Interpreting Graph Transformers for Long-Range Interactions

Ryan Hung ryhung@ucsd.edu & James Thai jqthai@ucsd.edu & Mentor: Yusu Wang yusuwang@ucsd.edu & Mentor: Gal Mishne gmishne@ucsd.edu

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

Introduction

Graph transformer architecture offers strong performance in predictive tasks requiring the capture of long-range interactions. But as model complexity grows, it becomes difficult to reason about the decision making process. How do we uncover, in a graph setting, the set of nodes that best explain a transformers prediction?

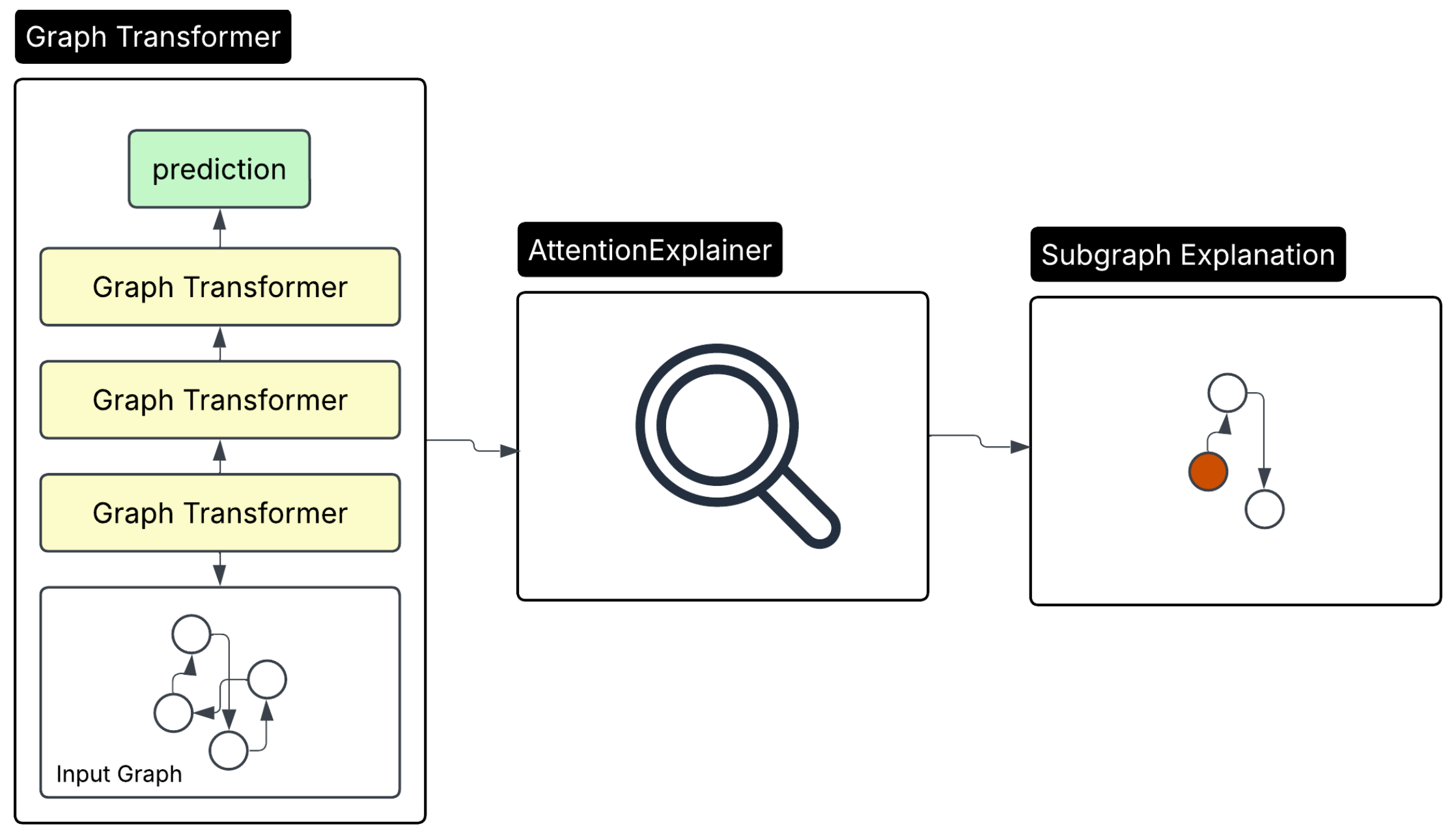

Figure 1. Explainability Flow

We introduce **AttentionExplainer**, an attention-based explainability algorithm for any graph transformer architecture leveraging global-self attention. We also propose **IGExplainer**, a gradient-based explainability algorithm inspired by NLP transformer interpretability.

Methods

We formulate **AttentionExplainer** with several algorithm variants based upon the attention matrix A in which A_{ij} denotes how much node x_i attends to node x_j.

An explanation for node x_i is a subgraph $G_s \subseteq G$ that best explains why node x_i was predicted with class $\hat{y}$.

- Greedily select the top K attention weights for node x_i, and perform a shortest path walk to generate subgraph explanation.
- Greedily select the top K attention weights for node x_i from its subset of neighbors to generate subgraph explanation.

We also propose **IGExplainer**, which uses integrated gradients to generate edge attributions – importance – to a prediction. Here, we linearly interpolate from a baseline $v' = \{0, ..., 0\}$ to the original input $v = \{1, ..., 1\}$. The method is dependent on the positonal encodings used, as these encodings provide graph topological information to both the transformer learning phase and the gradient computation step.

The algorithm is detailed below:

Compute positional encodings (random walk example shown)

$$p_i^{\text{RWPE}} = \left[\text{RW}_{ii}, \text{RW}_{ii}^2, \ldots, \text{RW}_{ii}^k\right] \in \mathbb{R}^k \text{ where } RW = AD^{-1} \tag{1}$$

$$PE = \begin{bmatrix} p_1^{\text{RWPE}} \\ \vdots \\ p_n^{\text{RWPE}} \end{bmatrix} \in \mathbb{R}^{n \times k} \tag{2}$$

Concatenate feature and positional encodings $X' = [X \ PE]$
Train GPS model $GPS(X', A)$
for each edge index $i \in \vec{v}$, **do**:

$$\text{IntegratedGrads}_i(v) ::= (v_i - v_i') \times \int_{\alpha=0}^{1} \frac{\partial \text{GPS}(v' + \alpha \times (v - v'))}{\partial v_i} \, d\alpha$$

Assemble attribution vector $\vec{I} = \{\text{IntegratedGrads}_1(\vec{v}), ..., \text{IntegratedGrads}_n(\vec{v})\}$
Initialize subgraph $G_s \leftarrow \emptyset$
$G_s = \text{compute}_{G_s}(I,\ A)$

Data

We use two datasets: BAShapes, a synthetically generated preferential attachment Barabasi-Albert graph with attached house motifs, and PascalVOC-SP, a computer vision network originating from semantic segmentation. BAShapes consists of a ground truth explanation mask we use to compute explanation accuracy, precision, and recall. PascalVOC-SP comes from the long-range interaction benchmark. Because this is not a synthetic dataset, there is no explanation truth between node classes and their influence upon each other. As such, we evalute PascalVOC-SP using fidelity, which does not require explanation truth labels.

An Illustrated Example

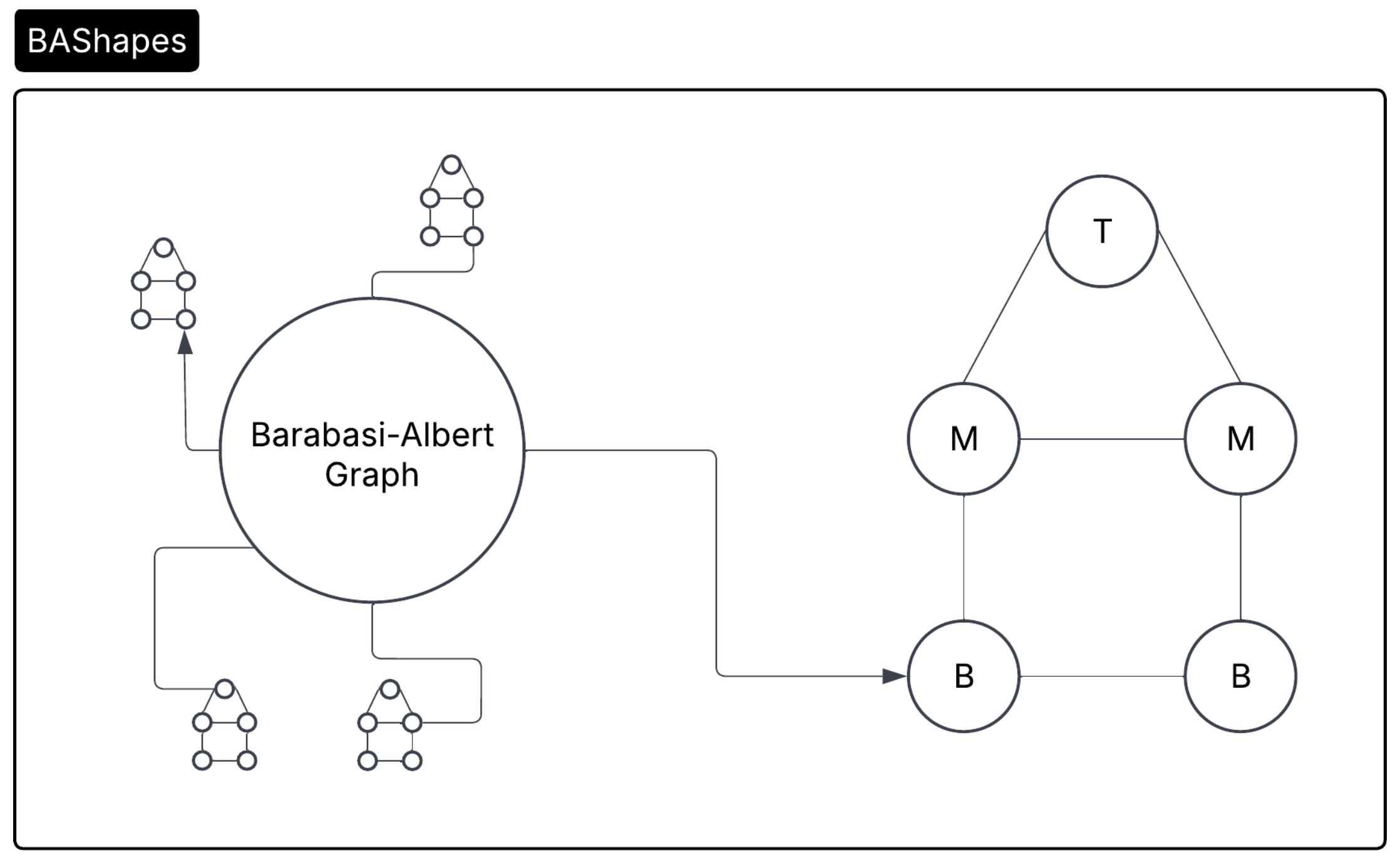

Figure 2. BAShapes Topology

A robust graph neural network shoudl easily identify the recurring house pattern structure by looking at its immediate neighbors. Each set of house nodes is labeled consecutively such that $\{x_i, ..., x_{i+4}\}$ is in a house. The classes are the structural roles inherited by a node: top, middle, and bottom. In BAShapes, there is an explanation truth because the graph is synthetically generated. The house motif structure provides a structural axiom in that nodes determine each other's classes based on 1-hop neighborhood.

At each layer post-training, we average each attention head and normalize to retrieve attention scores for each node.

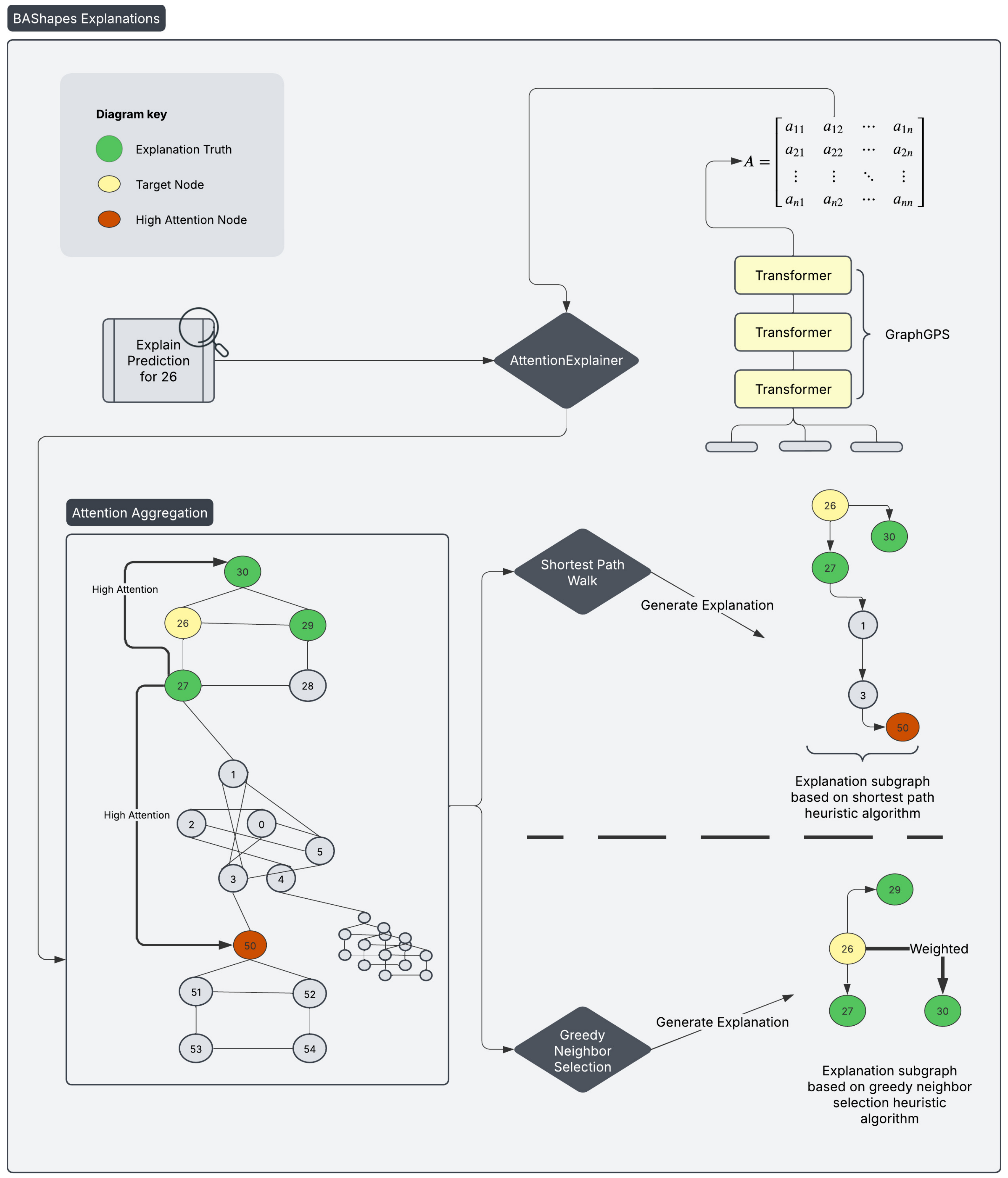
BAShapes Explanations
Diagram key
Explanation Truth
Target Node
High Attention Node
Explain Prediction for 26
AttentionExplainer
$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$
Transformer
Transformer
Transformer
GraphGPS
Attention Aggregation
High Attention
High Attention
30
26
29
27
28
1
2
0
5
3
4
50
51
52
53
54
Shortest Path Walk
Generate Explanation
Explanation subgraph based on shortest path heuristic algorithm
Greedy Neighbor Selection
Generate Explanation
Weighted
Explanation subgraph based on greedy neighbor selection heuristic algorithm

Figure 3. Shortest Path Walk Edge Mask Generation

For instance, to explain the decision-making behind node 26's prediction, we consult the trained matrix. From here, we identify the set of nodes that node 26 attends to most, and create an edge mask using a heuristic-based shortest path walk.

Results

Table 1. Explainer Accuracy for BAShapes

Method	GNN Model	Dataset	Explanation Accuracy ↑	Recall ↑	Precision ↑	F1 ↑
GNNExplainer	GPS GCN	BAShapes	94.0 ± 1.1 95.3 ± 0.8	10.0 ± 1.3 **100.0** ± 0.7	10.0 ± 1.5 36.3 ± 1.7	16.0 ± 1.3 **57.7** ± 1.2
DummyExplainer	GPS GCN	BAShapes	94.2 ± 1.2 91.3 ± 1.0	0.0 ± 0 0.0 ± 0	1.0 ± 0 0.0 ± 0	3.0 ± 1.2 0.0 ± 0
AttentionExplainer*	GPS	BAShapes	93.35 ± 1.3	25.0 ± 2.5	11.76 ± 1.5	15.99 ± 1.4
IGExplainer*	GPS	BAShapes	**96.35** ± 1.9	94.3 ± 2.1	**36.76** ± 1.1	52.9 ± 1.6

Table 2. Explainer Fidelity for BAShapes (Maximized Against Characterization Score)

Method	GNN Model	Dataset	Fid+ ↑	Fid- ↓	Characterization Score ↑
GNNExplainer	GPS GCN	BAShapes	66.6 ± 3.1 25.3 ± 0.3	66.6 ± 0.9 61.4 ± 0.9	44.4 ± 2.8 30.6 ± 0.5
DummyExplainer	GPS GCN	BAShapes	46.6 ± 1.5 45.3 ± 1.1	73.5 ± 0.3 **48.0** ± 2.5	34.0 ± 0.9 48.4 ± 1.5
AttentionExplainer*	GPS	BAShapes	61.33 ± 1.4	57.33 ± 0.5	50.32 ± 1.9
IGExplainer*	GPS	BAShapes	**91.73** ± 3.1	51.31 ± 2.3	**65.06** ± 2.7

Table 3. Explainer Fidelity for PascalVOC-SP (Maximized Against Characterization Score)

Method	GNN Model	Dataset	Fid+ ↑	Fid- ↓	Characterization Score ↑
GNNExplainer	GPS GCN	PascalVOC-SP	12.0 ± 1.5 28.2 ± 0.9	56.0 ± 1.8 68.2 ± 1.3	19.3 ± 1.2 15.3 ± 1.7
AttentionExplainer*	GPS	PascalVOC-SP	42.3 ± 0.7	**2.4** ± 1.3	58.7 ± 1.1
IGExplainer*	GPS	PascalVOC-SP	**48.0** ± 1.2	6.0 ± 1.9	**65.3** ± 1.4

- Explanation Accuracy reflects our model's ability to capture the proper house motifs
- Recall measures the proportion of house motif nodes that is captured; Precision measures the proportion of captured nodes that are house motif nodes (F1 Score balances both)
- Fid+ and Fid- quantify whether a generated explanation is "necessary" or "sufficient" to the outcome of a prediction, respectively (Characterization score combines both metrics)

Discussion

Our results indicate that **AttentionExplainer** and **IGExplainer** are the most robust compared to GNNExplainer and DummyExplainer when measured against Characterization Score for both BAShapes and PascalVOC-SP datasets. This means that **AttentionExplainer** and **IGExplainer** produce the highest quality explanations that are both necessary and sufficient.

While the recall, precision, and F1 score is maximized on GNNExplainer, it is maximized on GCNs, and the high performance is not reflected when applied to GPS. **IGExplainer** displays competitive results to GNNExplainer in this regard, meaning that it captures the relevant nodes in the explanations while avoiding the irrelevant ones.

Our metrics across all three tables for recall, precision, and fidelity demonstrate that **AttentionExplainer** and **IGExplainer** are able to effectively capture the decision making within long range interactions that occurs in graph transformers.

To conclude, the importance of neural network explainability has grown alongside the increasing complexity of both data and model architectures. Specifically, graph neural network explainability aims to shed light on the decision-making processes of models that incorporate both features and relational structures during training. However, there is a notable gap in the literature regarding the explainability of graph transformers. To address this, we propose two methods: **AttentionExplainer** and **IGExplainer**. The former utilizes trained attention weights to greedily generate subgraph explanations, while the latter applies integrated gradients to compute edge attribution for each edge in the graph. Our approach is directly applicable to any graph transformer model that employs self-attention, offering a more appropriate explanation framework for this architecture compared to existing explainer techniques that are primarily designed for message-passing neural networks. Our explainer algorithms provide efficient and vital interpretations of graph transformer decision-making, providing critical model transparency, fairness, ethicality, and legal adherence.

References

[1] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *CoRR*, abs/1703.01365, 2017. URL http://arxiv.org/abs/1703.01365.

[2] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNN explainer: A tool for post-hoc explanation of graph neural networks. *CoRR*, abs/1903.03894, 2019. URL http://arxiv.org/abs/1903.03894.

https://ryanhungry.github.io/Interpreting-Graph-Transformers-for-Long-Range-Interactions/ HDSI Senior Capstone 2025, San Diego Report - Click Here