothing in the first few layers may suffer from this as they traverse deeper into the layers, impacting the model's performance.

1.2.2 Oversquashing and its Implications

Similarly, oversquashing (Alon and Yahav 2021) occurs when nodes receive massive, exponential amounts of data, causing them to forget the information from distant nodes. This phenomenon is visualized in Figure 1 (Alon and Yahav 2021), where GNN bottleneck occurs due to the exponentially-growing information overload. Oversquashing becomes more prevalent in long-range interactions, causing crucial information from far away to get "squashed" or slowly forgotten. Consequently, the model leads to learning only short-ranged signals from training data and not generalizing well on test data.

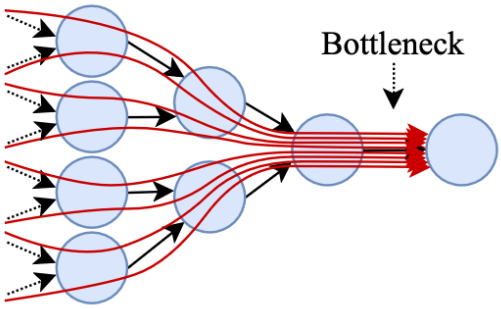

Figure 1: Oversquashing in GNNs (Alon and Yahav 2021)

1.3 Exploration of Traditional Graph Neural Networks (GNNs)

In light of these challenges, our research will investigate the performance of traditional graph neural networks (GNNs) - Graph Convolutional Networks (GCN), Graph Isomorphism Networks (GIN), and Graph Attention Networks (GAT).

1.3.1 GCN, GIN, GAT: Foundation and Approaches

GCN (Kipf and Welling 2017) uses the message-passing formalism described earlier, where it updates nodes based on the aggregation of its neighboring nodes. GATs (Veličković et al. 2018), on the other hand, aim at addressing oversquashing by leveraging masked self-attentional layers, allowing for adaptive weighting of neighboring nodes. This strategy has shown promise in improving performance on long-range interactions as it receives selective information from neighboring nodes. Additionally, GINs (Xu et al. 2019) add non-linearity after iteratively aggregating node features. These three GNN models (GCN, GAT, and GIN) serve as baseline models since they set a foundational approach for learning graph representations.

1.4 Introduction to GraphGPS: Transformer-based Model

In this study, we will compare the effectiveness of these traditional GNNs alongside a transformer-based model, GraphGPS. GraphGPS (Rampášek et al. 2022) presents a novel approach by leveraging self-attention mechanisms and transformer architectures to capture long-range interactions. We will assess GraphGPS' capabilities in contrast to traditional GNNs for machine learning tasks.

1.5 Machine Learning Tasks and Data Exploration

For our machine learning tasks, we'll perform node classification on Cora and PascalVOC-SP datasets, along with graph classification on IMDB and Enzymes datasets. The details

on these datasets can be found in Section 3: Results. While Cora, IMDB, and Enzymes are conventional datasets, PascalVOC-SP is part of the Long Range Graph Benchmark datasets, designed to assess models' capabilities in handling long-range interactions. Each dataset presents distinct challenges with its varying sizes and structures, and that combined with the diverse mix of machine learning tasks like node and graph classification and dataset types like regular and long-range will provide us with an ideal scenario to evaluate our models. By exploring these instances, we will assess whether our models can overcome the oversmoothing and oversquashing issues, while we shed light on their strengths and limitations.

2 Methods

In order to understand baseline models in greater detail, we will explore its most popular architecture: GCN. We will also compare it to the transformer mode GraphGPS. For both, we will speak on its strength and limitations.

2.1 GCN

GCN utilizes the message-passing framework to update node representations based on the information gathered from its adjacent nodes. Its framework includes these components:

- **Node features:** Each node has a feature vector that includes attributes (height, age).
- **Input layer:** The input nodes each contain feature vectors.
- **Convolution:** The convolutional layers perform message passing on the graph's adjacency matrix and the node features. In these convolutions, information from neighboring nodes gets incorporated via aggregate functions such as mean or sum. By using aggregate functions, each node is able to capture information from all its neighboring nodes.
- **Hidden layers:** There can be multiple GCN layers to capture deeper and more abstract representations of nodes. Each subsequent layer's input will be the output of its previous layer.
- **Activation Functions**: GCN can have activation functions after each hidden layers. It's task is to introduce non-linearities to the network, allowing the model to learn complexities. Common activation functions include:
  - **Rectified Linear Unit (ReLU):** This function is most popular among activation functions. For any input x, ReLU returns x if $x > 0$ and 0 otherwise. Mathematically, it's represented as:

$$f(x) = \max(0, x)$$

  - **Sigmoid:** This function is commonly used for binary classification, where the output can be viewed as a probability. It squashes values between 0 and 1.

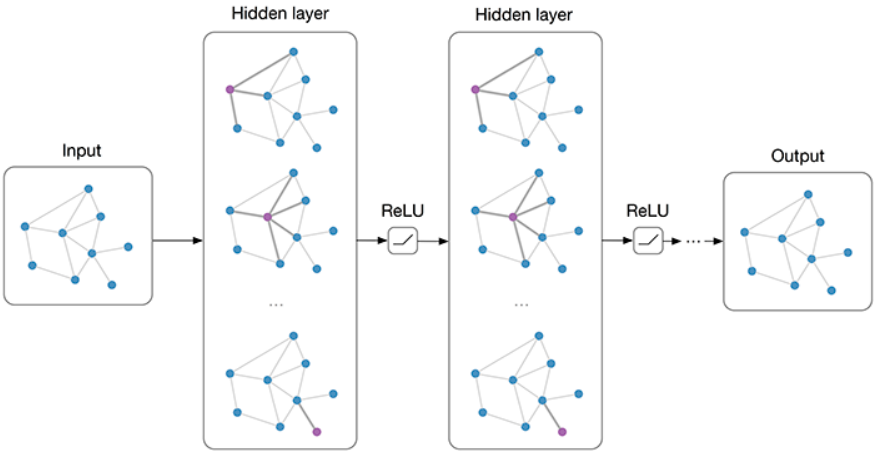

Figure 2: GCN architecture (Kipf and Welling 2017)

Mathematically, it can be represented as:

$$f(x) = \frac{1}{1 + e^{-x}}$$

- **Output:** The final output of GCN can be used for specific tasks like node and graph classification.

Figure 2 (Kipf and Welling 2017) showcases the GCN's architecture. Here, we can see the input, hidden layers, and the output. The hidden layers make use of the ReLU activation function.

GCN works well on shallow graph structures, where any given node contains information from its preceding layers, However, there's a downside to this architecture: oversquashing. Since a given node receives messages from all its previous nodes, it leads to information overload. This will force the information to be compressed, thus leading to oversquashing. Such problem may hinder the model from capturing long-range interactions effectively.

2.2 GraphGPS

On the other hand, GraphGPS (short-hand form for general, powerful, and scalable) adopts a transformer-based architecture that is capable of capturing global dependencies and long-range interactions by employing self-attention mechanisms. Graph transformers are known for their ability to capture intricate relationships and structural information in graphs. GraphGPS uses this 3-part recipe:

- **Positional/structural encoding:** This "ingredient" contains local encodings to capture immediate neighborhood information, global encodings to gain a broader context on nodes, and relative encodings to capture relationships between nodes.
- **Local message-passing mechanism:** This another "ingredient" exchanges information between nodes.
- **Global attention mechanism:** This last "ingredient" enables the information exchange and node-to-node interactions across the whole graph.

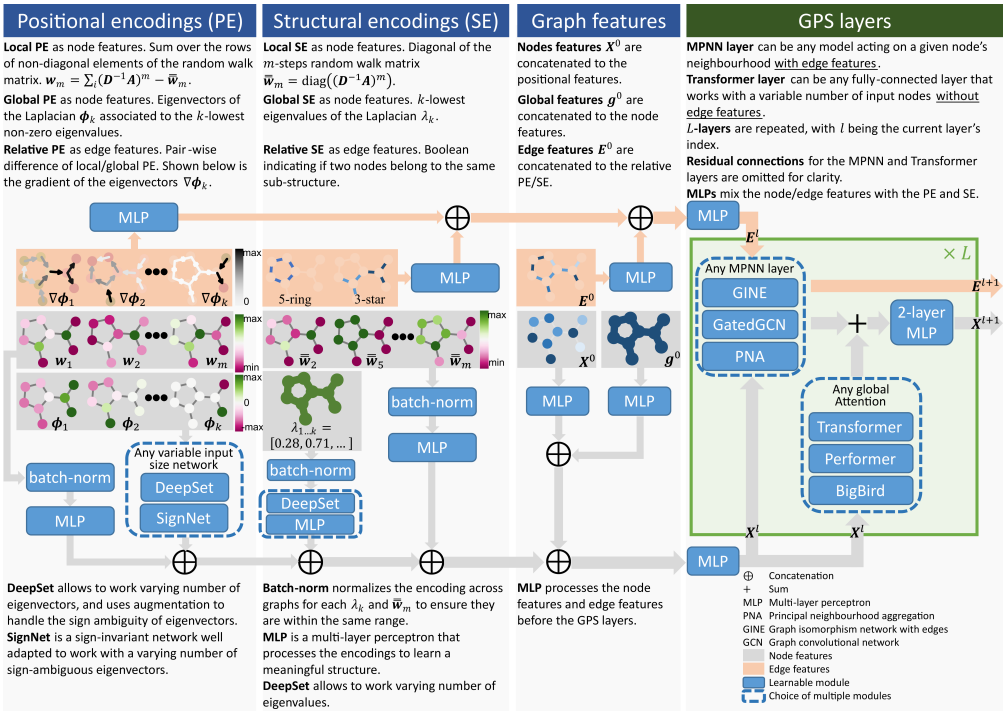

Figure 3: GraphGPS Architecture (Rampášek et al. 2022)

By utilizing these three ingredients for its recipe, GraphGPS is able to perform well on long-range interactions. Its architecture is visualized in Figure 3 (Rampášek et al. 2022). It gives an example of positional (PE) and structural encodings (SE) and analyzes how such encodings help express information in the model.

GraphGPS's self-attention mechanism can incorporate information from distant nodes, that is irrespective of distance. This concept helps mitigate oversmoothing and oversquashing issues prevalent in traditional GNNs.

3 Results

We will first provide some details on our datasets, and compare the models' performance on each of these datasets.

3.1 Datasets

Here are the specifics of our four datasets. Table 1 shows these statistics for an easy comparison. Table 2 provides specifics on the first graph of each dataset.

Table 1: Dataset Statistics

Datasets	# of Graphs	# of Features	# of Classes	ML Task
Cora	1	1433	7	Node classification
IMDB	1000	0	2	Graph classification
Enzymes	600	3	6	Graph classification
PascalVOC-SP	8498	14	21	Node classification

Table 2: Graph 1 Statistics

Datasets	# of Nodes	# of Edges	Avg. Node Degree	Has Isolated Nodes	Has Self-Loops
Cora	2708	10556	3.9	False	False
IMDB	20	146	7.3	False	False
Enzymes	37	168	4.54	False	False
PascalVOC-SP	460	2632	5.72	False	False

3.1.1 Cora

Cora provides insight into the academic citation network. This dataset consists of 2708 scientific publications classified into one of seven classes. The text content of the documents is described as a binary word vector 0/1, where 0 refers to the absence of and 1 refers to the presence of a corresponding word from the dictionary. Its dictionary also contains 1433 unique words used in the research papers. For instance, if a particular word from this dictionary is present on a particular document, then that document would get assigned a value of 1 for that particular word. This dataset is well-suited to test our graph-based architectures because of its natural graph structure and real-world relevance.

3.1.2 IMDB

IMDB is a graph dataset on movies where nodes represent movies and edges represent the relationships between them (shared actors, directors). It contains 1000 graphs and each graph is a movie labeled with a genre. We are tasked with predicting the genre of the movies, which is a graph classification task.

3.1.3 Enzymes

Enzymes contains graphs where nodes represent amino acids and edges represent the spatial relationships between them. It contains 600 graphs and each graph is a protein. We are tasked with predicting the enzyme class for each protein, which is also a graph classification task.

3.1.4 PascalVOC-SP

PascalVOC-SP is part of the long-range graph benchmark datasets (Dwivedi et al. 2022) and it contains graphs where nodes represent image patches and edges represent the spatial

relationships between them. It contains 8498 graphs and each graph is an image. This dataset can be used for either node and graph classification task, and in this paper we will focus on node classification to predict labels for these image patches. This dataset will be used to evaluate models' abilities to handle long-range interactions.

3.2 Models Performance

Table 3 highlights our models' performances. Upon comparing the results, it seems that our model might have overfitted the training set. GCN worked very well on Cora, since it's a short-range dataset but could not perform well on Enzymes, that has longer-range interactions. GIN also worked great on Cora but did not perform very well on Enzymes. GATv2 compared similarly to GIN. GraphGPS performed best on the long-range benchmark dataset, compared to other models. This proves that GraphGPS is useful for detecting long-range interactions. It, however, performed similarly to other models for the rest of the datasets. This tells us that this complex model might not be needed for most datasets, only the ones that contain long-range interactions.

Table 3: Model Accuracy

Datasets	**GCN**		**GIN**		**GATv2**		**GraphGPS**	
	Train	Test	Train	Test	Train	Test	Train	Test
Cora	96.43	**76.30**	**99.29**	64.70	95.99	73.70	91.43	61.70
IMDB	60.71	56.00	60.94	58.67	50.12	49.33	**72.31**	**70.15**
Enzymes	29.56	30.00	35.33	36.00	38.44	36.00	**76.82**	**73.33**
PascalVOC-SP	69.95	69.50	69.95	69.50	66.56	66.02	**74.12**	**73.95**

4 Implentation Details

All four datasets and the three baseline models were loaded from Pytorch Geometric (Fey and Lenssen 2019). GraphGPS is derived from a paper and its respective GitHub repo (Rampášek et al. 2022). The datasets were split into test and train sets, as shown in Table 2. The datasets were pre-processed. For instance, since Cora did not contain any node features, 1 was used as its replacement. We also normalized the features so our models can converge faster during training and for stability so the model will not allow certain values to dominate too much. We used accuracy as a metric to evaluate our model.

Details of the implementation:

- **Layers #:** 2
- **Weight decay:** 0.0005
- **Optimizer:** Adam
- **Learning rate:** 0.001

- **Epochs #:** 1000

For our inner layers of the model, we used the ReLU activation function to add non-linearity. For the GATv2 model, we used 8 heads to incorporate self-attention mechanisms. We used soft-max for each of our model's output layers. We also used a batch size of 64 for each model.

5 Conclusion

In this study, we were able to explore different ranges of datasets on baseline GNN and a transformer-based model. We saw that the baseline model works alright on datasets with short-range interactions and GraphGPS works well on datasets with long-range interactions.

In subsequent studies, we will use GridSearch CV to fine-tune our hyperparameters and choose the one that performs well on that model and that dataset. We will also explore the Enzymes dataset further and look into ways we can surpass current performance.

References

Alon, Uri, and Eran Yahav. 2021. "On the Bottleneck of Graph Neural Networks and its Practical Implications." In *International Conference on Learning Representations*. [Link]

Dwivedi, Vijay Prakash, Ladislav Rampášek, Mikhail Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. 2022. "Long Range Graph Benchmark." In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. [Link]

Fey, Matthias, and Jan E. Lenssen. 2019. "Fast Graph Representation Learning with PyTorch Geometric." In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Kipf, Thomas N., and Max Welling. 2017. "Semi-Supervised Classification with Graph Convolutional Networks." In *International Conference on Learning Representations (ICLR)*. [Link]

Oono, Kenta, and Taiji Suzuki. 2020. "Graph Neural Networks Exponentially Lose Expressive Power for Node Classification." In *International Conference on Learning Representations*. [Link]

Rampášek, Ladislav, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. 2022. "Recipe for a General, Powerful, Scalable Graph Transformer." *Advances in Neural Information Processing Systems* 35

Veličković, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. "Graph Attention Networks." *International Conference on Learning Representations*. [Link]

Xu, Keyulu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. "How Powerful are Graph Neural Networks?" In *International Conference on Learning Representations*. [Link]