3D Question Answering with Scene Graph Reasoning

Anonymous Authors

ABSTRACT

3DQA has gained considerable attention due to its enhanced spatial understanding capabilities compared to image-based VQA. However, existing 3DQA methods have explicitly focused on integrating text and color-coded point cloud features, thereby overlooking the rich high-level semantic relationships among objects. In this paper, we propose a novel graph-based 3DQA method termed 3DGraphQA, which leverages scene graph reasoning to enhance the ability to handle complex reasoning tasks in 3DQA and offers stronger interpretability. Specifically, our method first adaptively constructs dynamic scene graphs for the 3DQA task. Then we inject both the situation and the question inputs into the scene graph, forming the situation-graph and the question-graph, respectively. Based on the constructed graphs, we finally perform intra- and inter-graph feature propagation for efficient graph inference: intra-graph feature propagation is performed based on Graph Transformer in each graph to realize single-modal contextual interaction and high-order contextual interaction; inter-graph feature propagation is performed among graphs based on bilinear graph networks to realize the interaction between different contexts of situations and questions. Drawing on these intra- and inter-graph feature propagation, our approach is poised to better grasp the intricate semantic and spatial relationship issues among objects within the scene and their relations to the questions, thereby facilitating reasoning complex and compositional questions. We validate the effectiveness of our approach on SQA3D and ScanQA datasets, and expand the SQA3D dataset to SQA3D Pro with multi-view information, making it more suitable for our approach. Experimental results demonstrate that our 3DGraphQA outperforms existing methods.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision tasks**.

KEYWORDS

3D Question Answering, Scene Understanding, Graph Neural Networks, Spatial Relation Reasoning

1 INTRODUCTION

Recently, 3DQA, i.e., answering questions related to 3D scenes, has received widespread attention due to its applicability in various downstream tasks such as visual language navigation [19, 30], intelligent agents [27, 34], and autonomous driving [14]. To some extent,

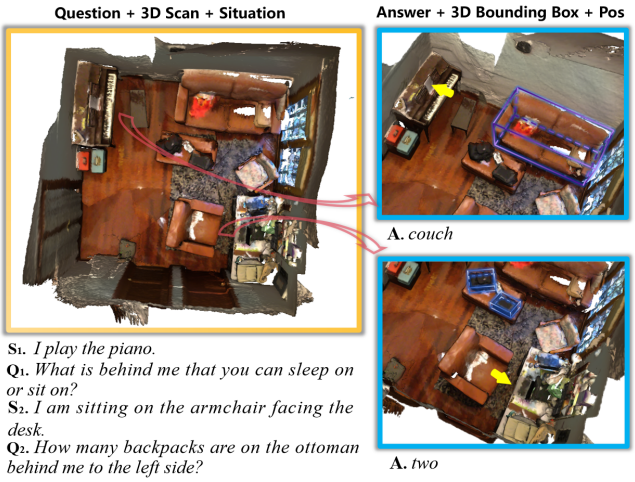

Figure 1: This figure illustrates examples of the 3DQA task, which involves reasoning about the complex semantic and spatial relationships among objects within the scenes and questions.

3DQA serves as an extension of the VQA task, since both tasks require the analysis and understanding of multimodal visual and textual content, followed by inference to obtain answers. However, compared to 2D VQA tasks, reasoning within authentic 3D environments allows for the avoidance of spatial ambiguities inherent in 2D data, thereby requiring the acquisition of accurate geometric information and object relationships. Moreover, 3D scenes typically encompass a greater number of objects and entail more intricate interrelationships among them.

Many efforts have been made to address the challenges of 3DQA. For example, Ye et al.[35] proposed 3DQA-TR, a transformer-based 3DQA model, that leverages two separate encoders to encode the appearance and geometry information and then adopts a 3D-linguistic Bert [11] to fuse multi-modal information including appearance, geometry and linguistic question for 3DQA. Azuma et al.[3] developed a baseline model based on their constructed ScanQA dataset, which primarily comprises 3D and language encoders, 3D and language fusion, and 3D object localization and 3D QA modules. It can be observed that these methods predominantly focus on aligning and fusing text and point cloud features. Though significant progress has been made, these approaches still exhibit limitations in interpretability and handling complex semantic and spatial relationship reasoning. For example, consider the question posed in Figure 1. “ What is behind me that you can sleep on or sit on?” or "How many backpacks are on the ottoman behind me to the left side?" not only requires detecting objects in geometric scenes but also understanding the complex semantic and spatial relationships between scene objects, questions and situations.

This paper is inspired by graph-based 2D VQA methods [18, 28, 31, 32], which have demonstrated the effectiveness of enhancing the

Unpublished working draft. Not for distribution.

ACM MM, 2024, Melbourne, Australia

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

interpretability and reasoning capabilities for complex questions in VQA. However, directly transferring the graph structure from VQA to 3DQA is not feasible, because 2D scene graphs only carry local information, whereas our 3D scene graphs can encapsulate both global scene information and support the generation of end-to-end dynamic graphs. Depending on the situation and the question setting, different scene graphs can be constructed for the same scene. For example, the situation: *I play the piano*, and the situation: *I am sitting on the armchair facing the desk.* require different scene graphs of the scene for the question: *What is behind me that you can sleep on or sit on?*.

In this paper, we propose a novel graph-based 3DQA method, termed 3DGraphQA, exploring using the graph reasoning to enhance the performance of 3DQA task. Specifically, in the graph construction stage, our method utilizes VoteNet [24] to detect objects in the scene based on the specified situational setting. The detected objects and their features act as nodes in the graph and are optimized using prior knowledge, thereby forming the initial scene graph. Note that as the situation setting changes, the final scene graph also changes accordingly, making the scene graph dynamic. Our method injects the input situation description and question description separately into the scene graph to obtain the situation-graph and the question-graph; In the graph feature fusion stage, to facilitate the high-order contextual interaction within the graph and the cross-modal contextual interaction between the situation-graph and the question-graph, our method investigates graph transformer to realize intra-graph contextual interaction, and adopts bilinear graph networks to realize inter-graph contextual interaction.

Our method is experimentally validated on the ScanQA and SQA3D datasets, both of which are built upon the ScanNet dataset. Moreover, to further explore the potential of our method, we also develop the SQA3D Pro dataset, which is an extension of the SQA3D dataset with additional multi-view situation information, drawing inspiration from ScanQA dataset. We show that the SQA3D Pro dataset provides rich information for the scenes and facilitates a better understanding of the current scene for scene graph reasoning, enabling performance improvement. The experimental results have shown that our proposed method achieves SOTA performance on these datasets.

In summary, our contributions are listed as follows:

- We propose a novel graph-based 3DQA method, which exploits dynamic scene graphs to facilitate the 3DQA tasks.
- We introduce a Graph Transformer-based model for intra-graph feature fusion, enabling contextual interactions between the scene objects and the question, and between the scene objects and the situation description.
- We leverage the bilinear graph neural network for inter-graph feature fusion, which can enhance contextual interactions between different graphs.
- We develop SQA3D Pro dataset, which is an extension of the SQA3D dataset with additional multi-view situation information, drawing inspiration from ScanQA dataset.
- We conduct extensive experiments on two public benchmark datasets, i.e., SQA3D and ScanQA datasets. Experimental results show that our model outperforms all baseline methods.

2 RELATED WORK

In this section, we give a brief review on recent advances in 3D Question Answering, 3D Captioning and Visual Grounding, as well as Graph-based Visual Question Answering.

2.1 3D Question Answering

3D Question Answering (3DQA) refers to answering questions related to 3D scenes, distinguishing it from VQA, which aims to answer questions related to 2D images. A few benchmarks have been devised in the field including SQA3D [20] and ScanQA datasets [3, 35] for the evaluation. Depending on the output answering content, we categorize previous 3DQA work into two types: those that solely provide answers and those that simultaneously provide answers along with object information.

For the methods that solely provide answers, Ye et al. [35] proposed 3DQA-TR, a transformer-based 3DQA method, that adopts a 3D-linguistic Bert to first encode the point cloud into tokens, then these tokens, along with the text encoding of the question, are jointly fed into the Bert for training and inference. The output answers are generated by the decoder of the Bert model. Zhou et al.[33] proposed the Chat-3D method, where they first segment each object in the scene. Then, they encode each object individually and fed them into a large language model (GPT) for answer prediction. Leveraging the powerful knowledge priors of GPT, this method can thus obtain open-ended answer results. Ma et al.[20] proposed a zero-shot 3DQA method, called SQA3D, which first inputs the point cloud scene into the pre-trained Scan2Cap model to obtain dense captions of the scene. Subsequently, these dense captions are combined with the context description and the question, to feed into the GPT-3 to obtain the answer.

For the methods that simultaneously provide answers along with object information, Azuma et al. [3] proposed ScanQA, which consists of 3D and language encoding layers, three-dimensional and language fusion layers, and object localization and language decoding layers. In the fusion layer, it mainly references the 2D MCAN [37], using transformer blocks to capture the relationship between objects and questions. This method facilitates answering questions and 3D object localization. Delitzas et al. [10] investigated Multi-CLIP, a pre-training multi-modal CLIP-based architecture that embeds the 3D scene features to their corresponding captions and multi-view images in the CLIP space. This method facilitates the tasks of question answering and referred object localization, and 3D situated question answering.

In this paper, to address the interpretability and complex reasoning problems in the field, we propose a novel scene graph-based method for 3D question answering. We also extend the SQA3D dataset to SQA3D Pro dataset with multi-view information, enabling more accurate reasoning.

2.2 3D Captioning and Visual Grounding

Witnessing the popularity of the vision-language models in image domains, many researchers have exploited the 3D-language models for 3D scene tasks including 3D captioning [36] and 3D visual grounding. Chen et al. [6] proposed ScanRefer, a novel method for 3D object localization using natural language. They also developed ScanRefer dataset, which is the first large-scale scene-linguistic

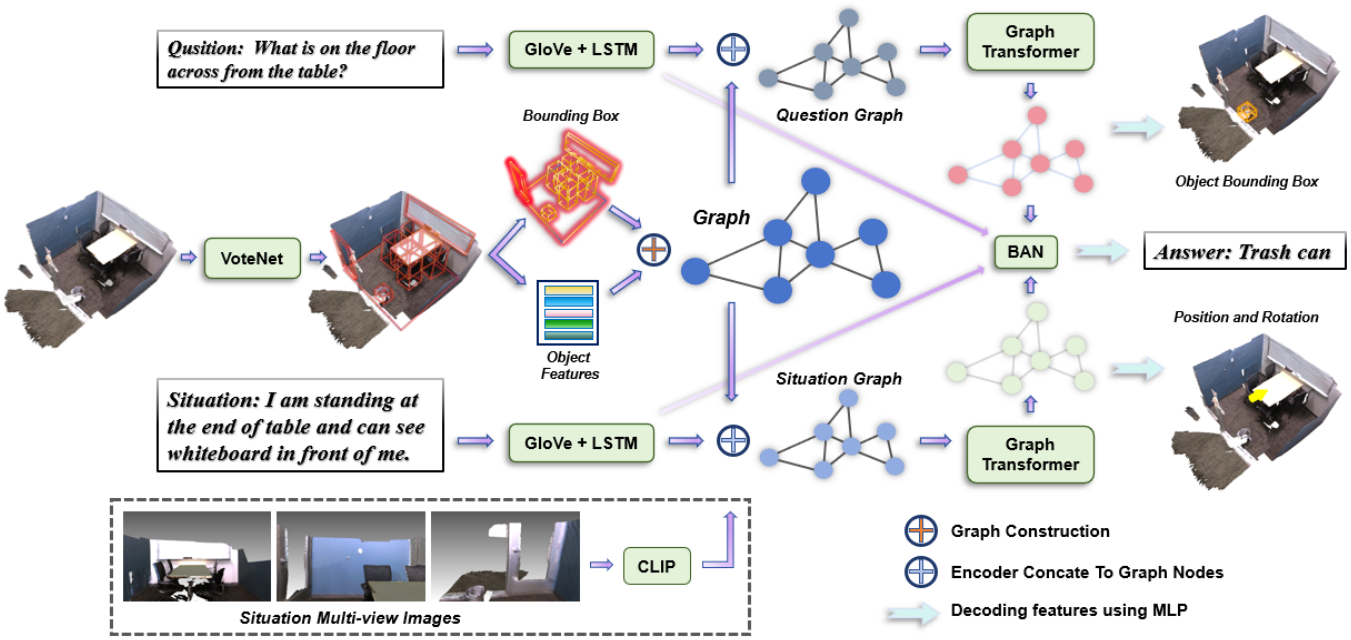

Figure 2: The overall architecture of our 3DGraphQA framework: our method takes the 3D scene context, the situation description and the question as input, ultimately yielding answers to the questions along with object positions and orientations. The key premise of our approach is that the scene graph and graph reasoning model can effectively facilitate complex reasoning tasks in the vision-language domain.

dataset. Subsequently, Achlioptas et al.[1] presented ReferIt3D dataset, which contains fine-grained multi-instance 3D object labels to a scene, aiming to facilitate the task of identifying the instance of the same fine-grained object class in a 3D scene. Based on the two datasets, a large body of work has been dedicated to 3D caption and 3d visual grounding. For example, Chen et al. introduced Scan2Cap [7] for the task of dense scene caption generation. This method utilizes PointNet++ [25] as its foundation to generate textual descriptions for objects in the scene. Following their work, Yuan et al.[38] further proposed X-Tran2Cap, which leverages 2D priors for optimization. Wu et al.[39] proposed EDA, which offers a decoupling approach during text encoding, achieving better alignment between scenes and text. Zhao et al. [39] introduced 3DVG-Transformer, a transformer-based 3D visual grounding method that exploits the proposal relations among objects in 3D scene. Cai et al. [5] and Chen et al. [8] offered unified architectures for joint 3D captioning and visual grounding.

This work also falls within the realm of multi-modal 3D-language tasks. In this work, our primary focus is 3D question answering.

2.3 Graph-based Visual Question Answering

Visual Question Answering (VQA) is a computer vision task that involves both image comprehension and text recognition. Its goal is to equip machines with the capability to comprehend image content and respond to questions about it using natural language. Tremendous efforts have been devoted to the task, with the central challenge lying in how to capture dense semantic interactions and reasoning within and across modalities of images and text. To alleviate this issue, some researchers have attempted to introduce Graph Neural Networks (GNN) into VQA tasks. These works have demonstrated that by integrating graph structural information, such as concept graphs and scene graphs, significant improvements can be made in the interpretability and effectiveness of VQA tasks, thereby enhancing the answer prediction accuracy. For example, Will et al.[21] utilize Graph Convolutional Networks (GCN) to model the semantic correlations between objects in the image scene and learn graph structure representations relevant to the questions. Zhu et al. [40] propose an object-aware graph learning module guided by questions, suggesting that differences provide more information when modeling semantic relationships. To capture complex semantic and spatial relationships in images, many researchers [18, 28, 31, 32] have exploited to use scene graphs to enhance the multi-model image-text learning, facilitating the complex reasoning regarding to the semantic and spatial relationship among objects in 3D scenes.

In this paper, we pioneer the graph-based 3DQA method by exploiting scene graph reasoning. Compared with 2D graph-based VQA, our graph to 3D scene is dynamic, depending on the situation and the question setting.

3 METHOD

In this section, we provide a detailed introduction to our 3DGraphQA, which employs graph reasoning to address the task of 3D question answering. As illustrated in the Fig. 2, 3DGraphVQA takes the point cloud P of the 3D scene context, the situation description s, the question q, as well as multi-view images I, as input, aiming to generate accurate answer from the answer set $a = \{a_1, \cdots, a_M\}$, and corresponding to the bounding box $bbox$ of the question, and predicting

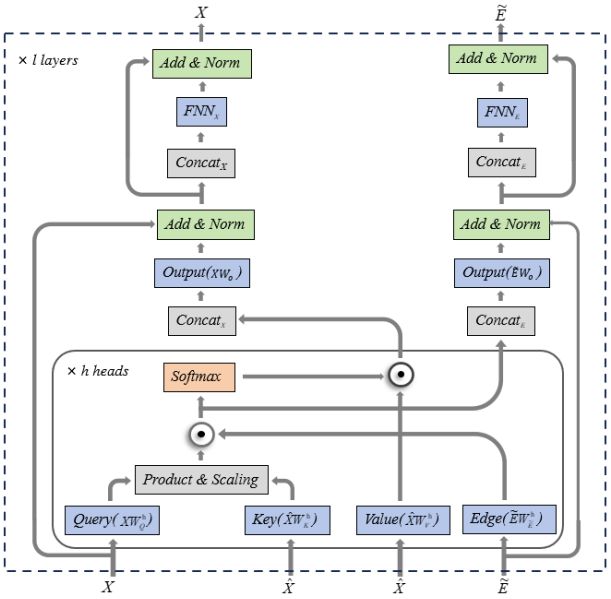

Figure 3: The intra-graph feature fusion architecture based on Graph Transformer.

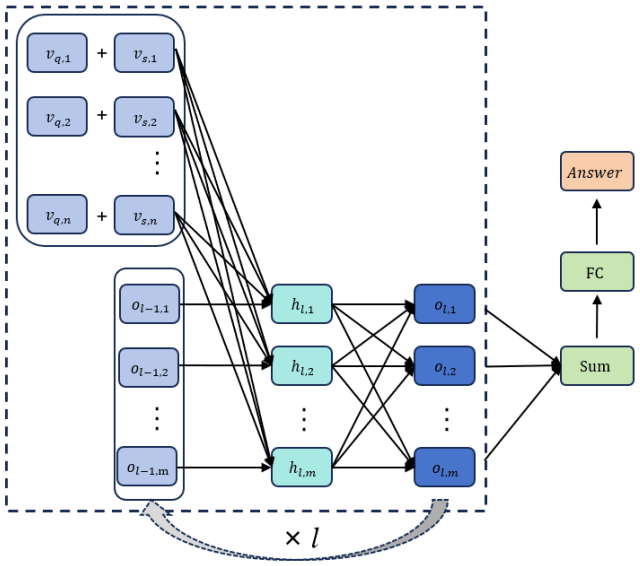

Figure 4: The inter-graph fusion architecture based on the bilinear graph networks.

the accurate situation position s^{pos}, and the current perspective s^{rot} from the situation description. As shown in Fig. 2, 3DGraphQA mainly consists of the three main components: encoder module, scene graph construction module, and graph-based feature fusion module. Below we elaborate on the details of each module.

3.1 Encoder Module

This module aims to encode the input language and scene images into features. Specifically, for the input situation description s and question q, we utilize GloVe [23], followed SQA3D, to encode the situation descriptions and questions. These embeddings are then fed into a biLSTM [15] for sequence modeling. Subsequently, the situation features F_s and the question features F_q are obtained by projecting the outputs of the LSTM through GELUs activation [16] non-linear layers. As for the scene image input, we employ the CILP model [26], a multi-modal pretrained model based on contrastive learning of images and text, to achieve image feature denoted as F_I.

3.2 Scene-graph Construction Module

To construct the scene graph, we first need to encode the point cloud of the scene. Given the point cloud $P = \{p_1, \cdots, p_N\} \in R^{N\times 3}$ of the scene as input, where N denotes the number of points in the scene, our method employs VoteNet [24], which is a point cloud object detection network based on PointNet++ [25], to extract various object proposals and obtain their features. We assume the number of extracted object proposals is denoted by K, and the features of the object proposals are represented by $F_P = \{F_1, \cdots, F_K\}$, where F_i refers to the features of the i-th object. The object bounding boxes are denoted by $B = \{b_1, \cdots, b_K\}$, where b_i denotes the position information of the boundaries, containing the position information of a center and eight corners. In the following, we will use the relative positional relationships between each object in the scene to construct the scene graph.

Initial Graph Construction. We use the K extracted objects in the scene as vertices to construct the initial fully connected undirected graph. Let the graph be represented as $G = (V, E)$ with K nodes $v_i \in V$ and edge element $e_{i,j} \in E$ denotes the edge between the vertices v_i and v_j. We associate the feature set F_P to V, seen as node features, and the elements in the edge set are initially set 1s, leading to a fully connected graph of G. Because there are potential relationships between every two objects in the fully connected graph, and no prior knowledge is used during the learning process, the weights can be learned and adjusted during the subsequent graph-based feature fusion module.

Pruned Graph with Prior-Knowledge. Based on the fully connected graph with uniform weights, which may not effectively measure the spatial relationships between objects in the scene. To overcome this issue, we utilize prior knowledge to prune the graph. Since there are explicit relationships between objects in the scene that can be utilized, we can prune non-existent edges using some prior-knowledge to transform the fully connected graph into a locally connected graph. Specifically, we observed that the performance improves when objects are not connected to themselves compared to when they are connected, so we assign a weight of 0 to the diagonal positions in the adjacency matrix. Then, when establishing local connectivity, we found that most questions are related to objects surrounding the current target object. Therefore, we set a neighborhood range (e.g. k-nearest-neighbor) as the receptive field of the current target and use the bounding boxes of each object as the receptive field to construct this locally connected graph, which can be defined as follows:

$$\tilde{e}_{i,j} = \begin{cases} 1, & p_j \in knn(p_i) \\ 0, & \text{otherwise} \end{cases}, \tag{1}$$

where $\tilde{e}_{i,j} \in \tilde{E}$ is the edge element of pruned edge set $\tilde{E}$, $knn(\cdot)$ represents the k-nearest-neighbor, which is calculated based on the absolute distances between the center of the bound box of an object and the center of the bound box of other surrounding nodes. The remaining positions of the adjacency matrix are set to 0s. Therefore, the connectivity graph can be defined as: $\tilde{G} = (V, \tilde{E})$.

Additionally, we dynamically set the weights of edges based on the proximity of bounding box relative positions. To ensure the robustness of the graph in multimodal fusion, we also randomly add some explicit unidirectional edges to the graph.

Situation-Graph and Question-Graph To better capture the interaction between the situation and question inputs with the 3D scene objects, we investigate injecting the features of perspective images, situation descriptions, and questions into the constructed scene graph. Specifically, considering that the features of individual objects carry local scene information, we connect the encoded image with the features of graph nodes to enhance the global perception capability of target nodes.

Additionally, in 3D scene question answering tasks, there exist different scenes, situation descriptions, and questions, and even the same scene may have different scenarios with varying questions. Therefore, in designing multi-modal feature fusion, it is crucial to address how the adaptive relationship between situation descriptions and questions can be effectively injected into the graph structure and how the weights between multi-modal graph nodes can be learned reasonably. Therefore, we first construct the situation graph $G_s = (V_s, \tilde{E})$ with the feature set of V_s is updated to $[F_P \| F_I \| F_S]$ from the initial F_P, and the question graph is similarly defined as $G_q = (V_q, \tilde{E})$ with the feature set of V_q is updated to $[F_P \| F_q]$ from the initial F_P, where $\|$ denotes the concatenation operation.

3.3 Graph Feature Fusion Module

Within the situation-graph and the question-graph, and between the situation-graph and the question-graph, there exist complex semantic and spatial relationships. To comprehend these relationships, we perform feature fusion in both intra-graph and inter-graph.

Intra-graph Fusion. Intra-graph feature fusion aims to capture complex relations among nodes of the same graph via message propagation. In this work, we adopt the Graph Transformer [12] to facilitate the task, which includes the attention mechanism and the positional encoding, benefiting the global perception while avoiding over-smoothing.

Figure 3 illustrates the architecture of graph transformer, which consists of two essential modules: a multi-head self-attention module (MHA) and a feed-forward network (FFN), akin to vanilla Transformer [29]. Without loss of generality, given the situation-graph $G_s = (V_s, \tilde{E})$, we use $X' = [F_P \| F_I \| F_S]$ to denotes the node features with $x'_i \in X'$ represents the feature of node v_i, and $\tilde{e}_{i,j} \in \tilde{E}$ denotes the edge between v_i and v_j. The graph transformer method first computer the Laplacian eigenvectors [4] δ of the graph G_s and uses them as node positional information to inject into the node feature: $X = [X' \| \delta]$, where $\|$ is the concatenation operation. The graph transformer takes the X and $\tilde{E}$ as inputs and passes them through blocks of MHA and FFN modules. Let $\hat{v}_i$ be one neighborhood node of v_i, the neighborhood feature set can accordingly be defined as $\hat{X}$, the specific one head attention and FFN are defined as follows:

$$A^h(X) = softmax(XW_Q^h(\hat{X}W_K^h)^T \cdot W_{\tilde{E}}\tilde{E}); \tag{2}$$

$$\tilde{E} = \tilde{E} + \sum_{h=1}^{H} XW_Q^h(\hat{X}W_K^h)^T; \tag{3}$$

$$Att(X) = X + \sum_{h=1}^{H} A^h(X)\hat{X}W_V^h W_O^h; \tag{4}$$

$$FFN(X) = X + Relu(XW_1)W_2, \tag{5}$$

where W_Q^h, W_K^h, W_V^h, W_O^h, $W_{\tilde{E}}$ represents the weight matrices of the Query (XW_Q), Key ($\hat{X}W_K$), Value ($\hat{X}W_V$), Output (XW_O) and the edge set for the h-th head, and H is the number of attention head. W_1 and W_2 are the learnable parameters that constitute the FFN with the inclusion of residual connections and normalization. The node features from the final layer will be passed to MLP to compute the output features.

Given the predefined situation-graph and question-graph, our method passes these graphs through the Graph Transformer for intra-graph feature propagation, and outputs the updated graph with updated node and edge representations.

Inter-graph Fusion. For inter-graph feature propagation, our method refers to bilinear graph networks (BAN) [13], as is illustrated in Figure 4. The original BAN work conducts message passing between *image-graph* and *question-graph* in their work. Here, we extend this work to our scene-graph case by analogously representing the learned situation-graph and question-graph as image-graph. To be specific, our method first concatenate the situation-graph and the question-graph, forming an initial joint embedding of scene objects and situation descriptions, and questions. We note that the joint embedding is aware of the words from the situation and question descriptions with objects, but lacks the interaction between questions and the situation description regarding the given scene. Hence, our method then computes the fused embedding between the vectors of the question words and the joint embedding, where the question word vectors are obtained through the text encoder in section 3.1. Finally, the fused embedding ($h_{l,i}$ in Figure 4) is further updated through an MLP to explore their complex interactions. Following the BAN [13], our method also stacked the modules multiple times (see the arrow in Figure 4) by reusing the fused feature output ($o_{l,i}$ in Figure 4) as the input of question vectors recursively.

3.4 Answer Prediction

After obtaining the fused features, we concatenate the output features of BAN module and input them into an MLP to predict a candidate answer from the answer set a. To generate multiple answers, we employ a decoding mechanism similar to ScanQA [3] and calculate the final score using a binary cross-entropy loss function.

To obtain the object proposals, we feed the graph node features, originating from the question-graph and are processed by the Graph Transformer, into an MLP through max-pooling for decoding to obtain the target object proposals. During training, our method calculates the cross-entropy loss by comparing the decoded features with the ground truth bounding box features for each target object.

Figure 5: This image showcases one example from the SQA3D Pro dataset. The left figure is the overhead view from the SQA3D dataset, where the yellow arrow indicates the position and orientation of the first-person perspective within the current scene. The right figure contains the four additional multi-view images supplemented by our dataset: the top-left image is the front view, the top-right image is the back view, the bottom-left image is the left view, and the bottom-right image is the right view.

To reach the localization and rotation of scene objects, a similar approach is employed where we also feed the graph node features, originating from the question-graph and are processed by the Graph Transformer, into an MLP through max-pooling for decoding. During training, we use the mean squared error (MSE) loss to compare the predicted scene position (s^{pos}) and orientation (s^{rot}) obtained from SQA3D with the ground truth preset values.

Our loss function is defined similarly to SQA3D, which includes the answering loss, the position loss, the perspective loss, the location loss, and the object detection loss. We set the final loss as a simple linear combination of these losses, computed as:

$$L = L_{ans} + \lambda_1 L_{obj} + \lambda_2 L_{rot} + \lambda_3 L_{loc} + \lambda_4 L_{det}, \quad (6)$$

where λ_1, λ_2, λ_3 and λ_4 are the weighting factors for the loss functions. In our experiments, we set all these hyper-parameters to 1s.

4 SQA3D PRO DATASET.

In this section, we describe our proposed SQA3D Pro dataset, which is an extension of the SQA3D dataset with additional multi-view situation information. We note that the SQA3D dataset is currently a publicly available dataset for 3D scene question answering, where various situations from a first-person perspective are predefined for each scene in ScanNet and provided in the form of textual descriptions. Unfortunately, the overhead views and videos provided in the SQA3D dataset only correspond to the global semantic information of the entire scene and do not perfectly correspond to the first-person perspective situations.

Inspired by the ScanQA dataset, which incorporates multi-view image information of the entire scene into the model training process, resulting in improved performance. Similarly, for the SQA3D dataset, integrating multi-view information from each first-person perspective into the training process is expected to enhance the effectiveness of the method. Therefore, we also adopt the strategy of incorporating multi-view images to complement the image information of the first-person perspective situations in the SQA3D dataset, resulting in the creation of the SQA3D Pro dataset, as illustrated in the Figure 5.

When supplementing the multi-view images, we select four viewpoints: the front view, the back view, the left view, and the right view. Particularly, if any of these views have missing information or are facing a wall, we exclude them from the dataset. Since the SQA3D dataset comprises 650 indoor scenes from the ScanNet dataset, with 6.8k unique situations and 20.4k situation descriptions, our dataset provides approximately 23k additional multi-view images for the 6.8k unique situations in the SQA3D dataset.

Table 1: The question answering accuracy on the SQA3D dataset. In the method column: "w/o s" denotes the method without situation descriptions, "+ pos" indicates the provision of object position information in addition to textual answers.

Method	Test set						Avg.
	What	Is	How	Can	Which	Other	
ClipBERT [17]	30.2	60.1	38.7	63.3	42.5	42.7	43.3
SQA3D (w/o s) [20]	28.6	65.0	**47.3**	66.3	43.9	42.9	45.3
SQA3D [20]	31.6	63.8	46.0	69.5	43.9	45.3	46.6
SQA3D (+ pos) [20]	33.5	**66.1**	42.4	69.5	43.0	46.4	47.2
3D-Vista [41]	34.8	63.3	45.4	**69.8**	47.2	48.1	48.5
3DGraphQA	**36.6**	64.2	46.0	69.6	**47.9**	47.6	49.0
3DGraphQA(SQA3D Pro)	36.4	64.7	46.1	**69.8**	47.6	**48.2**	**49.2**

5 EXPERIMENTS

In this section, we first describe the experimental setup of our approach, including the datasets, implementation details, and evaluation metrics. Subsequently, we discuss the experimental results of our approach. Finally, we conduct ablation experiments to validate the performance of the main modules in our model.

5.1 Experimental Setup

Datasets. In this work, we validate our method on the ScanQA dataset[3] and SQA3D dataset [20]. The ScanQA dataset [3] contains 800 3D scenes, 41,363 questions, and 58,191 answers, which is built upon ScanRefer [6] and Scan2Cap [7]. We follow the training, validation, and testing set configurations as described in ScanQA [3]. The SQA3D dataset [20] is designed for embodied scene understanding by integrating situation understanding and situated reasoning. It was constructed using 650 scenes from ScanNet [9], comprising 6.8k unique situations, 20.4k descriptions, and 33.4k diverse reasoning questions related to these situations.

Implementation. We implement our model using the PyTorch library, and all experiments are performed on a single Nvidia RTX 4090 GPU. We utilize the ADAM optimizer to train our model, with an initial learning rate of 0.0001, and a batch size of 16. Our model is trained for 80 epochs, after 40 epochs, the learning rate is reduced to 0.00002 for convergence. Additionally, when the number of edges in the graph structure is high, the batch size will also be decreased to 14.

Evaluation Metrics. On the ScanQA dataset [3], we employ the same evaluation metrics as the work [3], which include EM@1 and EM@10, where EM denotes the exact match and EM@K represents the percentage of predictions that exactly match any ground truth

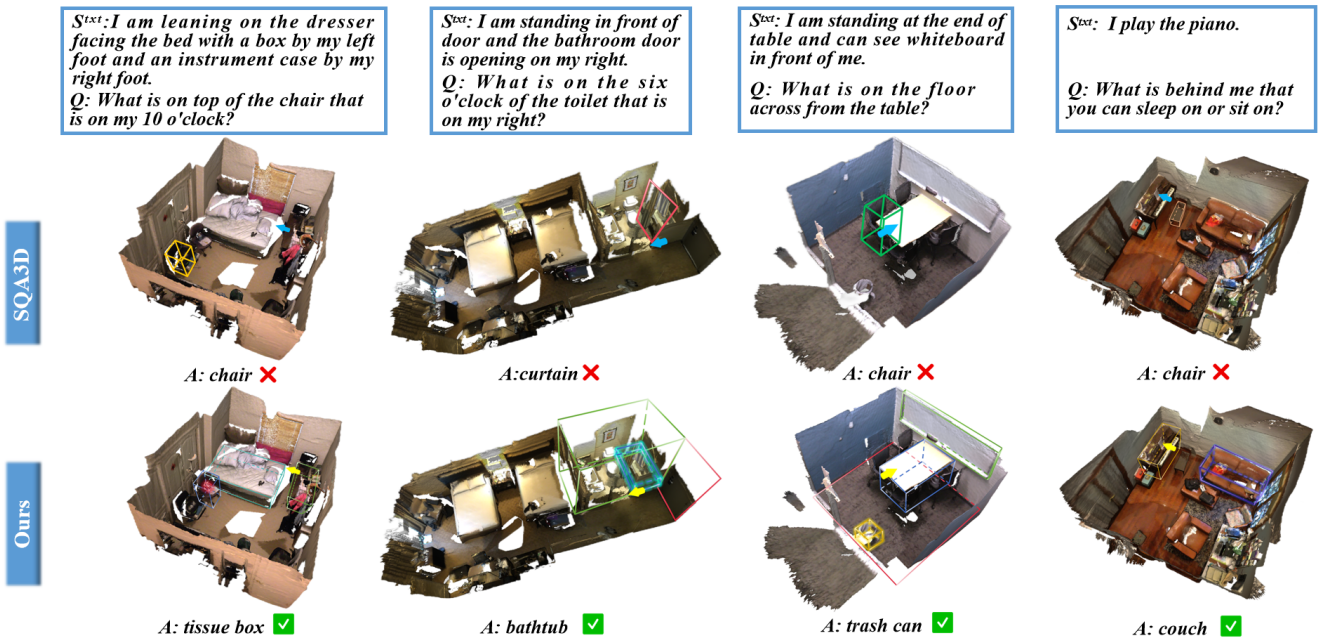

Figure 6: The qualitative comparison results between our method and SQA3D. From the figure, our approach achieves higher accuracy in answering "what"-related questions, and can provide accurate bounding boxes for relevant objects. In contrast, SQA3D's answers are often inaccurate and fall short in providing bounding boxes which serve as auxiliary aids for understanding.

answer among the top K predicted answers. Meanwhile, we utilize BLEU-4, ROUGE, METEOR, and CIDEr as the sentence-level evaluation metrics. On the SQA3D dataset [20], accuracy is determined by the average accuracy of answers obtained for six predefined questions. Similarly, our SQA3D Pro dataset also employs EM@1 and EM@10 as the evaluation metrics, consistent with SQA3D.

5.2 Experimental Results

In this section, we compare the performance of 3DGraphVQA on the SQA3D dataset [20] and the ScanQA dataset [3] with various SOTA methods, including SQA3D [20], ScanQA [3], CLIP-PVQA [22], Multi-CLIP [10] and 3D-VisTA[41]. The results are presented in Table 1 and Table 2.

We begin by conducting experiments on the SQA3D dataset. Our method uses the complete reasoning architecture described in Section 3. We set up both question and scene graphs, and both undergo graph reasoning using Graph Transformer. Additionally, in the fusion process, we use a bilinear graph network [13] for inter-graph fusion. As shown in Table 1, SQA3D performs better on *Is* and *Can* questions, but gets poorer results on *What* questions. This is because SQA3D has a simpler reasoning process, making it easier to answer binary questions with fewer answer options. However, due to the comprehensive graph reasoning of our approach, we achieve the best results on *What* and *Which* questions. Particularly, since we also utilize the SQA3D Pro dataset as supplemental image modality for the current scenario, our method also achieved improved performance on the SQA3D dataset. In Figure 6 we also illustrate the qualitative comparison results between our method and SQA3D.

We then conduct experiments on the ScanQA dataset. ScanQA is a question-answering dataset containing only scenes and specific questions, and no prior work has constructed dynamic scene graphs on ScanQA. To better facilitate graph reasoning, we set up only one question graph for graph reasoning. Since we only have one graph, there is no need for inter-graph fusion learning. However, we still aim to achieve appropriate multimodal fusion of text and graph during the multimodal fusion process. To achieve this, we refer to BUTD [2], which is a method that combines bottom-up and top-down attention mechanisms. The bottom-up attention focuses on the visual features themselves using a feed-forward attention mechanism, while the top-down attention relies on textual features to predict attention distribution in visual features. By inputting the node features after graph reasoning and the encoded question description, we enable the question features to receive more attention from matching graph nodes. As shown in Table 2, we have achieved the SOTA performance, especially in EM@10, where our method exceeds other SOTA methods by a large margin.

5.3 Ablation Study

In this section, we conduct various ablation experiments to validate the effectiveness of different modules in our pipeline including scene graph construction, graph reasoning, as well as bilinear graph networks.

Scene Graph Construction. Given that 3DGraphVQA is an end-to-end network architecture where target features and bounding boxes are dynamically acquired, we need to dynamically construct graph nodes and edges. Specifically, for the construction of graph nodes, our method directly takes the objects as the graph nodes

Table 2: The question answering accuracy on ScanQA dataset. Each entry denotes the values of "test w/ object" / "test w/o object".

Method	EM@1	EM@10	BLEU-4	ROUGE	METEOR	CIDEr
Image+MCAN [3]	22.3 / 20.8	53.1 / 51.2	14.3 / 9.7	31.3 / 29.2	12.1 / 11.5	60.4 / 55.6
ScanRefer+MCAN [3]	20.6 / 19.0	52.4 / 49.7	7.5 / 7.8	30.7 / 28.6	12.0 / 11.4	57.4 / 53.4
ScanQA [3]	23.5 / 20.9	56.5 / 54.1	12.0 / 10.8	34.3 / 31.1	13.6 / 12.6	67.3 / 60.2
CLIP-PVQA [22]	23.9 / 21.4	/	14.6 / 11.7	35.2 / 32.4	13.9 / 13.3	69.5 / 62.8
Multi-CLIP [10]	24.0 / 21.5	/	12.7 / 12.9	35.4 / 32.6	14.0 / 13.4	68.7 / **63.2**
3D-Vista [41]	**27.0** / **23.0**	57.9 / 53.5	**16.0** / 11.9	**38.6** / 32.8	**15.2** / 12.9	**76.6** / 62.6
3DGraphQA	25.6 / 22.3	**58.7** / **56.1**	15.1 / **12.9**	36.9 / **33.0**	14.7 / **13.6**	74.6 / 62.9

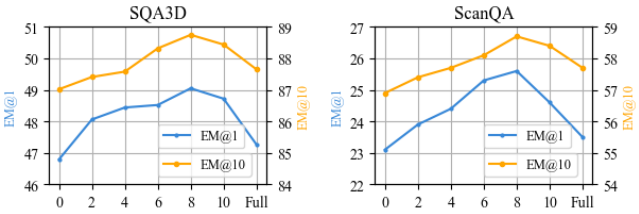

Figure 7: This figure depicts the EM@1 (the left indicator) and EM@10 (the right indicator) results of our method on SCA3D and ScanQA datasets. In this context, the horizontal axis represents the number of edges in the graph, ranging from no edges to 2, 4, 6, 8, 10, and fully connected graphs. The vertical axis represents accuracy, with the values of the left side indicating the accuracy of EM@1, and the values of the right side indicating EM@10.

and incorporates the features of each object as node features in the graph.

For the construction of graph edges, our method initially experiments with unconnected graphs or fully connected graphs. In the unconnected graph setup, we only inject the node features into the graph reasoning process. Conversely, in the fully connected graph setup, all edge weights of the bounding nodes were set to 1s and injected into the graph reasoning process along with the node features. In addition, considering the strong correlation between the problem design of the SQA3D dataset and the surrounding objects of the current perspective, we set edge nodes based on the prior information of each target bounding box. Here, we rank the objects closest to the current graph node's bounding box, and the top-ranked objects are selected within the neighborhood range, meaning the edge weight between this target and the current graph node is set to 1; otherwise, it is set to 0. We set the neighborhood range to 2, 4, 6, 8, and 10. As shown in Figure 7, the best performance is achieved when the neighborhood range is set to 8 targets. On the contrary, when using unconnected or fully connected graphs, the effect of both is relatively inferior.

Moreover, as we employed a dual-graph design for inter-graph processing, consisting of situation-graph and question-graph, we also compare the results with those obtained from a single-graph design. In this setup, we concatenate the situation description feature and question encoding feature into the node features as a single fusion graph. As shown in the table 3, we demonstrated that simultaneous inter-graph processing with situation-graph and question-graph yields superior results.

Table 3: The ablation study of our 3DGraphQA. We show the EM@1 and EM@10 results on SQA3D dataset.

Method	EM@1	EM@10
Graph Tansformer single Graph	48.32	88.09
Graph Tansformer without BAN	48.05	87.41
GAT	47.88	84.97
Graph Tansformer 5L	48.54	88.44
Graph Tansformer 10L	49.04	88.75
Graph Tansformer 10L on SQA3D Pro	**49.18**	**89.23**

Graph Reasoning Methods. There are various methods for reasoning over graph structures in the graph reasoning field. In this work, during the graph reasoning process, our method adopt the former constructed graphs as inputs to the Graph Transformer. Additionally, we conduct ablation experiments with different numbers of layers in the Graph Transformer, i.e. selecting 5 and 10 layers for comparison. As shown in the table, employing Graph Transformer with 10 layers outperformed the other setup.

Bilinear Graph Networks. Finally, we compared the results with and without the module of bilinear graph networks. When discarding bilinear graph networks, we concatenated the node features of the scene graph and question graph together and directly decoded the features. The results demonstrated that without the presence of bilinear graph networks, the accuracy of the answers decreased, which validates the effectiveness of the bilinear graph networks.

6 CONCLUSION

In this paper, we present 3DGraphQA, a novel graph-based 3D question answering method. The key premise of our method is that graph reasoning can facilitate the complex relation reasoning between 3D scene objects. To accomplish this goal, our method first constructs the scene graph based on the scene and injects the situation description and question description separately to form the situation-graph and the question-graph. Subsequently, for the intra-graph and inter-graph feature fusion of the situation-graph and the question-graph, we propose intra-graph message passing based on Graph Transformer and inter-graph message passing based on the Bilinear Attention Network (BAN). Experimental results demonstrate the effectiveness of our proposed approach.

REFERENCES

[1] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas J. Guibas. 2020. ReferIt3D: Neural Listeners for Fine-Grained 3D Object Identification in Real-World Scenes. In *Computer Vision - ECCV 2020 (Lecture Notes in Computer Science, Vol. 12346)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). 422–440.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. Computer Vision Foundation / IEEE Computer Society, 6077–6086.

[3] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. 2022. ScanQA: 3D Question Answering for Spatial Scene Understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*. IEEE, 19107–19117.

[4] Mikhail Belkin and Partha Niyogi. 2003. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Comput.* 15, 6 (2003), 1373–1396.

[5] Daigang Cai, Lichen Zhao, Jing Zhang, Lu Sheng, and Dong Xu. 2022. 3DJCG: A Unified Framework for Joint Dense Captioning and Visual Grounding on 3D Point Clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*. IEEE, 16443–16452.

[6] Dave Zhenyu Chen, Angel X. Chang, and Matthias Nießner. 2020. ScanRefer: 3D Object Localization in RGB-D Scans Using Natural Language. In *ECCV (Lecture Notes in Computer Science, Vol. 12365)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). 202–221.

[7] Dave Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X. Chang. 2021. Scan2Cap: Context-Aware Dense Captioning in RGB-D Scans. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. Computer Vision Foundation / IEEE, 3193–3203.

[8] Dave Zhenyu Chen, Ronghang Hu, Xinlei Chen, Matthias Nießner, and Angel X. Chang. 2023. UniT3D: A Unified Transformer for 3D Dense Captioning and Visual Grounding. In *IEEE/CVF International Conference on Computer Vision, ICCV*. IEEE, 18063–18073.

[9] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. 2017. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. IEEE Computer Society.

[10] Alexandros Delitzas, Maria Parelli, Nikolas Hars, Georgios Vlassis, Sotirios-Konstantinos Anagnostidis, Gregor Bachmann, and Thomas Hofmann. 2023. Multi-CLIP: Contrastive Vision-Language Pre-training for Question Answering tasks in 3D Scenes. In *34th British Machine Vision Conference 2023, BMVC*. BMVA Press, 748–749.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). 4171–4186.

[12] Vijay Prakash Dwivedi and Xavier Bresson. 2020. A Generalization of Transformer Networks to Graphs. *CoRR* abs/2012.09699 (2020).

[13] Dalu Guo, Chang Xu, and Dacheng Tao. 2023. Bilinear Graph Networks for Visual Question Answering. *IEEE Trans. Neural Networks Learn. Syst.* 34, 2 (2023), 1023–1034.

[14] Wencheng Han, Dongqian Guo, Cheng-Zhong Xu, and Jianbing Shen. 2024. DME-Driver: Integrating Human Decision Logic and 3D Scene Perception in Autonomous Driving. *CoRR* abs/2401.03641 (2024).

[15] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (1997), 1735–1780.

[16] Minhyeok Lee. 2023. GELU Activation Function in Deep Learning: A Comprehensive Mathematical Analysis and Performance. *CoRR* abs/2305.12073 (2023).

[17] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. 2021. Less Is More: ClipBERT for Video-and-Language Learning via Sparse Sampling. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. 7331–7341.

[18] Weixin Liang, Yanhao Jiang, and Zixuan Liu. 2021. GraghVQA: Language-Guided Graph Neural Networks for Graph-based Visual Question Answering. *CoRR* abs/2104.10283 (2021).

[19] Bingqian Lin, Yi Zhu, Yanxin Long, Xiaodan Liang, Qixiang Ye, and Liang Lin. 2022. Adversarial Reinforced Instruction Attacker for Robust Vision-Language Navigation. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 10 (2022), 7175–7189.

[20] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. 2023. SQA3D: Situated Question Answering in 3D Scenes. In *ICLR*. OpenReview.net.

[21] Will Norcliffe-Brown, Stathis Vafeias, and Sarah Parisot. 2018. Learning Conditioned Graph Structures for Interpretable Visual Question Answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 8344–8353.

[22] Maria Parelli, Alexandros Delitzas, Nikolas Hars, Georgios Vlassis, Sotiris Anagnostidis, Gregor Bachmann, and Thomas Hofmann. [n. d.]. CLIP-Guided Vision-Language Pre-training for Question Answering in 3D Scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*. 5607–5612.

[23] 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*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 1532–1543.

[24] Charles R. Qi, Or Litany, Kaiming He, and Leonidas J. Guibas. 2019. Deep Hough Voting for 3D Object Detection in Point Clouds. In *IEEE/CVF International Conference on Computer Vision, ICCV*. IEEE, 9276–9285.

[25] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems (NeurIPS)*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5099–5108.

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 8748–8763.

[27] Manolis Savva, Jitendra Malik, Devi Parikh, Dhruv Batra, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, and Vladlen Koltun. 2019. Habitat: A Platform for Embodied AI Research. In *IEEE/CVF International Conference on Computer Vision, ICCV*. IEEE, 9338–9346.

[28] Bruno Souza, Marius Aasan, Hélio Pedrini, and Adín Ramírez Rivera. 2023. Self-GraphVQA: A Self-Supervised Graph Neural Network for Scene-based Question Answering. In *IEEE/CVF International Conference on Computer Vision, ICCV - Workshops*. IEEE, 4642–4647.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems (NeurIPS)*. 5998–6008.

[30] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. 2019. Reinforced Cross-Modal Matching and Self-Supervised Imitation Learning for Vision-Language Navigation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. Computer Vision Foundation / IEEE, 6629–6638.

[31] Yanan Wang, Michihiro Yasunaga, Hongyu Ren, Shinya Wada, and Jure Leskovec. 2023. VQA-GNN: Reasoning with Multimodal Knowledge via Graph Neural Networks for Visual Question Answering. In *IEEE/CVF International Conference on Computer Vision, ICCV*. IEEE, 21525–21535.

[32] Zhecan Wang, Haoxuan You, Liunian Harold Li, Alireza Zareian, Suji Park, Yiqing Liang, Kai-Wei Chang, and Shih-Fu Chang. 2022. SGEITL: Scene Graph Enhanced Image-Text Learning for Visual Commonsense Reasoning. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI*. AAAI Press, 5914–5922.

[33] Zehan Wang, Ziang Zhang, Luping Liu, Yang Zhao, Haifeng Huang, Tao Jin, and Zhou Zhao. 2023. Extending Multi-modal Contrastive Representations. *CoRR* abs/2310.08884 (2023).

[34] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. 2018. Gibson Env: Real-World Perception for Embodied Agents. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. Computer Vision Foundation / IEEE Computer Society, 9068–9079.

[35] Shuquan Ye, Dongdong Chen, Songfang Han, and Jing Liao. 2024. 3D Question Answering. *IEEE Trans. Vis. Comput. Graph.* 30, 3 (2024), 1772–1786.

[36] Ting Yu, Xiaojun Lin, Shuhui Wang, Weiguo Sheng, Qingming Huang, and Jun Yu. 2024. A Comprehensive Survey of 3D Dense Captioning: Localizing and Describing Objects in 3D Scenes. *IEEE Trans. Circuits Syst. Video Technol.* 34, 3 (2024), 1322–1338.

[37] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep Modular Co-Attention Networks for Visual Question Answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 6281–6290.

[38] Zhihao Yuan, Xu Yan, Yinghong Liao, Yao Guo, Guanbin Li, Shuguang Cui, and Zhen Li. 2022. X -Trans2Cap: Cross-Modal Knowledge Transfer using Transformer for 3D Dense Captioning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*. IEEE, 8553–8563.

[39] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 2021. 3DVG-Transformer: Relation Modeling for Visual Grounding on Point Clouds. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV*. IEEE, 2908–2917.

[40] Xi Zhu, Zhendong Mao, Zhineng Chen, Yangyang Li, Zhaohui Wang, and Bin Wang. 2021. Object-difference drived graph convolutional networks for visual question answering. *Multim. Tools Appl.* 80, 11 (2021), 16247–16265.

[41] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 2023. 3D-VisTA: Pre-trained Transformer for 3D Vision and Text Alignment. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, 2899–2909.