Intrinsic Subgraph Generation for Interpretable Graph based Visual Question Answering

Pascal Tilli, Ngoc Thang Vu
Institute for Natural Language Processing (IMS)
University of Stuttgart
Stuttgart, Germany
{pascal.tilli, thang.vu}@ims.uni-stuttgart.de

Abstract

The large success of deep learning based methods in Visual Question Answering (VQA) has concurrently increased the demand for explainable methods. Most methods in Explainable Artificial Intelligence (XAI) focus on generating post-hoc explanations rather than taking an intrinsic approach, the latter characterizing an interpretable model. In this work, we introduce an interpretable approach for graph-based VQA and demonstrate competitive performance on the GQA dataset. This approach bridges the gap between interpretability and performance. Our model is designed to intrinsically produce a subgraph during the question-answering process as its explanation, providing insight into the decision making. To evaluate the quality of these generated subgraphs, we compare them against established post-hoc explainability methods for graph neural networks, and perform a human evaluation. Moreover, we present quantitative metrics that correlate with the evaluations of human assessors, acting as automatic metrics for the generated explanatory subgraphs. Our implementation is available at https://github.com/DigitalPhonetics/Intrinsic-Subgraph-Generation-for-VQA.

Keywords: Interpretability, Explainability, XAI, Graph based VQA, Subgraphs, GNNs, I-MLE

1. Introduction

Visual Question Answering (VQA) (Antol et al., 2015; Shih et al., 2016) is acknowledged as a challenging multi-modal task for Machine Learning (ML) algorithms as it requires a semantic comprehension of images in relation to the posed questions. State-of-the-art approaches for VQA are systems based on Deep Learning (DL), which are mostly assessed by accuracy and efficiency metrics. To enhance collaboration between humans and ML systems in real-world settings, it is essential to reliably comprehend the system's outputs. Nonetheless, the majority of DL based VQA systems are considered black boxes by both users and developers. Hence, deploying these systems in important decision-making domains carries considerable risk, despite their progress.

The domain of Explainable Artificial Intelligence (XAI) addresses the aforementioned issue, and can be categorized into two subdomains: *explainability* and *interpretability* (Rudin, 2019; Marcinkevičs and Vogt, 2020). The domain of interpretability centers on inherently interpretable models, i.e. where the decision making process of the models can be comprehended by humans, e.g. decision trees. Explainable ML focuses on methods that generate explanations post-hoc for already existing (black-box) models, possibly requiring additional hyperparameter tuning. Notably, interpretable models possess the advantage of intrinsic explanation generation, where the model itself generates explanations. This contrasts with post-hoc methods, which introduce an additional method or model aimed at explaining predictions, leading to increased computational cost.

Explainability methods for VQA often focus on pixel importance as visual explanations (Arras et al., 2022; Panesar et al., 2022). Some approaches address explainability by generating rationales to explain the system's predicted answers. These rationales can be generated either post-hoc by a another neural network or by the original network itself (Schwenk et al., 2022). However, it remains uncertain whether the answer and rationale generation processes influence one another. An alternative strategy involves formulating contrastive explanations (Arras et al., 2022). Moreover, some models that yield intermediate outputs are considered to offer interpretability (Caro-Martinez et al., 2023). For instance, in (Fu et al., 2023), the system translates images into textual descriptions relevant to the given question, uses these to predict the answer and is thereby interpretable. These methods purport to offer interpretability, yet they do not align with our definition wherein an interpretable model should intrinsically generate its own explanation.

In this work, we introduce an approach for graph based VQA that employs a structured graph representation of the displayed scene instead of the raw image input. The primary goal of our work, is to generate a subgraph alongside the model's prediction as explanation, highlighting the most relevant nodes for a given question, as opposed to employing post-hoc explanation methods.

We focus on the following research questions:

LREC-COLING 2024, pages 9204–9223
20-25 May, 2024.

RQ1 How can we increase the interpretability of deep learning-based VQA answer prediction through the utilization of Graph Neural Networks?

RQ2 How does the quality of explanations generated by our method compare to that of state-of-the-art post-hoc explanation methods when evaluated by human assessors?

RQ3 What methods can we employ to quantitatively assess the quality of explanations in cases where no ground-truth references are available, and to what extent do these quantitative measures align with human preferences?

To address these research questions, we propose a system featuring a Graph Attention Network (GAT) at its core component, which is able to extract a subgraph as explanation for the prediction. We validate our approach in a graph based VQA setting using GQA (Hudson and Manning, 2019b). Furthermore, we conduct a human evaluation to compare our internally generated subgraph with explanations generated by post-hoc methods. Additionally, we introduce evaluation metrics tailored to subgraphs used as explanations in scenarios where ground-truth explanations are unavailable.

Our contributions can be summarized as follows:

1. We propose a novel VQA system that not only provides answers but also offers relevant explanations. Our approach has been proven to deliver highly accurate results, and human evaluators have shown a preference for our intrinsic explanations over traditional post-hoc explainability methods.

2. We introduce quantitative metrics, which correlate with the results of the human evaluation, to measure the quality of explanations.

2. Related Work

2.1. Graph Neural Networks

Graph Neural Networks (GNNs) are evolving to an increasingly more popular area of research due to their recent successes (Xie et al., 2022; Dai et al., 2022). They are designed to harness graph-structured data and perform message passing among nodes to learn contextualized node embeddings. In addition to their natural applicability in domains where it is obvious to represent data as graphs, such as chemistry (where molecules can be modeled as graphs) or e-commerce (where users interactions can be represented as a graph) (Dai et al., 2022), GNNs have been successfully applied in Natural Language Processing (NLP) tasks (Wu et al., 2023) and Computer Vision (CV), including tasks like Scene Graph Generation (SGG) (Dai et al., 2022). For a comprehensive overview spanning various domains, readers are directed to Dai et al. (2022) regarding a general survey on GNNs. For in-depth exploration with a focus on NLP Wu et al. (2023) offer an extensive resource.

2.2. Visual Question Answering

Many different datasets (Johnson et al., 2017; Hudson and Manning, 2019b; Agrawal et al., 2018; Singh et al., 2019) and models have been published since the task itself has been introduced, and aim to evaluate divers capabilities of models (Väth et al., 2021; Lin et al., 2023; Shao et al., 2023). Given the graph-based nature of our VQA approach, the subsequent section will be dedicated exclusively to this aspect.

2.3. Graph based VQA Models

Graph based VQA represents a specialized variant of VQA where models use intermediate graph structures that represent the scenes in images, which enables the usage of powerful GNNs for the task. Hudson and Manning (2019a) proposed a Neural State Machine (NSM), which incorporated question guided traversal of scene graphs. This approach treats nodes as states and allows transitions along edges (relations) in the graph. Subsequently, Liang et al. (2021) extended this notion by employing instruction vectors to guide the information propagation of a Graph Neural Network (GNN). The authors conducted comparisons utilizing ground-truth scene graphs sourced from the GQA dataset (Hudson and Manning, 2019b). Remarkably, the GAT (Veličković et al., 2017; Brody et al., 2021) achieved the best performance by a substantial margin. In the work of Koner et al. (2021) a Reinforcement Learning (RL) based approach was developed, treating VQA as a graph traversal problem. Their reported performance on ground-truth scene graphs within the GQA dataset slightly trailed behind the GAT approach by Liang et al. (2021).

Li et al. (2022) proposed an approach that centers on a SGG model, which transforms the scene graph into two graphs. One of these graphs emphasizes objects, while the other focuses on the relational aspects. Wang et al. (2022) is characterized by a bidirectional fusion process between unstructured and structured multimodal knowledge to obtain unified knowledge representation. This fusion ultimately yields a unified knowledge representation. Both have a different focus compared to our approach prioritizing the SGG task, particularly in terms of the methods applied for crafting the scene graph.

The interpretability or explainability of such systems is often asserted without being tested, i.e. the human for whom the explanation is intended is left

out of the evaluation loop. Zhu (2022) construct three types of graphs during the reasoning process to generate an answer. Although the model is described as interpretable because humans can view the intermediate graph representations, there is no actual assessment of its interpretability conducted by human evaluators. Similarly, Sarkisyan et al. (2022) assert that they construct an interpretable model by mapping questions into a graph structures which is supposed to make it interpretable. However, no tests are carried out to assess this aspect of interpretability.

Chen et al. (2021) introduces an approach for knowledge-based VQA, where masking strategies are applied to predict answers to better generalize to out-of-distribution answers. In contrast, our approach employs a differentiable hard-attention masking to extract a subgraph during the message-propagation of the GNN.

2.4. Explainability and Interpretability

Interpretability and explainability have garnered increased attention alongside the successes of ML methods across diverse domains. Linardatos et al. (2020) conducted a review focusing on XAI and interpretable ML methods. In addition to their comprehensive literature review, they introduced a systematic taxonomy of these approaches, coupled with references to their programming implementations.

Liu et al. (2022) reviewed the interpretability in GNNs. Notably, most approaches for GNNs are post-hoc methods that try to explain predictions of black-box models. Transparent approaches in this context are relatively rare. DL methods are recognized as interpretable or transparent that predominantly rely on soft attention scores. Efforts have been dedicated to investigate if soft attention can be considered interpretable, yielding findings that challenge their alignment with true interpretability (Serrano and Smith, 2019).

Feng et al. (2022) proposed Kernel Graph Neural Networks (KerGNNs) that integrate kernels into the message passing process of GNNs in order to overcome the limitation of standard GNNs which are not capable of surpassing the performance of the Weisfeiler-Lehman (WL) algorithm in a graph isomorphism test.

Drawing inspiration from the idea of visualizing filters of Convolutional Neural Networks (CNNs) (Krizhevsky et al., 2012) to identify important feature regions, they visualize the trained kernel based filters to detect key structures in the input and thus referring to their method being interpretable.

2.5. Soft- and Hard-Attention

Soft-attention (Bahdanau et al., 2014) is generally defined by a continuous variable, while hard-attention is defined by a discrete variable. Consequently, we can employ gradient-descent methods to differentiate soft-attention, but we cannot utilize these methods for hard-attention due to the non-differentiable nature of a discrete step. In contrast, hard-attention operates with discrete values, which is typically implemented using sampling methods such as the Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2017) or RL techniques.

In terms of interpretability, hard attention's binary nature makes results clearer and reduces ambiguity about important data parts.

Niepert et al. (2021) proposed a framework for backpropagating through such a discrete sampling step. Their method can be described as a general-purpose algorithm for hybrid learning of systems that contain discrete components embedded in a computational graph. In our specific application, we leverage Implicit Maximum Likelihood Estimation (I-MLE) to approximate the gradients for a discrete *top-k* sampling procedure of nodes in a scene graph.

3. Problem Formulation

Our problem is framed as a graph representation learning challenge within the multi-modal domain of VQA. Typically, VQA is characterized by the function $f(q, i) \rightarrow a$, where q represents a question, i corresponds to an image, and a signifies the answer (treated as a classification problem, with $a \in \mathbb{R}^{n_a}$, where n_a denotes the predefined number of potential answers).

In our case, we depart from the conventional image i and instead employ a graph g as the visual representation, characterized as: $f(q, g) \rightarrow a$. The question is represented as a sequence of tokens $q \in \mathbb{R}^{n_q \times d_q}$, where n_q signifies the number of tokens and d_q the dimensionality of the vector representing the token (we use $300d$ GloVe embeddings (Pennington et al., 2014)).

A graph is defined as $G = (V, E)$, where $V :=$ is the set of nodes and $E :=$ the set of edges. Each node $v_i \in V$, is associated with a natural language description, and we initialize them using GloVe embeddings, leading to $v_i \in \mathbb{R}^{d_v}$, with $d_v = 300$. Similarly, we encode the edge information, with the identity of $e_{i,j}$ defined by its natural language token, represented as $e_{i,j} \in \mathbb{R}^{d_e}$, with $d_e = 300$ as well.

The primary aim of our interpretable approach is to intrinsically generate a subgraph S_g as an explanation, effectively identifying the most salient nodes within G as $V(S_g) \subset V(G)$. A prediction is defined as $\hat{y} = f(x, x_e, q)$, and our model implicitly

identifies the subgraph S_g from the node features x, edge features x_e, and question features q. We implement this through a trainable hard-attention node mask.

4. Proposed Model Architecture

We present the fundamental components of our approach in Figure 1 and Algorithm 1, providing an overview of the sub-modules within our system and the algorithmic procedures involved. At the center of our model, we employ a GAT, aimed to construct an increasingly inherently interpretable system. This GAT learns to intrinsically generate a relevant subgraph from the input graph that contains nodes, which are most relevant for the answer finding process. The prediction process exclusively utilizes nodes within the extracted subgraphs.

In Algorithm 1, the variable $X \in \mathbb{R}^{n \times d}$ denotes the node embedding matrix. Correspondingly, $X_e \in \mathbb{R}^{m \times d}$ represents the edge embedding matrix. A denotes the adjacency matrix, containing the interconnections between node pairs. Furthermore, Q defines the sequence of tokens in the questions.

Algorithm 1 Our model pipeline

Require: Input Data: X, X_e, A, Q
Encode Question:
  $Q_{\text{enc}} \leftarrow$ Transformer Encoder(Q)
Decode Question into Instruction Vectors:
  $I \leftarrow$ Transformer Decoder(Q_{enc})
Global Question Representation:
  $Q_{\text{global}} \leftarrow$ MLP(I)
Encode Scene Graph:
  $X', X'_e \leftarrow$ Scene Graph Encoder(X, X_e, A)
MGAT, outputs Mask for Subgraph:
  X', mask $\leftarrow$ MGAT(X', X'_e, A, I)
Apply Hard-Attention Mask:
  $X' \leftarrow X' \times$ mask
Global Graph Representation Vector:
  $X_{\text{g}} \leftarrow$ Global Attention(X', Q_{global})
Final Answer Prediction:
  logits $\leftarrow$ MLP$(X_{\text{g}}, Q_{\text{global}}, X_{\text{g}} \times Q_{\text{global}})$
return Model Output: logits, mask

Question Processing We process the questions Q with an encoder-decoder architecture. Both the encoder and decoder are transformer-based models (Vaswani et al., 2017), randomly initialized and not fine-tuned versions of pretrained transformer models. The token sequence Q get transferred into their respective vector representations through an Embedding layer, constructed from the dataset vocabulary, within the transformer encoder. We utilize Pennington et al. (2014) vectors as initial vector representations of the Embedding layer. The transformer decoder takes the encoded sequence Q_{enc} as input and generates a fixed-length sequence, denoted as instruction vectors, similar to the approach by Liang et al. (2021). These instruction vectors, represented as I, distribute information from the textual modality (questions) to the visual modality (scene graphs representing image scenes). The instruction vectors I are flattened and projected in the hidden dimensionality using a Multi-Layer Perceptron (MLP) to produce the global question vector Q_{global}. This global question vector plays an important role in the final answer token prediction and guides the global attention aggregation following the GNN processing step. The number of instructions corresponds directly to the number of layers we use in our GNN.

Scene Graph Encoding The scene graph encoding module functions as an interface, connecting the graph to the GNN processing it. This encoder translates node identities (e.g., *building*) and up to three corresponding attributes (e.g., *large*, *grey*, *square*) into vector representatins using 300-dimensional GloVe vectors, akin to the initialization of question tokens. The node representation is the summation of these vectors. In Algorithm 1, the node's identity and attributes are contained in X, while edge (relation) information in encoded in X_e. The bounding-box coordinates are encoded using a MLP. Subsequently, the bounding-box vector and node representation are concatenated and projected back into the defined hidden dimensionality.

To obtain the final representations, we process both the node embeddings X and edge embeddings X_e through an MLP to obtain X' and X'_e. In the case of edge embeddings, we additionally concatenate the connected nodes prior to the MLP.

Masking Graph Attention Network Our proposed approach aims to constrain the model's usage of a subset of nodes of the original input graph, referred to as *subgraph*, for making predictions. This subgraph should be the only information that the model has access to when generating answers for given questions. Hence, we want to discretely sample from the input graph $\mathcal{G}$, which is typically not easily differentiable using gradient-based backpropagation methods. To differentiate the aforementioned sampling step, we estimate the gradients with I-MLE (Niepert et al., 2021). In our implementation, we introduce an additional hard attention mask consisting of zeros and ones for nodes, i.e. a binary mask. We extent the GAT and refer to it as Masking Graph Attention Network (M-GAT).

The hard attention mask is computed during message-passing in a M-GAT layer. For each node x_i within the input graph $\mathcal{G}$, we compute a score s_i

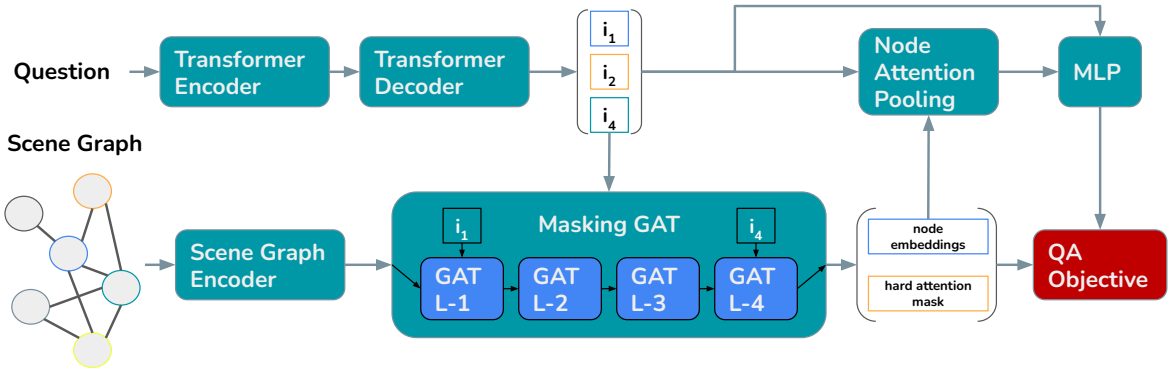


Figure 1: High-level architecture of our model. The hard attention mask gets computed in GAT-layer four. Only the $top-k$ node embeddings within the hard attention mask are passed in the node attention pooling module.

using the scoring function:

$$s_i = \sigma\left(\frac{X' I^T}{\sqrt{d_x}}\right). \tag{1}$$

This function calculates the scaled dot-product between node embeddings X' and the instruction vector I, and d_x representing the dimensionality of X' and I. The purpose of this step is to allow the model to learn an importance score for each node. To obtain our γ_i, we map the scores s_i to zero or one, which is typically non-differentiable but by incorporating I-MLE we can backpropagate through this discrete step. The idea is to expand the mask on a node level to get a mask on an edge level of the graph $\mathcal{G}$. For each $\alpha_{i,j}$ in a GAT we want to get a corresponding $\gamma_{i,j}$, which is

$$\gamma_{i,j} = \begin{cases} 0, & \text{otherwise} \\ 1, & \text{if } s_i * s_j = 1 \end{cases} \tag{2}$$

During message passing, we zero out edges and nodes that are not part of our mask. The new update rule is defined as:

$$\mathbf{x}'_i = \alpha_{i,i}\mathbf{\Theta}\mathbf{x}_i + \sum_{j \in \mathcal{N}(i)} \mathbf{\Theta}\mathbf{x}_j \alpha_{i,j}\gamma_{i,j}, \tag{3}$$

Here, $\mathcal{N}(i)$ denotes all neighboring nodes $\mathbf{x}_j$ of node $\mathbf{x}_i$. The computation of $\alpha_{i,j}$ remains unchanged. It is worth noting that, in our experiments, the hard attention mask and its effect on message-passing are only applied in the final layer.

During backpropagation, we aggregate gradients received on edge level via summation or multiplication to estimate the gradients of our mask at a node level.

To enhance the flow of information between GNN layers in our M-GAT, we employ residual connections (He et al., 2016) with a forget-gate mechanism, defined as $l(x) = f(x) + x$.

Global Attention Aggregation & Question Answering The node embeddings X' of the M-GAT (line none to twelve in Algorithm 1) are aggregated form a unified graph embedding vector, employing a global attention mechanism driven by the question vector Q_{global}. This Q_{global}, derived from the instruction vectors, performs a scaled dot-product with X' to obtain α_i scores, which are normalized using *softmax*. These α_i scores guide the weighted summation, resulting in a single graph embedding vector X_g. To perform question answering, we take the graph embedding vector X_g, the Q_{global} representation, and the the Hadamard product of both, concatenate them before feeding them through a MLP to receive our final answer token logits. We optimize the system using a cross-entropy loss and Adam-W (Loshchilov and Hutter, 2018).

5. Evaluation Methods

Due to the unavailability of ground-truth explanations, we employ post-hoc XAI methods for graph neural networks as a benchmark for our intrinsically generated subgraphs. To assess the explanations, we (i) perform a human evaluation as qualitative analysis, and (ii) introduce metrics for quantitative assessment.

5.1. Baselines for Comparison

We included *Integrated Gradients* (Sundararajan et al., 2017) a gradient-based method and popular XAI method, and *PGMExplainer* (Vu and Thai, 2020) and *GNNExplainer* (Ying et al., 2019) as perturbation-based methods. Additionally, we include a Random explainer as a baseline. We leverage the implementations of Agarwal et al. (2023), except for the GNNExplainer[1].

[1]The implementation of Fey and Lenssen (2019) offered more flexibility, leading to improved results.

GNNExplainer learns a soft-mask applied the nodes. To ensure consistency with the number of nodes in the subgraph, we utilize the same $topK$ value that we computed for our mask.

5.2. Human Evaluation

In our study, we include four explainability methods: (1) our intrinsic subgraph, (2) GNNExplainer due to its notable performance in quantitative metrics (cf. Section 6.2.3 and Section 6.2.4), (3) Integrated Gradients as gradient based method, and (4) the random explainer serving as a baseline. PGMExplainer was excluded due to GNNExplainer performing better, and both being pertubation-based methods (cf. Section 6).

To perform pairwise comparisons among these methods, we conducted a total of six comparisons that span all four techniques. Participants were presented with 18 randomly selected graph-question pairs. It's worth mentioning that the image was additionally displayed solely as a reference, as the model does not utilize the image as input. Notably, each possible method pair occurred exactly three times. For each graph, two subgraph visualizations from different explainability methods were provided as the corresponding explanations given the question and graph. Participants were tasked with selecting their preferred explanation, or they could opt for one of two additional choices: *equally good* or *equally bad*. The latter was to be chosen when none of the explanations was deemed suitable for the given graph-question pair, while the former represented a valid choice when both explanations were deemed acceptable, with no preference between them.

To (i) minimize potential psychological biases (instances where users refrain from selecting "equally good" because they deem both explanations unsatisfactory) and (ii) collect a more comprehensive dataset, we split the *equal* option into *equally good* and *equally bad*. We randomize the order and orientation of the comparisons, enabling a robust evaluation of user preferences.

We choose the Bradley-Terry model (Bradley and Terry, 1952; Bradley and El-Helbawy, 1976) to analyze the results, and we treated *equally good* and *equally bad* as ties with a score of 0.5, given their identical nature in pairwise comparisons.

5.3. Metrics for Subgraphs

Answer and Question Token Co-occurences To get an estimate how well the subgraphs capture the information given by both the question and its corresponding answer, we conduct a token co-occurrences analysis. We expect the subgraphs to include objects that are mentioned in the question, as well as the answer token (if the answer is indeed an object), to appear more frequently in the explanation subgraph. Otherwise, the model's prediction seems implausible.

When the ground-truth answer is an object that can occur in the scene graph, we count the occurrences of the respective token within the subgraph and report the resulting percentage. Likewise, we calculate the percentage of question tokens present in the subgraph.

Removing Subgraphs To measure the nodes' importance for predictions, we suspect the nodes included in our explanation subgraph to be crucial. Accordingly, we aim to *remove* the subgraph and passing the remaining graph through the network to measure answer accuracy once more.

To *remove* a subgraph we do not alter the structural integrity of the scene graph. Instead, we randomize the node embeddings, i.e. for each $x_i \in X$, where X is the matrix of node embeddings, we randomize x_i if γ_i equals 1.

Furthermore, the edge embeddings are also randomized. Specifically, each $e_{i,j}$ between nodes x_i and x_j also provides no information, if $\gamma_{i,j}$ equals 1. This process ensures that the message-passing paradigm among nodes remains undisturbed, enabling it to occur to the same extent as in the original graph.

If the system remains capable of answering questions when information from the nodes within the subgraph is absent, two potential interpretations arise: (i) either the identified subgraph may not be as relevant, or (ii) the model could be exploiting biases present in the question-graph pair.

6. Experimental Setup and Results

6.1. Setup

Resources We conduct experiments[2] on the GQA[3] dataset (Hudson and Manning, 2019b) since it is designed to test a model's real-world reasoning capabilities. It provides ground-truth scene graphs, which enable *perfect sight* using graph-structured representations of images.

Due to the unavailability of the scene graphs for the test sets of GQA, we report the performance on the validation set. This practice aligns with common conventions in the field, as observed in previous works such as Koner et al. (2021) and Liang et al. (2021). Each question type in GQA corresponds to a specific *structural question* type as well as a *semantic question* type. The *structural question* types are: **Verify**: yes/no questions. **Query**: open-ended questions. **Choose**: questions offer two

[2]Implementation details are in the Appendix A.2

[3]Additional information is in the Appendix A.1.

Method	All	Attr	Rel	Obj	Global	Cat	Query	Verify	Choose	Logical	Compare
GNNExplainer	0.35	**0.41**	**0.39**	0.18	0.5	**0.99**	**0.48**	0.26	0.36	0.26	**0.41**
Int. Grad.	0.09	0.13	0.13	0.18	0.0	0.01	0.10	0.24	0.14	0.14	0.0
Random	0.04	0.12	0.10	0.12	0.0	0.0	0.10	0.13	0.12	0.07	0.26
Ours	**0.52**	0.35	0.38	**0.52**	0.5	0.0	0.32	**0.37**	**0.38**	**0.53**	0.33

Table 1: Results of the Bradley-Terry model. Each real valued p_i score for the corresponding explainability method is displayed column-wise. *Int. Grad.* abbreviates Integrated Gradients.

alternatives to choose from. **Logical**: involve logical inference. **Compare**: comparison among two or more objects in the scene. The *semantic question* types include: **Object**: existence questions. **Attribute**: questions about properties or position of an object. **Category**: object identification within classes. **Relation**: questions asking about the subject or object of an relation. **Global**: questions about general properties of a scene.

6.2. Results

6.2.1. RQ1: Question Answering Performance

In terms of question-answering performance, as indicated by answer token accuracy, presented in Table 2, our approach yields competitive results. It slightly surpasses the GAT model of Liang et al. (2021) (94.79% compared to 94.78% accuracy). Table 2 additionally contains several *topK%* configurations. It is important to note that the hard attention masks are exclusively computed and applied within the final layer. The first three layers learn contextualized embeddings for all nodes. Only in the last layer do we introduce constraints on the message-passing to learn the hard attention mask (as indicated by the values in the *Masks* column, with a value of 1.0 for the first three layers) [4] We use the top-performing model in the remaining experiments.

Model	Masks	QA-%
GAT	-	94.78
Graphhopper	-	92.30
Ours$_{k\%=.50}$	$(1.0, 1.0, 1.0, 0.50)$	93.20
Ours$_{k\%=.30}$	$(1.0, 1.0, 1.0, 0.30)$	94.21
Ours$_{k\%=.25}$	$(1.0, 1.0, 1.0, 0.25)$	94.72
Ours$_{k\%=.20}$	$(1.0, 1.0, 1.0, 0.20)$	94.15
Ours$_{k\%=.15}$	$(1.0, 1.0, 1.0, 0.15)$	**94.79**
Ours$_{k\%=.10}$	$(1.0, 1.0, 1.0, 0.10)$	77.88

Table 2: Question answering performance. GAT by Liang et al. (2021) and Graphhopper by Koner et al. (2021).

6.2.2. RQ2: Human Evaluation

In our study, we gathered data from 16 participants aged between 20 and 59, resulting in a total of 288 data points[5]. Utilizing the Bradley-Terry model (Bradley and Terry, 1952; Bradley and El-Helbawy, 1976), we established a probabilistic model to ascertain human preferences for various explainability methods, subsequently producing a ranking. The derived p_i scores are positive real values associated with the *i-th* explainability method. These outcomes were determined both on an overall scale and specifically for each question type. The detailed p_i scores can be observed in Table 1. In this context, the scores serve as an interpretation of the relative preferences of humans.

6.2.3. RQ3: Token Co-occurrences

Answer Token Co-occurrence (AT-COO) The co-occurrence rates between answer tokens and graph nodes are presented in Table 3. We utilized a random subset of 10% of the evaluation data to compute the results in Table 3 and Table 4, due to the intensive computational costs of all explainability methods. For our method, in 75.15% of cases, the intrinsic subgraph captures the node aligned with the answer token. Notably, the GNNExplainer outperforms with a coverage of 89.12%, suggesting a higher precision in encompassing the answer token node. Conversely, other methods display diminished focus on the node associated with the answer token.

Method	AT-COO
Ours$_{k\%=.15}$	75.15
Random$_{k\%=.15}$	30.59
GNNExplainer$_{k\%=.15}$	89.12
PGMExplainer$_{k\%=.15}$	22.37
Integrated Gradients$_{k\%=.15}$	8.14

Table 3: AT-COO, represented as a percentage of potential matches.

[4] Please refer to Appendix A.4, Table 7 for a comprehensive overview of results.

[5] Additional information is available in the Appendix A.3.

Question Token Co-occurrence (QT-COO) Table 4 presents the co-occurrence results for question tokens and their alignment with graph nodes. The subgraph generated by our method contains 78.35% of the feasible question token matches. Other methods show a reduced frequency of including the respective question token nodes in their explanatory subgraphs. Noteworthy among these are GNNExplainer and Integrated Gradients, with inclusion rates of 59.67% and 39.95%, respectively.

Method	QT-COO
Ours$_{k\%=.15}$	78.35
Random$_{k\%=.15}$	29.79
GNNExplainer$_{k\%=.15}$	59.67
PGMExplainer$_{k\%=.15}$	24.67
Integrated Gradients$_{k\%=.15}$	39.95

Table 4: QT-COO, represented as a percentage of potential matches.

6.2.4. RQ3: Removing Subgraphs

Table 5 presents the question-answering accuracy when removing the explanatory subgraph, based on the same 10% of the evaluation data, as in Section 6.2.3. A larger reduction in accuracy implies the importance of the subgraph in capturing essential graph nodes for the given question. Notably, the highest degradations were observed for Integrated Gradients, GNNExplainer, and our method. In contrast, PGMExplainer and Random resulted in lesser performance reductions, indicating their generated subgraphs were less important than those of the other methods. It is worth noting, that only our approach, our variant of GNNExplainer (refer to Section 6.1), and PGMExplainer are restricted to leveraging just 15% of the input graph.

Method	QA-%
Ours$_{k\%=.15}$	37.13
Random$_{k\%=.15}$	52.10
GNNExplainer$_{k\%=.15}$	33.28
PGMExplainer$_{k\%=.15}$	69.46
Integrated Gradients$_{k\%=.15}$	33.28

Table 5: Question answering performance after randomizing the important nodes of the respective methods.

7. Analysis

RQ1: Intrinsically Interpretable GNN To answer the first research question, we incorporated a discrete sampling method into the message passing mechanism of GNNs. This was to learn a discrete, adjustable mask capable of isolating the most important subgraph in the input with respect to a given question. Our findings underscore that, even when constraining the model to operate on a subset of nodes, it remains feasible to attain competitive performance, thereby mitigating the gap between entirely black-box models and those that are interpretable. It is noteworthy that the $topK\%$-hyperparameter, crucial in the model configuration, yielded the highest accuracy when just 15% of nodes were actively used in making predictions.

RQ2: Human Evaluation In order to address the second research question, we conducted a user study to ascertain the preferred explainability methods among human participants[6]. In the evaluation, explanations derived from our method were notably preferred, registering a score of $p_i = 0.52$. The GNNExplainer ranked second with a score of $p_i = 0.35$, while both the Integrated Gradients and the random explainer demonstrated lower preferences. Analyzing the results on a question-type specific basis, divergent preferences were identified, underscoring the competitive performance between our method and GNNExplainer. The GNNExplainer was predominant in for the question types *attribute*, *relation*, *category*, *query*, and *compare*. In contrast, our method exhibited superior performance in the *object*, *verify*, *choose*, and *logical* question types. For the *global* category, both methods achieved congruent scores.

RQ3: Quantitative Evaluation We introduced two metrics to assess the quality of the explanations, specifically the subgraphs, whether they captured crucial aspects of the questions and answers. Firstly, token co-occurrences between either the question or answer and the graph nodes enabled us to evaluate if the model appropriately concentrated on relevant input segments. Secondly, we measured the performance drop when the explanation subgraphs were omitted from the input graph.

Metric	Pearson	Spearman
AT-COO	0.84	0.60
QT-COO	0.99	1.00
Subg-Rem	-0.48	-0.32

Table 6: Pearson and Spearman correlation scores between our quantitative metrics and the outcomes of the human evaluation.

To determine the effectiveness of our metrics, we computed the Pearson and Spearman correlation

[6] Qualitative examples are in the Appendix A.5

between the outcomes from our human assessment and our quantitative evaluations, as illustrated in Table 6. The AT-COO exhibits a strong positive Pearson correlation and a moderate Spearman correlation. Both Pearson and Spearman metrics for QT-COO present a very high correlation. Interestingly, the subgraph removal metric exhibits a mild negative correlation for both Pearson and Spearman, a trend that aligns with our expectations as a lower accuracy for this metric indicates better performance.

From these findings, it can be inferred that explainability methods with higher token co-occurrence values or lower subgraph removal performances are also generally favored by human evaluators, as supported by our human evaluation results.

8. Conclusion

We proposed an interpretable approach for graph-based VQA that intrinsically generates a subgraph as an explanation during the answer prediction. Despite utilizing only a subset of the nodes from the input graph, the model demonstrates competitive performance. Through a human evaluation, it was discerned that our intrinsically produced explanations were more frequently favored over other state-of-the-art post-hoc explainability methods. Moreover, our evaluation extended beyond the conventional answer token accuracy metric, leveraging token co-occurrences between the question, answer, and graph nodes and assessing the performance decreases upon removal of the subgraph. We presented the results of these metrics for the selected post-hoc explainability methods in comparison to ours. The quantitative measures demonstrated correlations with human evaluators, thus serving as effective metrics for the explanation's quality. Additionally, our method's inherent capability to generate the subgraph explanation concurrently with the answer eliminates the need for an additional method with further hyperparameter tuning, reducing the computational overhead.

9. Ethics Statement

All subjects gave their informed consent for inclusion before they participated in the study. We provided a detailed description of the task and research objectives and did not collect personally identifying data from any users. All logs and survey responses are encrypted using an anonymous hash generated based on the freely chosen username, rather than the plain username. We verified the estimated time in our pilot study to ensure the time we selected was below the median time. All participants took part voluntarily and could stop participating at any time.

10. Limitations

Generally, the performance of machine learning models is dependent on the quality and quantity of data they are trained on. Consequently, every model learns biases from the data distributions they have been exposed to during training. This limits the applicability to real-world scenarios, which should be tested before deployment. In our case, the model might have picked up certain biases regarding the distributions of objects, scenes, or relations among objects. As a result, certain categories of objects and scenes might be over- or underrepresented. While we chose scene graphs to represent images as graphs to increase the interpretability of deep learning architectures, this simplification of scenes displayed in images might lose information about tiny nuances contained in the raw image.

11. Acknowledgement

Funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2075 – 390740016. We acknowledge the support by the Stuttgart Center for Simulation Science (SimTech).

12. Bibliographical References

Chirag Agarwal, Owen Queen, Himabindu Lakkaraju, and Marinka Zitnik. 2023. Evaluating explainability for graph neural networks. *Scientific Data*, 10(144).

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4971–4980.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Leila Arras, Ahmed Osman, and Wojciech Samek. 2022. Clevr-xai: A benchmark dataset for the ground truth evaluation of neural network explanations. *Information Fusion*, 81:14–40.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Ralph A Bradley and Abdalla T El-Helbawy. 1976. Treatment contrasts in paired comparisons: Basic procedures with application to factorials. *Biometrika*, 63(2):255–262.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Shaked Brody, Uri Alon, and Eran Yahav. 2021. How attentive are graph attention networks? In *International Conference on Learning Representations*.

Marta Caro-Martinez, Anjana Wijekoon, Belén Diaz-Agudo, and Juan A Recio-Garcia. 2023. The current and future role of visual question answering in explainable artificial intelligence. CEUR Workshop Proceedings.

Zhuo Chen, Jiaoyan Chen, Yuxia Geng, Jeff Z Pan, Zonggang Yuan, and Huajun Chen. 2021. Zero-shot visual question answering using knowledge graph. In *The Semantic Web–ISWC 2021: 20th International Semantic Web Conference, ISWC 2021, Virtual Event, October 24–28, 2021, Proceedings 20*, pages 146–162. Springer.

Enyan Dai, Tianxiang Zhao, Huaisheng Zhu, Junjie Xu, Zhimeng Guo, Hui Liu, Jiliang Tang, and Suhang Wang. 2022. A comprehensive survey on trustworthy graph neural networks: Privacy, robustness, fairness, and explainability. *arXiv preprint arXiv:2204.08570*.

Aosong Feng, Chenyu You, Shiqiang Wang, and Leandros Tassiulas. 2022. Kergnns: Interpretable graph neural networks with graph kernels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6614–6622.

Matthias Fey and Jan E. Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

Xingyu Fu, Ben Zhou, Sihao Chen, Mark Yatskar, and Dan Roth. 2023. Interpretable by design visual question answering. *arXiv preprint arXiv:2305.14882*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Drew Hudson and Christopher D Manning. 2019a. Learning by abstraction: The neural state machine. *Advances in Neural Information Processing Systems*, 32.

Drew A Hudson and Christopher D Manning. 2019b. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910.

Rajat Koner, Hang Li, Marcel Hildebrandt, Deepan Das, Volker Tresp, and Stephan Günnemann. 2021. Graphhopper: Multi-hop scene graph reasoning for visual question answering. In *The Semantic Web–ISWC 2021: 20th International Semantic Web Conference, ISWC 2021, Virtual Event, October 24–28, 2021, Proceedings 20*, pages 111–127. Springer.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Hao Li, Xu Li, Belhal Karimi, Jie Chen, and Mingming Sun. 2022. Joint learning of object graph and relation graph for visual question answering. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 01–06. IEEE.

Weixin Liang, Yanhao Jiang, and Zixuan Liu. 2021. Graghvqa: Language-guided graph neural networks for graph-based visual question answering. In *Proceedings of the Third Workshop on Multimodal Artificial Intelligence*, pages 79–86.

Zhihong Lin, Donghao Zhang, Qingyi Tao, Danli Shi, Gholamreza Haffari, Qi Wu, Mingguang He, and Zongyuan Ge. 2023. Medical visual question

answering: A survey. *Artificial Intelligence in Medicine*, page 102611.

Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. 2020. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18.

Ninghao Liu, Qizhang Feng, and Xia Hu. 2022. Interpretability in graph neural networks. *Graph Neural Networks: Foundations, Frontiers, and Applications*, pages 121–147.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*.

Ričards Marcinkevičs and Julia E Vogt. 2020. Interpretability and explainability: A machine learning zoo mini-tour. *arXiv preprint arXiv:2012.01805*.

Mathias Niepert, Pasquale Minervini, and Luca Franceschi. 2021. Implicit mle: backpropagating through discrete exponential family distributions. *Advances in Neural Information Processing Systems*, 34:14567–14579.

Amrita Panesar, Fethiye Irmak Doğan, and Iolanda Leite. 2022. Improving visual question answering by leveraging depth and adapting explainability. In *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 252–259. IEEE.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215.

Christina Sarkisyan, Mikhail Savelov, Alexey K Kovalev, and Aleksandr I Panov. 2022. Graph strategy for interpretable visual question answering. In *International Conference on Artificial General Intelligence*, pages 86–99. Springer.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-okvqa: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision*, pages 146–162. Springer.

Sofia Serrano and Noah A Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Zhenwei Shao, Zhou Yu, Meng Wang, and Jun Yu. 2023. Prompting large language models with answer heuristics for knowledge-based visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14974–14983.

Kevin J Shih, Saurabh Singh, and Derek Hoiem. 2016. Where to look: Focus regions for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4613–4621.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Dirk Väth, Pascal Tilli, and Ngoc Thang Vu. 2021. Beyond accuracy: A consolidated tool for visual question answering benchmarking. *arXiv preprint arXiv:2110.05159*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Minh Vu and My T Thai. 2020. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. *Advances in neural information processing systems*, 33:12225–12235.

Yanan Wang, Michihiro Yasunaga, Hongyu Ren, Shinya Wada, and Jure Leskovec. 2022. Vqa-gnn: Reasoning with multimodal semantic graph for visual question answering. *arXiv preprint arXiv:2205.11501*.

Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, Bo Long, et al. 2023. Graph neural networks for natural language processing: A survey. *Foundations and Trends® in Machine Learning*, 16(2):119–328.

Yaochen Xie, Zhao Xu, Jingtun Zhang, Zhengyang Wang, and Shuiwang Ji. 2022. Self-supervised learning of graph neural networks: A unified review. *IEEE transactions on pattern analysis and machine intelligence*, 45(2):2412–2429.

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32.

Zihao Zhu. 2022. From shallow to deep: Compositional reasoning over graphs for visual question answering. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8217–8221. IEEE.

A. Appendix

A.1. GQA Dataset

GQA (Hudson and Manning, 2019b) is a dataset for real-world visual reasoning and compositional question answering. The questions are created automatically based on Visual Genome (Krishna et al., 2017) scene graphs. The dataset consists of $113,018$ images and $22,669,678$ questions that are grouped into different splits. The splits are divided as follows: 70% train, 10% validation, 10% test and 10% as challenge test set. Each image of the training and validation split has a corresponding ground-truth scene graph (Hudson and Manning, 2019b).

A.2. Implementation Details

We use the AdamW (Loshchilov and Hutter, 2018) optimizer with a learning rate of $1e-4$ and a weight decay factor of $1e-5$. Additionally, we apply an exponential learning rate scheduler with an initial learning rate of $1e-6$ and a warmup duration of 15 epochs. Afterward, we decrease the learning by a factor of 0.98 for each epoch. The total number of epochs depends on the loss on the validation set, which we observed to occur after training for 60 to 70 epochs (i.e. we apply early stopping). To further enhance the training procedure, we use a gradient scaler. We train on four GPUs with a batch size of 128 samples. Our model has $44,945,761$ trainable parameters. One training epoch consists of $943,000$ data instances, the validation split of $132,062$ instances.

A.3. Human Evaluation

Our user study was performed online via a web interface. First of all, users need to agree on a data collection policy before a few questions regarding demographic information are asked. Afterward, the actual user study was performed.

Data Collection Policy In the following, we list the information users need to agree to participate in the user study:

- **Purpose of research:** To examine preferences of different explainability methods.
- **What users will perform:** Users will be provided with 18 images, a corresponding question, the prediction from the model, and explanations of two explainability methods in the form of graphs.
- **Time required:** Participation will take approximately 5-10 minutes.

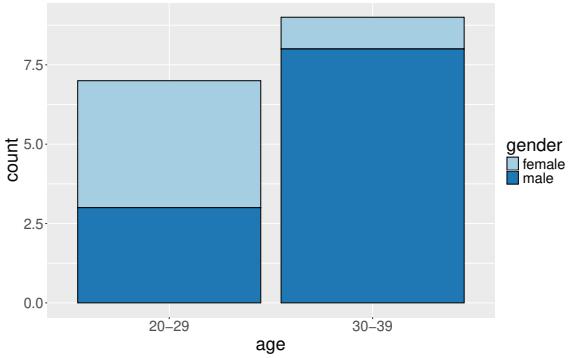


Figure 2: Age and gender distribution of our participants in the user study.

- **Risks:** There are no anticipated risks associated with participating in this study. The effects of participating should be comparable to those you would experience from viewing a computer monitor for 5-10 minutes and using a mouse and keyboard.
- **Limitations:** This task is suitable for all people who can read from and input text into a computer.
- **Confidentiality:** Your participation in this study will remain confidential. Your responses will be assigned a code number. You will be asked to provide your MechanicalTurk ID, but this will not be stored, but rather converted to an anonymous hashed ID. You will be asked to provide your age and gender and previous experience with chatbots/business travel. Throughout the experiment, we may collect data such as your textual input, and your feedback in the form of a questionnaire. The records of this study will be kept private. In any sort of report we make public we will not include any information that will make it possible to identify you. Research records will be kept in a locked file; only the researchers will have access to the records.
- **Participation and Withdrawl:** Your participation in this study is voluntary, and you may withdraw at any time.
- **Data Regulation:** Your data will be processed for the following purposes: (1) Analysis of the respondents' evaluations of the dialog and their experience, (2) analysis of potential influencing factors for individual behavior of the participants in the interaction with the dialog system, and (3) scientific publication based on the results of the above analyses.

Demographic Information We asked each participant about the gender they identified as, given a

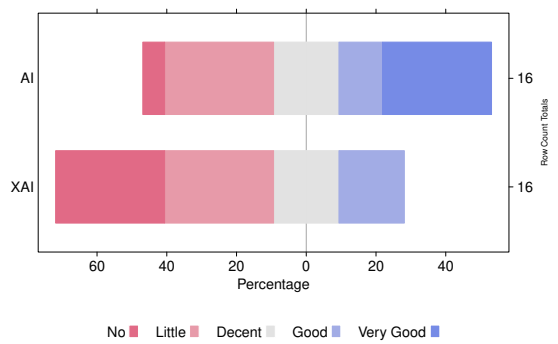


Figure 3: Results of the self-assessment regarding the general topics of artificial intelligence (AI) and explainable artificial intelligence (XAI).

set of three options: female, male, and other. Participants were given the option to select their age category from the following ranges:

- Less than 20
- 20 to 29
- 30 to 39
- etc.

The gender and age distributions are displayed in Figure 2. The study participants were in the range from 20 to 39 with slightly more male users.

We asked two more questions about the participants' general understanding of machine learning (ML) or artificial intelligence (AI) and general understanding of the field around explainable artificial intelligence (XAI). Figure 3 contains the results, measured by a Likert scale. Our participants range from no understanding of ML to a very good understanding, contrary to the understanding in XAI where most participants have little to no knowledge about.

Explainability User Study Each participant receives 18 images accompanied by two scene graphs with the respective highlighted subgraph that serves as an explanation of the model's prediction. The image is displayed as a reference only, it is communicated to the users that the image was not used by the model to predict an answer. Next to the original image, we displayed the question, the predicted answer, and the ground-truth label. We mention that some questions might be ambiguous and that this should not be evaluated or taken into account when evaluating the explanations.

We further describe that we display the corresponding graphs below the original image. The whole graph is input to the model alongside the question. The graph itself (all nodes, green and blue nodes combined) might not be a perfect representation of the image. Nodes, which represent objects in the image, might be missing, or the annotation (the label/name) might be misleading. We displayed the edges between nodes in the visualization of the graph, but we excluded the annotation (the name of the relation). Edges represent relations between objects, e.g. *a man holding a racket* would result in two nodes, *man* and *racket*, and one edge (relation) *holding* between them.

We always perform a pair-wise comparison between two explainability methods, users can find their explanations next to each other. All nodes colored in green are part of the subgraph that represents the explanation. All nodes colored in blue are excluded, so they are not part of the explanation. To judge which explanation users prefer, they should take the question and answer into account, and evaluate if the nodes in green form a more valid explanation than the other explanation. Some explanation methods include more graph nodes in the explanations, while others tend to have fewer nodes included.

A.4. Extended Results

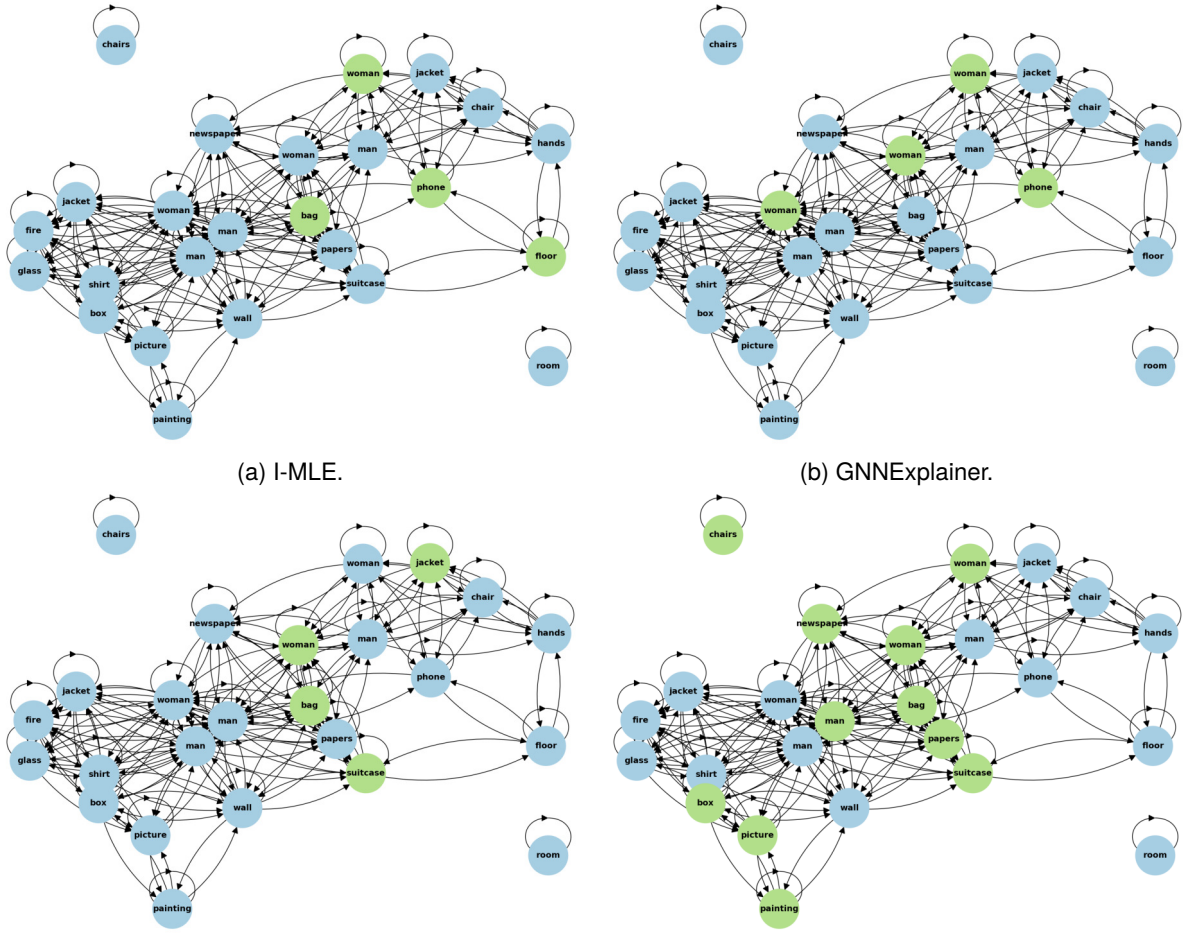
We provide the results from Section 6 for all *top-k* values of Table 2 in Table 7. We report the question answering accuracy as **QA**, the answer token co-occurrences as **AT-COO**, the question token co-occurrences as **QT-COO**, and the question answering accuracy after removing the explanation subgraph as **QA-SubG**. We find that the results for the methods *Random*, *PGMExplainer*, and *Integrated Gradients* stay roughly

Method	TopK%	QA	AT-COO	QT-COO	QA-SubG
Random	.15	94.79	30.59	29.79	52.10
PGMExplainer	.15	94.79	22.37	24.67	69.46
Integrated Gradients	.15	94.79	8.14	39.95	33.28
GNNExplainer	.15	94.79	89.12	59.67	33.28
Ours	.15	94.79	75.15	78.35	37.13
Random	.20	94.15	28.35	27.88	51.07
PGMExplainer	.20	94.15	26.29	29.94	70.29
Integrated Gradients	.20	94.15	8.76	32.78	30.87
GNNExplainer	.20	94.15	91.24	50.66	30.87
Ours	.20	94.15	33.51	80.68	45.63
Random	.25	94.72	30.41	33.63	52.10
PGMExplainer	.25	94.72	34.34	39.62	69.46
Integrated Gradients	.25	94.72	7.14	40.06	33.28
GNNExplainer	.25	94.72	92.99	63.60	33.28
Ours	.25	94.72	81.26	79.20	37.13
Random	.30	94.21	30.46	29.53	48.58
PGMExplainer	.30	94.21	37.93	38.23	61.74
Integrated Gradients	.30	94.21	6.90	36.00	34.82
GNNExplainer	.30	94.21	94.83	61.42	34.82
Ours	.30	94.21	78.74	89.11	39.88
Random	.50	93.20	31.28	29.82	48.49
PGMExplainer	.50	93.20	39.89	38.92	58.48
Integrated Gradients	.50	93.20	7.33	32.62	33.15
GNNExplainer	.50	93.20	94.68	61.70	33.15
Ours	.50	93.20	74.81	88.50	35.37

Table 7: **AT-COO** abbreviates answer-token co-occurrence. **QT-COO** abbreviates question-token co-occurrence. **QA-SubG** refers to the question answering accuracy, when the explanation subgraph is removed.

the same across different *tok-k* sized models. However, increasing the subgraph size with the number of nodes included (determined by the *top-k* factor), the AT-COO and QT-COO increase, while the QA-SubG remains roughly the same. Since we match the number of nodes included from the soft mask learned by *GNNExplainer* with the number of nodes received by our *top-k* factor, this effect is unsurprising. Nevertheless, it is worth noting that the QA-SubG accuracy drops by a similar order of magnitude when we learn to identify a subgraph with a size of *tok-k*$= .15$ or *tok-k*$= .5$, highlighting the importance of nodes included in the subgraphs.

A.5. Qualitative Examples



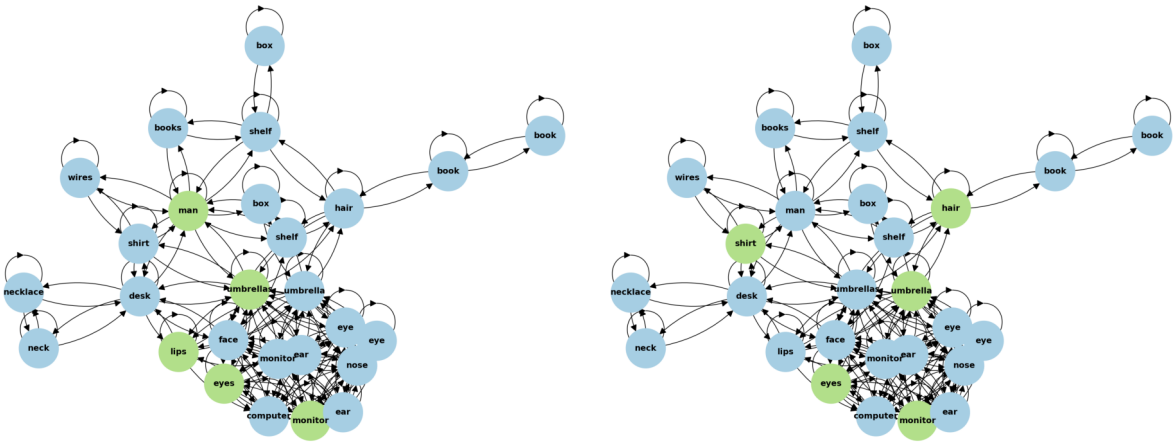
(a) I-MLE.

(b) GNNExplainer.

(c) Random.

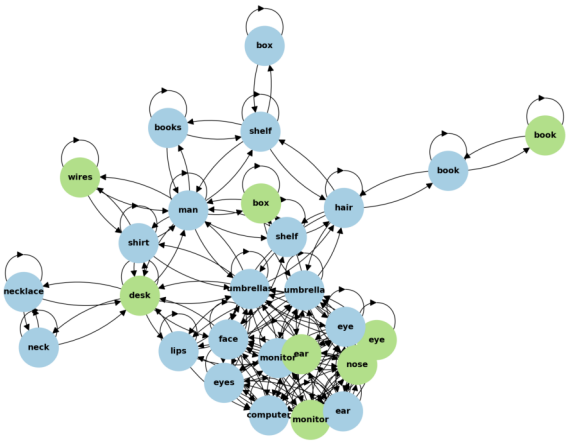
(d) Integrated-Gradients.

Figure 4: Graph and image for question Id: 17745707. Question: *Is the woman to the left or to the right of the phone?*Prediction: *left*. Ground-truth answer: *left*. Semantic type: *relation*. Structural type: *choose*.



(a) I-MLE.

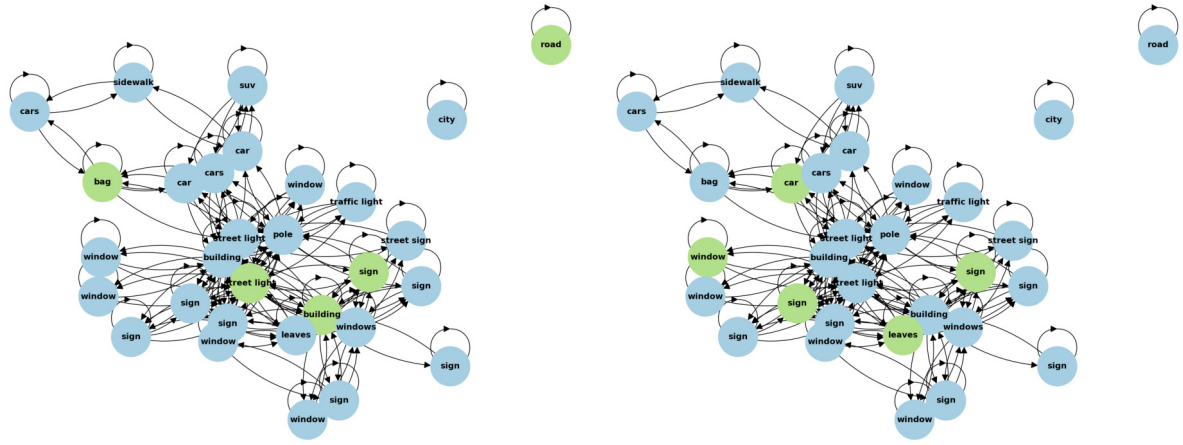
(b) GNNExplainer.



(c) Random.

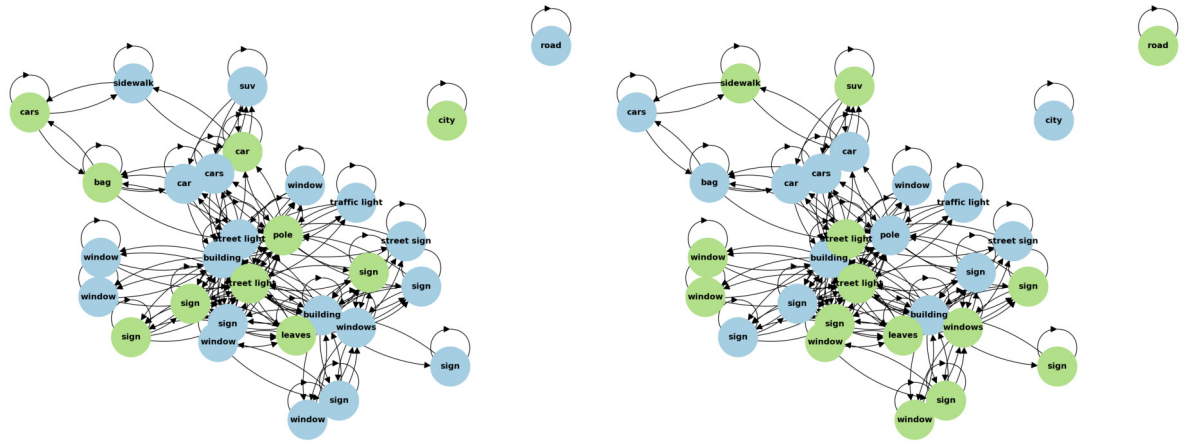
(d) Integrated-Gradients.

Figure 5: Graph and image for question Id: 17267496. Question: *Are his eyes large and green?* Prediction: *no*. Ground-truth answer: *no*. Semantic type: *attribute*. Structural type: *logical*.



(a) I-MLE.

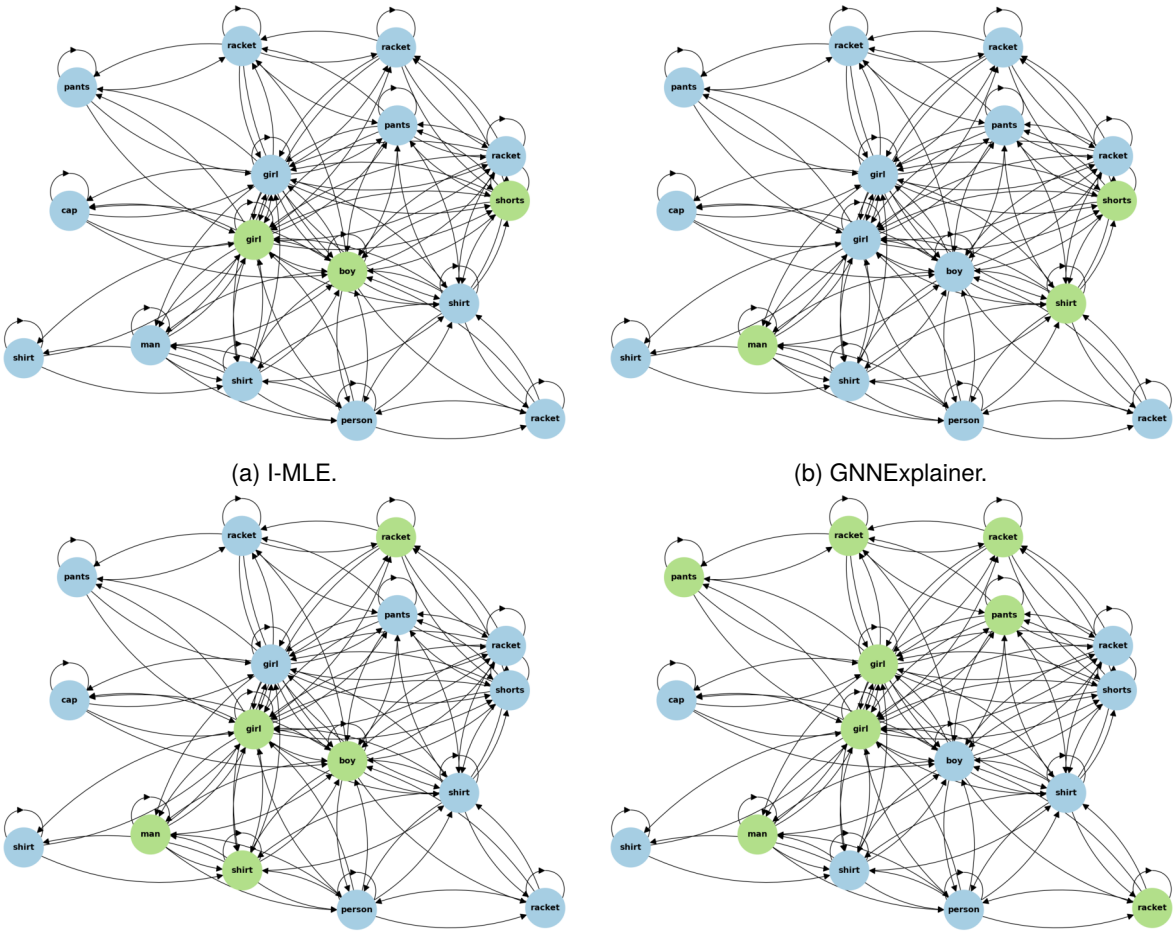
(b) GNNExplainer.



(c) Random.

(d) Integrated-Gradients.

Figure 6: Graph and image for question Id: 07339770. Question: *Are there either pizza trays or hand soaps in the image?* Prediction: *no*. Ground-truth answer: *no*. Semantic type: *object*. Structural type: *logical*.



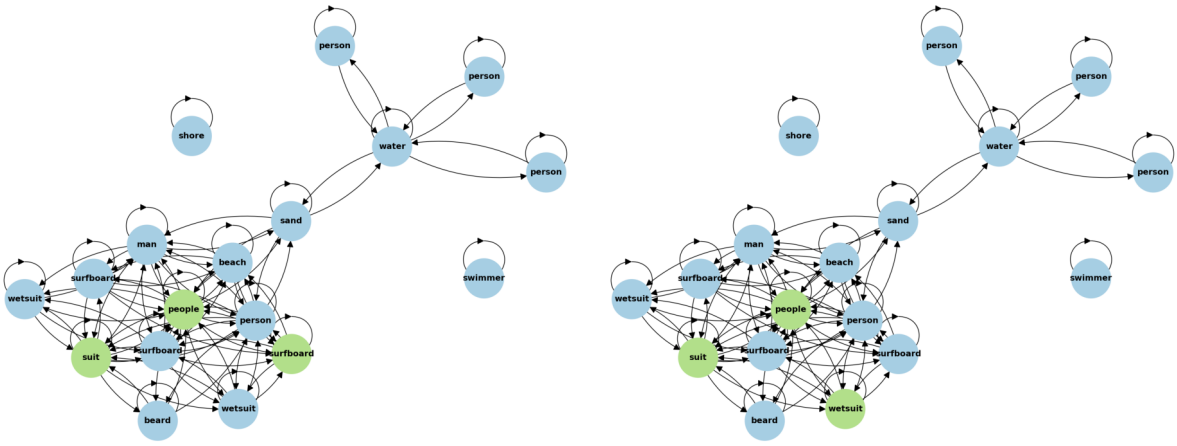
(a) I-MLE.

(b) GNNExplainer.

(c) Random.

(d) Integrated-Gradients.

Figure 7: Graph and image for question Id: 11389703. Question: *Does the girl to the right of the person wear shorts?* Prediction: *yes*. Ground-truth answer: *yes*. Semantic type: *relation*. Structural type: *verify*.



(a) I-MLE.

(b) GNNExplainer.

(c) Random.

(d) Integrated-Gradients.

Figure 8: Graph and image for question Id: 1662748. Question: *Who is wearing a suit?* Prediction: *people*. Ground-truth answer: *people*. Semantic type: *relation*. Structural type: *query*.