Towards Semantic Interpretation and Validation of Graph Attention-based Explanations

Efimia Panagiotaki[1], Daniele De Martini[2], Lars Kunze[1]
[1]Cognitive Robotics Group and [2]Mobile Robotics Group, Oxford Robotics Institute,
Department of Engineering Science, University of Oxford, UK
{efimia,daniele,lars}@robots.ox.ac.uk

***Abstract*— In this work, we investigate the use of semantic attention to explain the performance of a Graph Neural Network (GNN)-based pose estimation model. To validate our approach, we apply semantically-informed perturbations to the input data and correlate the predicted feature importance weights with the model's accuracy. Graph Deep Learning (GDL) is an emerging field of machine learning for tasks like scene interpretation, as it exploits flexible graph structures to describe complex features and relationships in a very concise format. However, due to the unconventional structure of the graphs, traditional explainability methods used in eXplainable AI (XAI) require further adaptation and thus, graph-specific methods are introduced. Attention is a powerful tool, introduced to estimate the importance of input features in deep learning models. It has been previously used to provide feature-based explanations on the predictions of GNN models. In our proposed work, we exploit graph attention to identify key semantic classes for lidar pointcloud pose estimation. We extend the current attention-based graph explainability methods by investigating the use of attention weights as importance indicators of semantically sorted feature sets by analysing the correlation between attention weights distribution and model accuracy. Our method has shown promising results for post-hoc semantic explanation of graph-based pose estimation.**

***Index Terms*— Attention, eXplanable AI, graph neural networks, pose estimation**

I. Introduction

Trustworthy Graph Learning (TwGL) identifies reliability, explainability, accountability, and other trust-oriented features, as key requirements for trustworthy Graph Deep Learning (GDL) [1], [2]. Undeniably, trust is a critical design factor for the successful development and deployment of self-driving vehicles. Trust and explainability are inherently linked; explaining the decisions of autonomous vehicles enables users and regulatory bodies to use and work on a transparent and accountable system. Having a clear understanding of the capabilities and limitations of the autonomous system increases trust in the underlying technology and fosters its adoption.

In real-world deployment, autonomous vehicles must operate safely in unknown and dynamic environments. To ensure safe operation, the system needs to assess the complexity of the environment and make logical decisions based on expected performance. A critical prior requirement for reliable decision-making is for those vehicles to know their precise location relative to the observed environment. This relates to the task of *pose estimation*, which calculates the position of the ego-vehicle with respect to the perceived features.

Our proposed research focuses on analysing and explaining the complexity of the environment using learned attention weights to identify the contribution of each semantic element, i.e. static and dynamic agents as well as morphological structures, to the performance of a baseline lidar pointcloud-based pose estimation model. Similar to [3], to investigate the validity of using attention weights as feature importance indicators, we take inspiration from perturbation-based Graph eXplainable AI (GXAI) methods. In our work, we extract sorted semantic sets based on their attention scores and then semantically perturb the input to measure the correlation between attention weights and model accuracy. These measurements correspond to semantic importance indicators of input features. As proposed in [4], [5], [6], in each perturbation, we measure the distribution divergence to calculate the contribution of each sets' attention weights to the overall attention distribution, assessing the validity of the importance estimates.

Our key contributions are as follows:

- A methodology for semantic interpretation of attention to explain the predictions of a graph-based model.
- A semantically-informed perturbation process for evaluating the explanations for GXAI.

The model used as baseline is a graph-attention-based pose estimation model, SEM-GAT [1], trained on the KITTI Odometry Dataset [7].

II. Related Work

Recent studies have investigated the topic of explainability in Graph Neural Networks (GNNs) proposing different approaches to explain the predictions. Following the taxonomy introduced in [8] for instance-level explanations, these methods can be categorised in: gradient/feature-based, decomposition-based, surrogate-based, and perturbation-based.

Gradient/feature-based methods [9], [10] calculate the gradients and feature values, to approximate importance scores for the input. Decomposition-based methods [10],

This work was supported by a DeepMind Engineering Science Scholarship, the EPSRC project RAILS (grant reference: EP/W011344/1), and the Oxford Robotics Institute research project RobotCycle.

[1]Under review at IROS 2023.

[11], [12] estimate the importance scores by decomposing the output predictions and finding the corresponding input features with back propagation. Surrogate-based methods [13], [14], [15] use simple and interpretable input features extracted from the neighbors of the input nodes to explain the original model. Perturbation-based methods [16], [17], [18], [19], [20], [21] measure importance scores by masking the input and calculating the changes in the output predictions, generating post-hoc explanations.

Perturbation-based methods are the most relevant to our approach. However, these methods rely on random masks to perturb the input. We argue that exploiting the properties of input features to extract the masks generates more efficient and concise perturbations. In our proposed methodology, we estimate importance scores for the input features to generate semantic sets for masking. Through sequential perturbations using those sets, we generate explanations for a Graph Neural Network (GNN) model.

Various methods exploit attention to interpret the input features and explain the predictions of deep learning models [22], [23], [24]. However, using attention weights to provide a holistic explanation of the output predictions has previously been regarded as an insufficient and inaccurate interpretability technique [5]. This argument was challenged in later studies [4] claiming that to test if attention can be considered an explainability method, we need to examine all aspects of the model. Attention can, in some cases, be used as an explainability technique, however this is not always accurate and cannot be generalised [6]. As suggested in [3], further investigation is required to verify that attention weights relate to feature importance.

Following these studies, we evaluate the validity of our attention-based explainability method by correlating the accuracy of the baseline model with the divergence of the attention weights distribution in each perturbation. Our results demonstrate that, for the model used as baseline, attention can be useful to identify important semantics in the environment that contribute towards reliable performance.

III. Preliminaries

The overview of our explainability pipeline is visualised in Fig. 1. In this section, we formulate the problem addressed and then briefly describe the graphs and the GNN model used as baseline.

A. Problem Definition and Notations

Let $P_t : \{\mathbf{p}_i \mid \mathbf{p}_i \in \mathbb{R}^3\}$ be a pointcloud at discrete timestamp t in a total of N consecutive scans. P_t can be subdivided into a set of semantic classes $\mathbb{S}$ that may include *terrain*, *buildings*, *trees*, *vehicles*, and *pedestrians*, among others. For each point $\mathbf{p}_i$, we assign a semantic label $s_i \in \mathbb{S}$. In our proposed work, we aim to identify those semantic classes that contribute the most in the accurate estimation of the relative pose transformation between two consecutive pointclouds, P_t and P_{t+1}, denoted as $\mathbf{R}_{t,t+1} \in \mathbb{SO}(3)$ for rotation and $\boldsymbol{\tau}_{t,t+1} \in \mathbb{R}^3$ for translation.

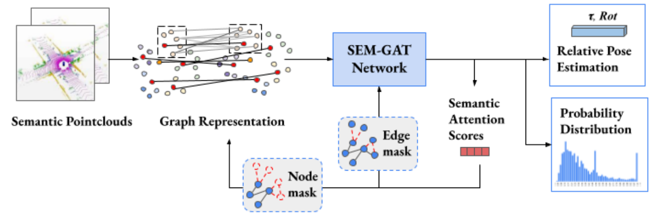


Fig. 1: Overview of our proposed methodology. After retrieving the attention weights for each semantic class from vanilla SEM-GAT, we use an *edge mask* to mask the highest ranking semantic class sets at the last layer of the model and measure the divergence of the attention weights distribution. We correlate this measurement with the pose estimation error from masked SEM-GAT, to generate importance scores for each semantic set. We repeat this process, perturbing the input of the model using a *node mask* to mask the adjacency matrix of the input graphs.

B. SEM-GAT

The model used as the basis for generating attention-based explanations is SEM-GAT, a semantic graph-based pose estimation GNN model depicted in Fig. 2. SEM-GAT estimates the relative transformation between two pointclouds by identifying potential point matching correspondences, known as *registration candidates*. SEM-GAT then explicitly exploits attention to weigh each candidate pair for pose estimation, making it a suitable baseline to test our evaluation methodology.

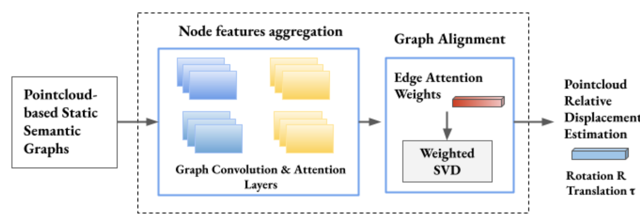


Fig. 2: Outline of SEM-GAT, the attention-based GNN used as baseline for generating and validating the semantic explanations of our method.

Input: We define the input graphs as $G_k = \langle V_k, E_k \rangle$, $k \in \{1, ..., N-1\}$, where V_k and E_k represent the sets of nodes and edges, respectively. Given P_t and P_{t+1}, we construct a static graph representation G_k of the two pointclouds by semantically linking the nodes in the graph to generate a graph-structure representation of the environment. Each point $\mathbf{p}_i$ is represented as a node. The edges correspond to the semantic relationships between the points according to their associated semantic label $s_i \in \mathbb{S}$ and their geometric characterisation as *corner* or *surface* points, based on their local neighbourhood's geometry.

SEM-GAT: As described in Sec. III-A, SEM-GAT estimates the relative pose transformations, $\mathbf{R}_{t,t+1}$ and $\boldsymbol{\tau}_{t,t+1}$, between P_t and P_{t+1}. To achieve this, the model first needs to align the two pointclouds by finding the nearest point-to-point correspondences for pointcloud registration. SEM-GAT finds strong registration candidates by generating embedding representations of the nodes in the input graphs. These embeddings encode structural and semantic information from the local neighborhood of the nodes. The model uses a series of Graph Convolution Networks (GCNs), followed by multi-head Graph Attention Networks (GATs) that assign attention weights α as confidence scores to the edges connecting potential registration candidates. These scores are then used as weights in a Singular Value Decomposition (SVD) module to align the pointclouds and eventually recover the relative transformation $\mathbf{R}_{t,t+1}$ and $\boldsymbol{\tau}_{t,t+1}$ between them.

IV. Attention-based Semantic Explanations

We estimate the importance of various semantic elements in the environment using the attention weights α predicted in the last layer of SEM-GAT. To validate the suitability of using attention to semantically explain the performance of SEM-GAT, we iteratively perturb the input, correlating the attention weights distribution divergence with the changes in the model's accuracy.

We first investigate the semantic interpretation of attention weights α, by ranking the semantic classes at inference step according to their predicted average α scores. Based on this ranking, we extract semantic feature sets to iteratively mask the model's input while measuring the variations in the output. The perturbations are then conducted in two steps, visualised in Fig. 3:

1) Masking the adjacency matrix of the input graphs based on the average overall attention score of the semantic sets.
2) Zeroing the edge attention weights that belong to our estimated most important semantic sets, at the last layer of SEM-GAT.

Following the outcome of the perturbations, we evaluate the adequacy of using attention weights as importance scores. The validation process can be split in two parts: measuring the attention distribution divergence and correlating the attention scores with the accuracy of the model.

A. Attention Distributions

To estimate the importance of the perturbed attention weights to the overall weights distribution, we measure the distribution divergence in correlation with the model's output prediction scores. To calculate the divergence, we measure the similarity of the distributions α_{k_b} and α_{k_a} respectively, before and after masking, using the Jensen-Shannon Divergence (JSD) distance:

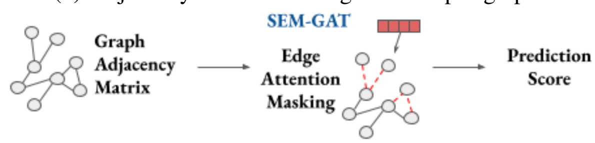
$$JSD(\alpha_{k_b}, \alpha_{k_a}) = \sqrt{\frac{D_{KL}(\alpha_{k_b} \parallel \bar{\alpha}) + D_{KL}(\alpha_{k_a} \parallel \bar{\alpha})}{2}} \tag{1}$$

with $0 \leq JSD(\alpha_{k_b}, \alpha_{k_a}) \leq 1$ and $\bar{\alpha} = \frac{\alpha_{k_b} + \alpha_{k_a}}{2}$.

D_{KL} corresponds to the Kullback-Leibler divergence and $\bar{\alpha}$ is the pointwise mean of α_{k_b} and α_{k_a}. The JSD distance corresponds to the square root of the JSD metric, used in [4], [5], [6].

(a) Average attention scores as importance weights.

(b) Adjacency matrix masking of the input graphs.



(c) Attention weights zeroing at the last layer of SEM-GAT.

Fig. 3: Overview of the perturbation process as *Input* $\rightarrow$ *Model* $\rightarrow$ Output: (a) visualises the process of extracting the semantic importance weights from vanilla SEM-GAT. These weights then inform the adjacency matrix masking in (b), and the edge attention weights masking in (c). The masking steps in (b) and (c) are independent of one another.

B. Attention-Accuracy Correlation

As we perturb the input, we measure the variations in SEM-GAT's pose estimation accuracy to assess the correlation between attention and model performance. The authors in [3] propose using the discrepancy in the model's accuracy, before and after masking, to estimate the importance of the input features. Similar to this approach, we calculate the average absolute discrepancy, $\mathbb{E}$, of the accuracy scores $\hat{\mathbf{y}}_{k_b}$ and $\hat{\mathbf{y}}_{k_a}$ from before and after applying masking, respectively. For our case, $\hat{\mathbf{y}}_k = \{RelativeRotationalError(RRE), RelativeTranslationalError(RTE)\}$.

$$\mathbb{E}(\hat{\mathbf{y}}_b, \hat{\mathbf{y}}_a) = \frac{\sum_{i=1}^{N-1} |\hat{\mathbf{y}}_{k_b} - \hat{\mathbf{y}}_{k_a}|}{N-1} \tag{2}$$

This metric is a good indicator of the fluctuations in the output predictions in each perturbation step.

V. Experimental Setup

SEM-GAT is trained on Sequences $00, 02, 03$ of the KITTI Odometry Dataset [7]. We test and evaluate our approach on Sequences $00 - 10$ as we are interested in performance deviations. To generate our semantic graphs and evaluate the performance of SEM-GAT, we use the ground truth labels and ground truth poses from SemanticKITTI [25].

A. Evaluation Metrics

The rotation $\hat{\mathbf{R}}$ and translation $\hat{\boldsymbol{\tau}}$ estimation outputs from SEM-GAT are evaluated using the error metrics RRE [°] and RTE [m]:

$$RRE = \operatorname{acos}\left(\frac{1}{2}(\operatorname{tr}(\mathbf{R}_{gt}^{\top}\hat{\mathbf{R}}) - 1)\right) \tag{3}$$

$\hat{\mathbf{R}}$ and $\mathbf{R}_{gt}$ are the estimated and ground-truth rotation matrices, respectively, and

$$RTE = \|\boldsymbol{\tau}_{gt} - \hat{\boldsymbol{\tau}}\|_2 \tag{4}$$

$\hat{\boldsymbol{\tau}}$ and $\boldsymbol{\tau}_{gt}$ are the estimated and ground-truth translation vectors. The combined average absolute discrepancy $\mathbb{E}$ is then calculated as follows:

$$\mathbb{E}(RRE_{a,b}, RTE_{a,b}) = \frac{\sum_{i=1}^{N-1} |RRE_{k_b} - RRE_{k_a}|}{N-1} + \frac{\sum_{i=1}^{N-1} |RTE_{k_b} - RTE_{k_a}|}{N-1} \tag{5}$$

We correlate $\mathbb{E}$ with JSD in Eq. (1) to estimate the contribution of the query semantic importance scores to the accuracy of the model.

B. Semantic Masking

We use the predicted attention weights from the last layer of SEM-GAT to rank the semantic classes in the dataset and extract semantic feature sets. To estimate the importance of each set, we identify five classes with the highest average learned attention scores for each sequence.

	Per-class Average Attention Scores in Descending Order (→)				
00	pole (0.55)	sidewalk (0.53)	fence (0.44)	building (0.4)	bicycle (0.4)
01	fence (0.51)	vegetation (0.42)	terrain (0.39)	car (0.29)	ground (0.18)
02	sidewalk (0.56)	fence (0.48)	trunk (0.45)	vegetation (0.4)	pole (0.36)
03	pole (0.55)	sidewalk (0.55)	fence (0.5)	vegetation (0.38)	terrain (0.38)
04	sidewalk (0.6)	pole (0.49)	fence (0.45)	car (0.44)	vegetation (0.43)
05	sidewalk (0.56)	terrain (0.5)	fence (0.47)	car (0.4)	building (0.4)
06	pole (0.6)	sidewalk (0.57)	trunk (0.52)	terrain (0.45)	car (0.45)
07	pole (0.56)	sidewalk (0.54)	fence (0.46)	building (0.4)	car (0.39)
08	sidewalk (0.55)	pole (0.51)	terrain (0.43)	trunk (0.42)	building (0.4)
09	sidewalk (0.55)	terrain (0.44)	trunk (0.43)	vegetation (0.39)	fence (0.38)
10	pole (0.49)	fence (0.47)	sidewalk (0.44)	vegetation (0.38)	building (0.37)

TABLE I: Attention-based importance ranking of semantic classes in Sequences $00-10$ in SemanticKITTI [25]. This ranking guides the perturbations. Seq. 00, 02, and $06-09$ were captured in urban environments, Seq. $03-05$, and 10 in the countryside, and Seq. 01 in a highway.

According to the ranking in Tab. I, we split and perturb the input data in the following semantic sets:

- Single-class attention weights; masking of the top 3 highest-scoring classes, successively
- Multi-class attention weights; masking of the top 3 and top 5 highest-scoring classes
- Single-feature; masking of all the corner or all the surface points

We then evaluate whether the attention weights on these sets actually represent key semantic structures in the environment based on their contribution to SEM-GAT's performance.

VI. Results

A. Attention Distributions

To estimate the contribution of each masking set to the total distribution of attention weights predicted in the last layer of SEM-GAT, we freeze the attention weights and calculate the JSD distance of the distributions before and after removing the weights that correspond to each set.

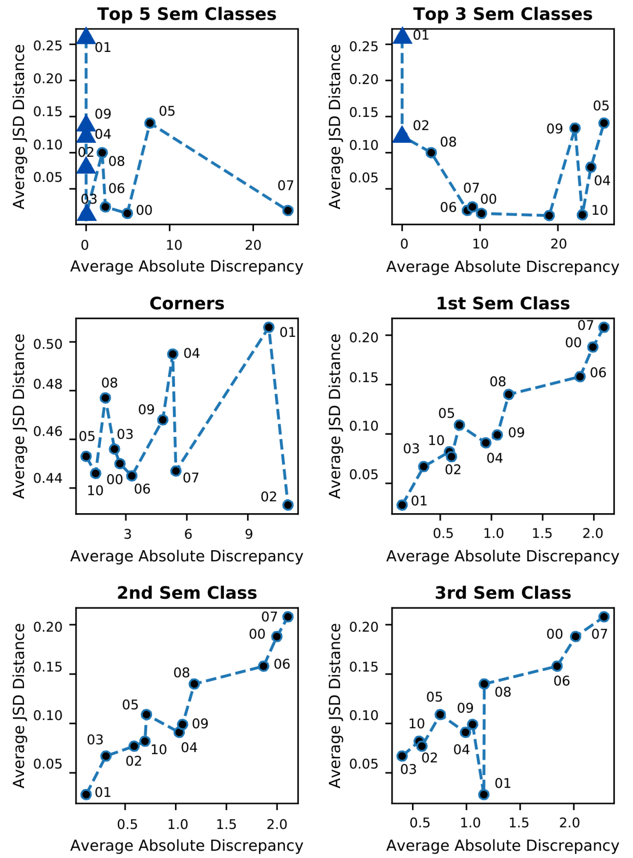


Fig. 4: Average JSD distance correlation with the average absolute discrepancy $\mathbb{E}$, calculated after perturbing the last layer of SEM-GAT for Seq. 00-10 in SemanticKITTI.

Higher JSD values correspond to larger overall contribution of the query set of semantic attention weights to the total distribution of attention.

Our initial results, visualised in Fig. 4, demonstrate that the attention weights in the *Single-feature;Corner* set correspond to almost half of the total distribution. The JSD distances gradually decrease as we mask the *Multi-class;5*, *Multi-class;3*, and *Single-Class* sets, indicating lesser contribution of the perturbed data to the attention weights distribution. As expected, the average absolute discrepancy scores follow a similar behavior.

For semantically-poor sequences, our multi-class masking methodology proved to be too aggressive; the model's prediction accuracy would significantly decrease resulting to pose estimation loss. This behavior is visualised in Fig. 4 where the Average Absolute Discrepancy returned an invalid value.

We are particularly interested in these results as they are an initial indicator that the average absolute discrepancy is almost proportional with JSD for most masking sets. As expected, the masking sets *Single-feature;Corner* and *Multi-class* produce higher overall JSD and $\mathbb{E}$ scores compared to the scores from *Single-class* masking.

seq	Multi-class				Single-feature				Single-class					
	Top 5 Classes		Top 3 Classes		Surfaces		Corners		1st Class		2nd Class		3rd Class	
	RTE	RRE	RTE	RRE	RTE	RRE	RTE	RRE	RTE	RRE	RTE	RRE	RTE	RRE
00	**0.752**	**0.042**	0.057	0.010	0.066	**0.037**	**2.980**	0.022	0.352	0.016	0.352	0.016	**0.355**	**0.017**
01	**—**	**—**	**—**	**—**	4.440	**0.042**	**6.500**	0.027	0.306	0.002	**0.405**	**0.003**	0.305	0.001
02	**—**	**—**	**—**	**—**	**3.715**	**0.056**	1.279	0.008	**0.263**	**0.003**	0.259	0.003	0.257	0.003
03	**—**	**—**	0.215	0.017	2.038	0.016	**5.372**	**0.040**	**0.152**	**0.002**	0.152	0.002	0.152	0.002
04	**—**	**—**	1.444	0.010	2.919	0.026	**4.059**	**0.032**	**1.112**	**0.002**	1.112	0.001	1.112	0.001
05	**0.62**	**0.07**	0.221	0.024	0.025	0.027	**3.346**	**0.036**	0.231	**0.005**	0.231	0.005	**0.238**	0.005
06	**0.76**	**0.015**	0.349	0.006	**2.224**	0.020	1.137	**0.026**	1.012	**0.009**	1.012	0.009	**1.028**	0.008
07	**3.232**	**0.209**	0.195	0.06	0.744	**0.049**	**4.571**	0.017	0.798	**0.013**	0.799	0.013	**0.97**	0.013
08	0.267	**0.022**	**0.704**	0.003	0.095	**0.032**	**2.823**	0.013	0.338	**0.008**	**0.339**	0.008	0.338	0.008
09	**—**	**—**	1.203	0.010	**1.715**	**0.035**	0.611	0.009	0.329	**0.007**	0.329	0.007	**0.332**	0.007
10	**—**	**—**	0.103	0.024	0.971	0.020	**4.666**	**0.060**	0.153	0.004	**0.171**	**0.005**	0.154	0.004

seq	Multi-class				Single-feature				Single-class					
	Top 5 Classes		Top 3 Classes		Surfaces		Corners		1st Class		2nd Class		3rd Class	
	RTE	RRE	RTE	RRE	RTE	RRE	RTE	RRE	RTE	RRE	RTE	RRE	RTE	RRE
00	**0.753**	**0.042**	0.055	0.009	0.337	**0.036**	**2.478**	0.022	**0.352**	**0.016**	0.352	0.016	0.352	0.016
01	**—**	**—**	**—**	**—**	5.025	**0.045**	**5.205**	0.030	0.306	0.002	**0.406**	**0.003**	0.306	0.002
02	**—**	**—**	**—**	**—**	**4.050**	**0.058**	0.976	0.009	0.259	**0.003**	0.257	0.003	**0.263**	0.003
03	**—**	**—**	0.212	0.016	2.542	0.017	**4.535**	**0.035**	0.151	**0.002**	**0.152**	0.002	0.152	0.002
04	**—**	**—**	1.445	0.011	2.998	**0.029**	**3.040**	0.027	**1.112**	0.002	1.112	0.002	1.112	**0.003**
05	**0.621**	**0.070**	0.227	0.023	0.196	0.027	**2.653**	**0.033**	0.232	**0.005**	0.232	0.005	**0.239**	0.005
06	**0.76**	**0.016**	0.344	0.005	**2.625**	0.022	0.776	**0.024**	1.012	**0.009**	1.012	0.008	**1.027**	0.008
07	**3.233**	**0.208**	0.192	0.007	1.306	**0.050**	**3.685**	0.015	0.798	**0.014**	0.797	0.014	**0.97**	0.013
08	0.267	**0.022**	**0.704**	0.003	0.180	**0.033**	**2.198**	0.013	**0.338**	**0.008**	0.338	0.008	0.338	0.008
09	**—**	**—**	1.203	0.01	**1.842**	**0.035**	0.500	0.008	0.33	0.007	0.329	0.007	**0.332**	**0.008**
10	**—**	**—**	0.104	0.024	1.275	0.018	**3.875**	**0.058**	0.157	0.004	**0.174**	**0.005**	0.157	0.004

TABLE II: Average absolute discrepancy $\mathbb{E}(RRE)$ and $\mathbb{E}(RTE)$ for Sequences $00-10$ in SemanticKITTI, where $RRE[°]$ and $RTE[10^{-2} \times \text{m}]$, after masking the input adjacency matrix (up) and the attention weights from the last layer of SEM-GAT (down). The colors indicate highest discrepancy scores after perturbation with **red** indicating highest scores overall, **blue** highest scores after semantic masking, and **purple** highest scores for each individual semantic class. For multi-class masking,the symbol "**—**" indicates error in pose estimation due to insufficient registration points or confidence loss.

B. Attention-Accuracy Correlation

Following the analysis in Sec. VI-A, the initial results suggest correlation between attention and model performance.

To investigate this further, we retrieve the average absolute discrepancy scores $\mathbb{E}(RRE)$ and $\mathbb{E}(RTE)$ as well as the total $\mathbb{E}$ for every sequence and correlate it with the JSD results. This process is done for both parts of the perturbation process described Sec. IV. We are mainly interested in the *Single-Class* set because the number of points masked is very low compared to the corresponding JSD, suggesting that the average absolute discrepancy $\mathbb{E}$ increase is due to the attention weights masking and not the downsampling of the pointcloud. For each sequence, we compare the results in Tab. II and **??** and correlate them with Fig. 4. For most sequences, the ranking of $\mathbb{E}$ scores isproportional with the ranking in JSD scores. These results indicate that the semantic sets with the highest JSD are the most important for SEM-GAT. This analysis provides a good first insight into the validity of using attention as an indicator of semantic importance. Further experiments are necessary to verify the accuracy and robustness of our approach.

VII. Conclusions

In this work, we investigated the semantic interpretation of attention scores for identifying key semantic elements in a pointcloud and introduced a methodology to evaluate the validity of our findings. Our initial results indicate that attention can be used to estimate the importance of semantic features with respect to their contribution to the output of a baseline GNN model. Additional experiments are essential to verify the fidelity of the method. This work contributes towards identifying the environmental elements that are important for a graph-based pose estimation model. This methodology can be used to explain the model's performance in correlation with the semantics present.

References

[1] B. Wu, Y. Bian, H. Zhang, J. Li, J. Yu, L. Chen, C. Chen, and J. Huang, "Trustworthy Graph Learning: Reliability, Explainability, and Privacy Protection," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 4838–4839, 2022.

[2] H. Zhang, B. Wu, X. Yuan, S. Pan, H. Tong, and J. Pei, "Trustworthy Graph Neural Networks: Aspects, Methods and Trends," pp. 1–36, 2022.

[3] Y. Fan, Y. Yao, and C. Joe-Wong, "GCN-SE: Attention as Explainability for Node Classification in Dynamic Graphs," *Proceedings - IEEE International Conference on Data Mining, ICDM*, vol. 2021-Decem, pp. 1060–1065, 2021.

[4] S. Wiegreffe and Y. Pinter, "Attention is not not explanation," *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp. 11–20, 2019.

[5] S. Jain and B. C. Wallace, "Attention is not explanation," 2019.

[6] S. Serrano and N. A. Smith, "Is attention interpretable?," *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pp. 2931–2951, 2020.

[7] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012.

seq	Average Absolute Discrepancy $\mathbb{E}$: Adjacency Matrix Masking						
	Top 5 Classes	Top 3 Classes	Surfaces	Corners	1st Class	2nd Class	3rd Class
00	**4.932**	1.019	3.807	**5.214**	1.99	1.998	**2.017**
01	**—**	**—**	8.688	**9.162**	0.128	0.115	**0.162**
02	**—**	**—**	**9.316**	2.043	**0.61**	0.589	0.585
03	**—**	1.888	3.629	**9.378**	0.339	0.311	**0.401**
04	**—**	2.424	5.538	**7.228**	0.945	**1.034**	0.991
05	**7.64**	2.592	2.698	**6.970**	0.687	0.711	**0.756**
06	**2.308**	0.903	**4.196**	3.734	1.865	**1.867**	1.844
07	**24.263**	0.837	5.642	**6.305**	2.099	2.108	**2.281**
08	**1.929**	0.374	3.329	**4.162**	1.169	**1.183**	1.166
09	**—**	2.22	**5.211**	1.506	1.057	**1.067**	1.057
10	**—**	2.314	3.010	**10.656**	0.59	**0.696**	0.56

seq	Average Absolute Discrepancy $\mathbb{E}$: Attention Weights Masking						
	Top 5 Classes	Top 3 Classes	Surfaces	Corners	1st Class	2nd Class	3rd Class
00	**4.954**	0.989	3.969	**4.653**	1.98	1.967	**1.99**
01	**—**	**—**	**9.481**	8.198	**0.149**	0.113	0.143
02	**—**	**—**	**9.826**	1.907	0.593	0.591	**0.605**
03	**—**	1.861	4.211	**8.047**	0.351	**0.364**	0.331
04	**—**	2.502	**5.884**	5.743	**0.948**	0.925	0.855
05	**7.605**	2.556	2.916	**5.935**	0.684	0.695	**0.733**
06	**2.362**	0.872	**4.776**	3.187	**1.911**	1.854	1.873
07	**24.082**	0.859	**6.280**	5.158	2.170	2.176	**2.302**
08	**1.926**	0.384	3.493	**3.504**	1.15	1.141	**1.166**
09	**—**	2.2	**5.375**	1.268	1.054	1.055	**1.086**
10	**—**	2.295	3.088	**9.666**	0.576	**0.685**	0.559

TABLE III: Total average absolute discrepancy $\mathbb{E} \times 10^{-2}$ across Sequences 00-10 in SemanticKITTI after masking the adjacency matrix from the input graphs (up) and masking the attention weights at the last layer of SEM-GAT (down). Similar to Tab. II, the colors indicate highest discrepancy scores after perturbation with **red** indicating highest scores overall, **blue** highest scores after semantic masking, and **purple** highest scores for each individual semantic class. For multi-class masking,the symbol ”**—**” indicates error in pose estimation due to insufficient registration points or confidence loss.

[8] H. Yuan, H. Yu, S. Gui, and S. Ji, “Explainability in Graph Neural Networks: A Taxonomic Survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–19, 2022.

[9] F. Baldassarre and H. Azizpour, “Explainability techniques for graph convolutional networks,” 2019.

[10] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, “Explainability methods for graph convolutional neural networks,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10764–10773, 2019.

[11] R. Schwarzenberg, M. Hübner, D. Harbecke, C. Alt, and L. Hennig, “Layerwise relevance visualization in convolutional text graph classifiers,” 2019.

[12] T. Schnake, O. Eberle, J. Lederer, S. Nakajima, K. T. Schutt, K.-R. Muller, and G. Montavon, “Higher-order explanations of graph neural networks via relevant walks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 7581–7596, nov 2022.

[13] Q. Huang, M. Yamada, Y. Tian, D. Singh, D. Yin, and Y. Chang, “Graphlime: Local interpretable model explanations for graph neural networks,” 2020.

[14] Y. Zhang, D. Defazio, and A. Ramesh, “Relex: A model-agnostic relational model explainer,” 2020.

[15] M. N. Vu and M. T. Thai, “Pgm-explainer: Probabilistic graphical model explanations for graph neural networks,” 2020.

[16] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, “GNNExplainer: Generating explanations for graph neural networks,” *Advances in Neural Information Processing Systems*, vol. 32, no. i, pp. 9240–9251, 2019.

[17] D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang, “Parameterized explainer for graph neural network,” 2020.

[18] T. Funke, M. Khosla, M. Rathee, and A. Anand, “Zorro: Valid, sparse, and stable explanations in graph neural networks,” 2022.

[19] M. S. Schlichtkrull, N. D. Cao, and I. Titov, “Interpreting graph neural networks for nlp with differentiable edge masking,” 2022.

[20] X. Wang, Y. Wu, A. Zhang, X. He, and T. seng Chua, “Causal screening to interpret graph neural networks,” 2021.

[21] H. Yuan, H. Yu, J. Wang, K. Li, and S. Ji, “On explainability of graph neural networks via subgraph explorations,” in *Proceedings of the 38th International Conference on Machine Learning* (M. Meila and T. Zhang, eds.), vol. 139 of *Proceedings of Machine Learning Research*, pp. 12241–12252, PMLR, 18–24 Jul 2021.

[22] D. Xu, W. Cheng, D. Luo, X. Liu, and X. Zhang, “Spatio-temporal attentive RNN for node classification in temporal attributed graphs,” *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2019-Augus, pp. 3947–3953, 2019.

[23] X. Zuo, T. Jia, X. He, B. Yang, and Y. Wang, “Exploiting Dual-Attention Networks for Explainable Recommendation in Heterogeneous Information Networks,” *Entropy*, vol. 24, no. 12, pp. 1–19, 2022.

[24] B. Rath, X. Morales, and J. Srivastava, “SCARLET: Explainable Attention Based Graph Neural Network for Fake News Spreader Prediction,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12712 LNAI, pp. 714–727, 2021.

[25] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, “SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-October, no. iii, pp. 9296–9306, 2019.