ProMo3D: Programmatic Modular Reasoning for 3D Visual Question Answering

Qilin Huang[1,†] Jianuo Zhu[1,†]
[1]Southern University of Science and Technology
[†] equally contributed

Abstract

*Current 3D Visual Question Answering (3D-VQA) approaches, often monolithic in design, typically struggle with interpretability and generalizing to novel reasoning pathways or unseen scene configurations. This inherent limitation hinders their ability to perform robust, multi-step inference effectively. We introduce **ProMo3D**, a neuro-symbolic framework that addresses these challenges by uniquely synergizing the strong generalization capabilities of Large Language Models (LLMs) with the structured reasoning of a specialized 3D Neural Module Network (NMN). ProMo3D leverages a pre-trained LLM (Gemini) for on-the-fly synthesis of symbolic programs directly from natural language questions using prompt engineering. Crucially, this is achieved without requiring any LLM fine-tuning. These LLM-generated programs then orchestrate our 3D-NMN, where each module executes a distinct reasoning step—such as object identification, relational grounding, or attribute querying—within the 3D scene. This decomposition fosters transparent, step-by-step reasoning, thereby enhancing model interpretability and promoting compositional generalization through an explicitly structured inference process. Extensive experiments on the challenging ScanQA benchmark demonstrate that ProMo3D achieves competitive performance against leading methods, while significantly improving model transparency and exhibiting robust generalization, particularly when adapting to new reasoning structures.*

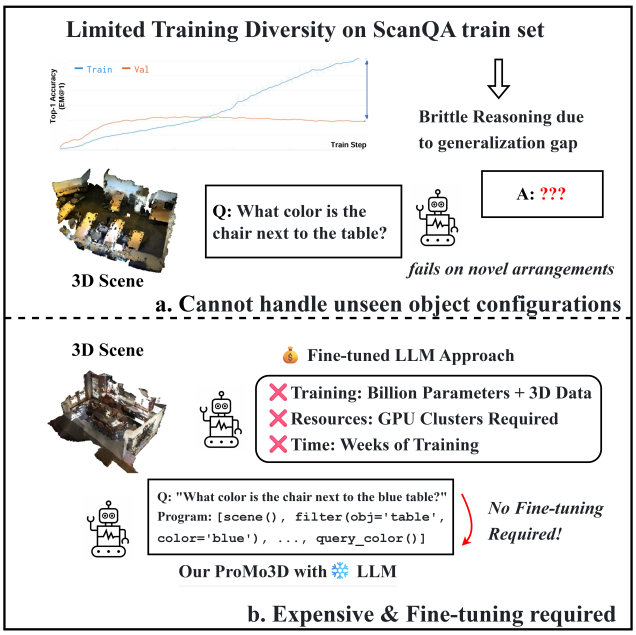

Figure 1. Caveats of current 3D-VQA methods. (a) Traditional monolithic 3D-VQA models struggle with novel object configurations and unseen spatial arrangements due to ScanQA's limited training diversity, leading to brittle reasoning pathways that fail on complex compositional questions. (b) Current approaches fine-tune large language models on specific 3D tasks, requiring substantial computational resources. In ProMo3D, we leverage the strong generalization capabilities of LLMs for on-the-fly program synthesis without fine-tuning, while employing a structured 3D Neural Module Network to execute robust, interpretable reasoning pathways.

1. Introduction

The ability to interpret and reason about complex 3D environments from natural language queries is a pivotal goal in artificial intelligence, essential for applications in robotics, augmented reality, and human-computer interaction [3, 21]. 3D Visual Question Answering (3D-VQA) serves as a demanding testbed for this capability, requiring models to not only perceive 3D objects and their spatial relationships but also perform systematic, multi-step reasoning within semantically rich environments.

Current 3D-VQA systems, often relying on monolithic deep learning architectures [3, 29], tend to falter when questions involve unfamiliar scene arrangements, as illustrated in Fig. 1(a). Training on datasets like ScanQA cannot exhaustively cover all compositional variants, resulting in brit-

tle performance and limited real-world applicability. Furthermore, the "black-box" nature of these models obscures their decision-making process, hindering interpretability and trust. Neural Module Networks (NMNs) [1, 2, 25] offer a promising alternative by decomposing questions into modular operations, thereby fostering interpretability and compositional generalization. Yet, their widespread adoption has been hampered by the difficulty of developing semantic parsers capable of reliably translating diverse natural language questions into accurate, executable programs, especially without extensive, task-specific program annotations [25]. Traditional parsers often lack the requisite flexibility for the linguistic variance in datasets like ScanQA.

The advent of Large Language Models (LLMs) has opened new avenues. Some recent approaches attempt to harness LLM capabilities by fine-tuning them on specific 3D-VQA tasks [6, 16], as depicted in Fig. 1(b). While this can improve reasoning, it often incurs substantial computational costs and requires significant amounts of domain-specific data, making such solutions less accessible and scalable. The challenge, therefore, is to leverage the impressive language understanding and generalization of LLMs without the burdensome overhead of fine-tuning, while still enabling structured and grounded 3D reasoning.

To address these limitations, we introduce **ProMo3D** (**Pro**grammatic **Mo**dular **3D** VQA), a neuro-symbolic framework that uniquely synergizes the strong generalization capabilities of pre-trained LLMs with the structured reasoning of a specialized 3D Neural Module Network. Our central idea is to employ an LLM (specifically, Gemini [24]) as an on-the-fly program synthesizer. Through carefully engineered prompts incorporating few-shot examples, the LLM translates natural language questions into symbolic programs—sequences of functional primitives designed for 3D reasoning (e.g., object identification, relational grounding, attribute querying). Critically, this is achieved *without any LLM fine-tuning*, thereby harnessing the LLM's powerful inherent generalization capabilities while circumventing the associated computational and data demands. The LLM, in essence, provides a robust, high-level reasoning plan.

These LLM-generated programs then orchestrate our 3D-NMN. Each module in the NMN is a learnable operator tailored to execute a specific step in the program, operating directly on 3D object features and intermediate attentional states. This program-guided execution ensures that the reasoning process is explicit, interpretable, and systematically grounded in the visual 3D context. The NMN effectively "regulates" the execution of the reasoning pathway, ensuring that each step is performed with 3D-awareness. This explicit decomposition fosters transparent, step-by-step inference, enhancing model interpretability and promoting compositional generalization by allowing novel combinations of learned modular operations.

Our contributions can be summarized as follows:

- We introduce ProMo3D, a novel neuro-symbolic framework for 3D-VQA that effectively integrates LLM-driven, on-the-fly program synthesis with a specialized 3D Neural Module Network, achieving strong performance without LLM fine-tuning.
- We demonstrate a robust methodology for leveraging pre-trained LLMs via prompt engineering to translate complex natural language questions into structured symbolic programs, exhibiting high parsing flexibility suitable for diverse 3D-VQA queries.
- We develop a tailored suite of 3D-aware neural modules designed for fine-grained reasoning over 3D scene representations; when guided by LLM-generated programs, these modules lead to improved interpretability and compositional generalization on the challenging ScanQA benchmark.

2. Related Work

3D Visual Question Answering. The field of 3D Visual Question Answering has evolved rapidly, with approaches broadly categorized into traditional task-specific methods and recent LLM-based solutions. Early works like ScanQA [3] and SQA3D [21] established foundational benchmarks by extending 2D VQA paradigms to 3D environments, typically employing cross-modal attention mechanisms to fuse visual and linguistic representations. Subsequent methods such as 3D-VisTA [29] and Multi-CLIP [10] developed specialized architectures with auxiliary supervision strategies, incorporating object detection and classification losses to enhance spatial understanding. However, these approaches often struggle with compositional generalization due to their monolithic architectures and reliance on dataset-specific patterns.

Recent advances have leveraged Large Language Models to address these limitations. 3D-LLM [13] pioneered the integration of LLMs with 3D scene understanding by introducing positional embeddings and location tokens, while LL3DA [6] employed Q-Former [9, 19] architectures to bridge point clouds with language instructions. LEO [16] and Scene-LLM [12] further refined this paradigm through sophisticated multi-view feature fusion and two-stage training schemes. Chat-Scene [15] achieved precise object referencing by incorporating object identifiers into 3D LMMs. While these LLM-based methods demonstrate improved reasoning capabilities, they require substantial computational resources for fine-tuning billion-parameter models on domain-specific data, limiting their practical scalability. Our approach uniquely addresses this limitation by leveraging frozen LLMs for program synthesis, achieving competitive performance without fine-tuning overhead.

Neural Module Networks. Neural Module Networks [2]

represent a foundational approach to compositional visual reasoning, decomposing complex questions into sequences of specialized neural operations. The original NMN framework introduced modular architectures where each module implements a specific reasoning primitive, enabling systematic composition for diverse query types. Subsequent developments enhanced this paradigm: NS-VQA [25] replaced neural modules with symbolic execution engines for improved interpretability, while Meta-Module Networks [7] introduced adaptive modules capable of adjusting to novel sub-tasks through attention mechanisms.

For systematic generalization, Vector-NMN [4] achieved state-of-the-art performance by incorporating image features into each module's input and employing vector-based representations. More recent work has explored integrating attention mechanisms from Transformer architectures into modular designs [28], demonstrating that combining structured reasoning with modern attention mechanisms can improve both interpretability and generalization capabilities. However, a persistent challenge in NMN deployment has been the development of robust semantic parsers capable of translating natural language into executable programs, particularly without extensive program annotations [25].

Traditional semantic parsing approaches often require task-specific training data and struggle with linguistic diversity [18]. Recent efforts have explored end-to-end differentiable parsing [14] and weakly supervised methods [1], but these approaches remain limited in their ability to handle the complexity and variance found in challenging datasets like ScanQA. Our work addresses this fundamental limitation by leveraging the strong instruction-following capabilities of modern LLMs as zero-shot program synthesizers, eliminating the need for specialized parser training while maintaining the interpretability and compositional benefits of modular architectures. This represents a significant advancement in making NMN-based approaches practical for complex 3D reasoning tasks.

3. Method

Existing 3D Visual Question Answering (VQA) systems often falter when faced with questions requiring novel compositional reasoning or generalization to unseen 3D environments. Their monolithic architectures typically lack explicit reasoning mechanisms, making it difficult to trace errors or adapt to diverse queries. To overcome these limitations, **ProMo3D** introduces a neuro-symbolic framework that decouples complex question understanding from visual grounding and reasoning. As depicted in Figure 2, our core strategy is to first leverage the strong generalization and instruction-following capabilities of a pre-trained Large Language Model (LLM) to dynamically synthesize symbolic programs from natural language questions. These programs then orchestrate a specialized 3D Neural Module Network (NMN), which executes the program to derive the answer through a sequence of interpretable, grounded operations on 3D scene representations.

The methodology of ProMo3D encompasses three key stages: (1) On-the-fly symbolic program synthesis via an LLM, detailed in Section 3.1; (2) Representation of the 3D scene and linguistic concepts, which provide the grounding for reasoning (Section 3.2); and (3) Program-guided execution using our 3D Neural Module Network, which is trained end-to-end while the LLM remains fixed (Section 3.3).

3.1. LLM-Powered Program Synthesis

A central challenge in 3D-VQA is guiding models to perform accurate, multi-step reasoning, especially when encountering novel object configurations or question structures unseen during training. While traditional semantic parsers for Neural Module Networks (NMNs) often struggle with the linguistic diversity of datasets like ScanQA [3] and require extensive program annotations, recent Large Language Models (LLMs) exhibit remarkable proficiency in understanding and decomposing natural language. Our approach, ProMo3D, capitalizes on this by employing Gemini 2.0 [24] as an on-the-fly program synthesizer. Given an input natural language question Q, the LLM, guided by a carefully engineered prompt incorporating few-shot examples and our Domain-Specific Language (DSL) definition, translates Q into an executable symbolic program P. This process leverages the LLM's strong generalization capabilities purely at inference time, without any LLM fine-tuning, thus avoiding substantial computational costs and dataset dependencies associated with many LLM-based QA methods, while still enabling the model to operate correctly on unseen scenes and complex queries.

Our DSL is carefully constructed to encompass the core reasoning capabilities essential for 3D-VQA. Rather than a monolithic function, it provides a vocabulary of composable primitives that allow for nuanced interaction with the 3D scene representation. These primitives can be broadly categorized: **Object-Centric Operations** form the foundation, enabling the system to first identify and isolate relevant entities within the 3D scene. This includes functions to select all objects (`Scene`), filter them based on semantic attributes like category, color, or material (e.g., `filter(..., chair)`). **Relational Reasoning Primitives** allow the model to understand and verify relationships between objects or object sets. This covers spatial relations (e.g., `relate_pos(..., left)`) and comparative attributes such as size (e.g., `relate_size(..., larger)`). **Logical and Set Operations** facilitate the composition of more complex queries by enabling operations like intersection (`intersect`) or union (`or`) over sets of identified objects, refining the attentional focus. Finally, **Attribute Querying Functions** allow the model to

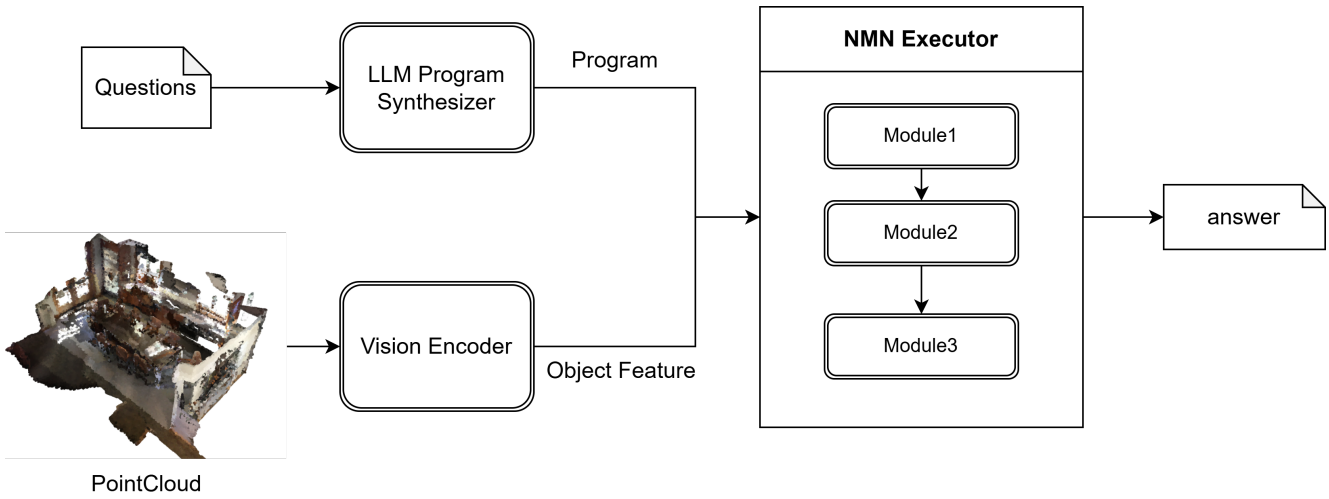

Figure 2. **Overall architecture of ProMo3D.** (1) A natural language question is input to the LLM Program Synthesizer, which generates a symbolic program. (2) The 3D scene is processed for object features, and language concepts are embedded. (3) The NMN Executor sequentially invokes neural modules corresponding to program instructions. (4) The final module outputs answer logits.

extract specific information about the finalized set of attended objects, such as their color (`query_color`), material (`query_material`).

The output of this synthesis stage is a program $P = (I_0, I_1, \ldots, I_L)$, a sequence of instructions where each instruction I_k specifies a DSL function f_k. Crucially, I_k also contains references to the outputs of prior instructions, defining the dataflow dependencies, and any textual arguments (e.g., "cabinet," "sink") extracted directly from the original question Q. This symbolic program serves as an explicit, interpretable reasoning blueprint, guiding the subsequent 3D Neural Module Network to follow an accurate reasoning pathway. Further details on the prompt design and the formal DSL specification are available in the Appendix.

3.2. Multi-Modal Feature Encoding

To ground the reasoning process dictated by the LLM-generated program, ProMo3D requires robust representations of both the 3D visual scene and relevant linguistic concepts. For each input 3D scene, an object-centric representation is derived using a pre-trained PointNet++ based backbone [22]. This yields a set of O object features, $\mathbf{F}_{obj} \in \mathbb{R}^{O \times D_{feat}}$, which encapsulate the geometric and semantic properties of individual objects within the scene. Concurrently, the natural language question Q is encoded via a pre-trained BERT model [11] into a global question embedding $\mathbf{e}_{ques} \in \mathbb{R}^{D_{lang}}$. Similarly, textual arguments embedded within the program instructions (e.g., specific object categories like "chair" or relational terms like "left of," as identified during program synthesis) are encoded into concept embeddings $\mathbf{e}_{concept} \in \mathbb{R}^{D_{lang}}$. These distinct representations—the visual object features $\mathbf{F}_{obj}$, the global question embedding $\mathbf{e}_{ques}$, and the instruction-specific concept embeddings $\mathbf{e}_{concept}$—serve as the foundational inputs for the NMN execution stage.

3.3. NMN for Guided 3D Reasoning

Our 3D Neural Module Network (NMN) executes the LLM-synthesized program $P = (I_0, \ldots, I_L)$ through modular, compositional reasoning. Each instruction I_k invokes a specialized neural module M_k that progressively refines object representations $\mathbf{F}_{\text{out}}^{(k)} \in \mathbb{R}^{O \times D}$ within the 3D scene. Figure 3 provides a visual example of this program-guided execution, demonstrating how they chain together to process information and arrive at an answer.

Execution Flow. Program execution begins with a `Scene` module that enriches initial object features $\mathbf{F}_{obj}$ with positional and token type embeddings. Each subsequent module M_k takes refined features from prior modules, object masks for padding handling, and concept embeddings $\mathbf{e}_{concept}^{(k)}$ from instruction arguments. The dataflow dependencies, specified in each instruction I_k, enable complex reasoning chains where object representations are progressively contextualized.

Module Architecture. All modules employ lightweight transformer blocks with multi-head attention and feed-forward networks. Token type embeddings distinguish different input modalities within attention mechanisms. Our module library implements four key operation types:

- **Filtering:** Cross-modal attention between language concepts and objects, refined through transformer blocks.
- **Relational:** Spatial relationship grounding via cross-modal attention and transformer-based reasoning.

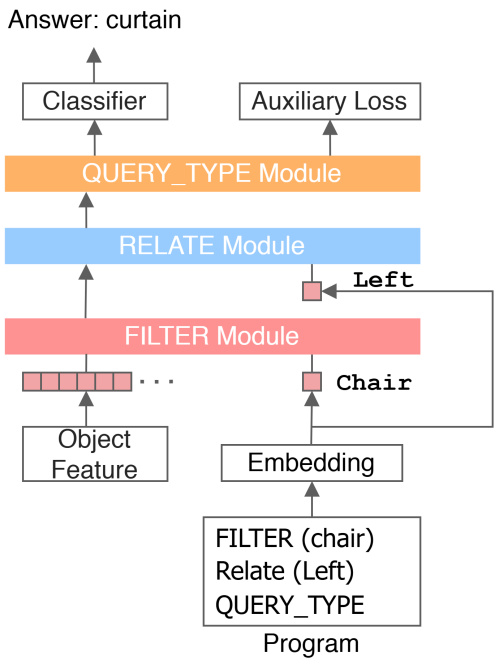

Figure 3. Execution flow of our 3D Neural Module Network (NMN) for an example program: `FILTER (chair)`, `Relate (Left)`, `QUERY_TYPE`. The diagram illustrates how initial object features are sequentially processed. First, the `FILTER Module` utilizes an embedding of the concept 'chair'. Its output is then passed to the `RELATE Module`, which is guided by an embedding of the concept 'Left'. Finally, the `QUERY_TYPE Module` processes these refined features, and a `Classifier` produces the answer (e.g., 'curtain'). This visualizes the modular execution where program instructions direct neural modules to progressively refine object representations for reasoning.

- **Logical:** Set operations (e.g.: `intersect`, `or` and `unique`) through transformer-based feature processing with specialized embeddings.
- **Query:** Joint transformer processing of objects, text, and learnable answer tokens for final classification.

Answer Generation. The final `Query` module processes refined object features alongside textual representations through a unified transformer architecture. Using learnable answer tokens and multi-modal attention, it produces classification logits $\mathbf{z}_{\text{ans}} \in \mathbb{R}^{C_{\text{ans}}}$ for the final answer.

This modular execution provides an interpretable reasoning pathway where each program step corresponds to a specific neural operation, enabling systematic question decomposition and complex, non-linear reasoning chains.

3.4. Learning Paradigm

The learnable parameters Θ of ProMo3D, which comprise the 3D NMN modules and the associated 3D object and question encoders, are optimized end-to-end. Crucially, the LLM-based program synthesizer remains fixed and is not updated during this training phase, functioning solely in inference mode. The primary training objective is to accurately answer visual questions, formulated as minimizing a cross-entropy loss $\mathcal{L}_{\text{VQA}}$:

$$\mathcal{L}_{\text{VQA}}(\Theta) = -\mathbb{E}_{(Q,S,a_{gt})\sim\mathcal{D}} \left[\log p(a_{gt}|Q,S,P;\Theta)\right] \quad (1)$$

where (Q, S, a_{gt}) denotes a triplet from the training dataset $\mathcal{D}$, P is the LLM-generated program for question Q, and $p(a_{gt}|Q,S,P;\Theta)$ is the probability assigned by ProMo3D to the ground-truth answer a_{gt}.

To further enhance representation learning and promote generalization, ProMo3D incorporates auxiliary multi-task learning. Features derived during NMN execution, such as final attended object features or intermediate attention distributions, are channeled to auxiliary prediction heads. These heads are trained on related tasks, for instance, object grounding or predicting descriptive attributes of attended objects. Each auxiliary task j introduces a loss term $\mathcal{L}_{\text{aux},j}(\Theta)$. The comprehensive training objective is a weighted sum of these losses:

$$\mathcal{L}_{\text{total}}(\Theta) = \mathcal{L}_{\text{VQA}}(\Theta) + \sum_j \lambda_j \mathcal{L}_{\text{aux},j}(\Theta) \quad (2)$$

where λ_j are hyper-parameters that balance the contribution of each auxiliary task. This multi-task framework encourages the NMN to learn robust and versatile representations beneficial beyond the primary VQA task.

4. Experiments

In this section, we conduct extensive evaluations to examine ProMo3D's capacity for interpretable 3D visual reasoning. We begin by introducing datasets, evaluation metrics, and implementation details (Sec. 4.1). We then compare ProMo3D against state-of-the-art methods across different categories (Sec. 4.2), conduct thorough ablation studies on model design and training strategies (Sec. 4.3).

4.1. Experimental Setup

Datasets. We evaluate ProMo3D on the challenging ScanQA dataset [3], which contains 41,363 question-answer pairs across 800 indoor 3D scenes from ScanNet [8]. The dataset covers diverse reasoning types including object identification, spatial relationships, counting, and attribute queries. Following established protocols [3], we use the official train/validation split with 33,370 training and 8,893 validation samples. The 3D scenes contain rich annotations including object bounding boxes, semantic labels, and point cloud data.

Evaluation Metrics. We adopt multiple complementary metrics to comprehensively evaluate answer quality.

Table 1. **Comprehensive performance comparison on ScanQA validation set.** Our method, ProMo3D, achieves competitive performance without requiring LLM fine-tuning. Methods with numerically grayed-out scores typically involve fine-tuned LLMs.

	ScanQA (Val)				
	EM@1	BLEU-1	ROUGE-L	METEOR	CIDEr
Task-specific fine-tuned LLMs					
3D-LLM (FlanT5) [13]	20.5	-	35.7	14.5	69.4
LL3DA [6]	-	-	37.3	15.9	76.8
LEO [16]	24.5	-	**49.2**	20.0	**101.4**
Scene-LLM [12]	**27.2**	-	40.0	16.6	80
Task-specific models					
Votenet [23]+MCAN [26]	17.3	28.1	29.8	11.4	54.7
ScanRefer [5]+MCAN [26]	18.6	26.9	30.0	11.5	55.4
ScanQA [3]	21.1	30.2	33.3	13.1	64.9
3D-VLP [17]	21.7	30.5	34.5	13.5	67.0
FE-3DGQA [27]	22.3	-	-	-	-
3D-VisTA [29]	**22.4**	-	35.7	13.9	**69.6**
Zero-shot 2D LLMs					
VideoChat2 [20]	19.2	-	28.2	9.5	49.2
ProMo3D (Ours)	22.3	**31.3**	**36.6**	**20.7**	68.3

EM@1 (Exact Match accuracy) serves as our primary metric, measuring the percentage of predictions that exactly match ground-truth answers. Given ScanQA's free-form, long-text answers, we additionally report text similarity metrics: **BLEU-1** for n-gram overlap, **ROUGE-L** for longest common subsequence matching, **METEOR** for semantic alignment with synonyms and paraphrases, and **CIDEr** for consensus-based evaluation. These metrics collectively assess both exact correctness and semantic quality of generated responses.

Implementation Details. Our 3D object features are extracted using a pre-trained PointNet++ [22] backbone, producing 768-dimensional object representations. Question and concept embeddings utilize BERT-base [11] with matching 768-dimensional outputs for feature alignment. The NMN modules employ lightweight 2-layer transformer blocks with 4-head attention mechanisms and 768-dimensional hidden states. Query modules use 4 layers to handle final reasoning complexity. For LLM-based program synthesis, we employ Gemini-2.0-Flash-Thinking-Exp-01-21 with carefully engineered prompts containing 2 diverse few-shot examples covering spatial reasoning, and attribute queries tasks.

Training Configuration. We train ProMo3D end-to-end for 30 epochs using AdamW optimizer with learning rate 1e-4, weight decay 0.01, and batch size 64. Following 3D-VisTA conventions [29], we incorporate auxiliary losses for enhanced representation learning: object localization ($\mathcal{L}_{objloc}$), object classification ($\mathcal{L}_{objcls}$), and text classification ($\mathcal{L}_{txtcls}$). The total loss is: $\mathcal{L} = \mathcal{L}_{VQA} + \mathcal{L}_{objloc} + \mathcal{L}_{objcls} + \mathcal{L}_{txtcls}$. Training is conducted on single NVIDIA A100 GPU with 40GB memory, requiring approximately 15 hours. The LLM program synthesizer remains frozen throughout training, operating purely in inference mode without requiring computational resources during training.

NMN Architecture	EM@1	M	C
Attention-based (mask-only)	20.3	18.5	-
Simplified (1-layer modules)	21.6	20.2	66.5
Full model (2+4 layers)	**22.3**	**20.6**	**68.3**

Table 2. **Ablation study on neural module network architecture.** We compare different module designs: attention-based modules that output attention masks instead of refined object features, simplified modules with uniform 1-layer transformers, and our full architecture with 2-layer standard modules and 4-layer query modules. **M** and **C** denote METEOR and CIDEr scores respectively. Results on ScanQA validation set.

4.2. Comparison with SoTA Methods

Table 1 presents comprehensive comparisons on ScanQA validation set. We categorize existing methods into three groups to provide fair evaluation:

Task-specific Fine-tuned LLMs (shown in gray) require substantial computational resources and domain-specific fine-tuning of large language models. While these methods achieve high exact match accuracy, they involve significant training overhead and may suffer from overfitting to specific

question patterns.

Task-specific Models including our approach, develop specialized architectures without LLM fine-tuning. These methods balance performance with computational efficiency, making them more practical for real-world deployment.

Zero-shot Methods apply pre-trained models directly without task-specific training, representing the most resource-efficient but typically lower-performing category.

ProMo3D achieves highly competitive performance in the task-specific category, obtaining 22.3% EM@1 while significantly outperforming existing methods in text quality metrics. Notably, our METEOR score of 20.7 substantially exceeds the previous best task-specific method (3D-VisTA: 13.9) by +6.8 points, and our ROUGE-L score of 36.6 surpasses 3D-VisTA [29]'s 35.7 by +0.9 points. This indicates that while some fine-tuned LLM approaches may achieve higher exact match accuracy, ProMo3D's neuro-symbolic design allows it to generate answers with better semantic alignment to the ground truth, all without the significant computational overhead associated with fine-tuning large language models. This is a key advantage that showcases the efficiency and effectiveness of our LLM-guided NMN framework.

4.3. Ablation Studies

We conduct systematic ablation studies to validate our design choices and analyze the contribution of key components in ProMo3D.

NMN Architecture Analysis. Table 2 examines critical architectural decisions in our neural module network. We compare three variants: (1) Attention-based (mask-only): modules output attention masks for subsequent processing instead of refined object features; (2) Simplified (1-layer modules): all modules use uniform single transformer layers; (3) Full model (2+4 layers): our design with 2-layer standard modules and 4-layer query modules for enhanced reasoning capacity.

The results demonstrate several key insights: Rich feature processing is essential, as attention-based modules underperform by -1.9 points EM@1, indicating that refined object features provide superior reasoning foundation compared to simple attention mechanisms. Adequate model capacity matters significantly—our full architecture with deeper query modules outperforms the simplified version by +0.7 points EM@1 and +1.8 points CIDEr, justifying the computational overhead for complex multi-step reasoning tasks.

4.4. Analysis and Discussion

While quantitative metrics (e.g., EM@1, METEOR, ROUGE-L, CIDEr) demonstrate that ProMo3D achieves competitive performance against state-of-the-art baselines,

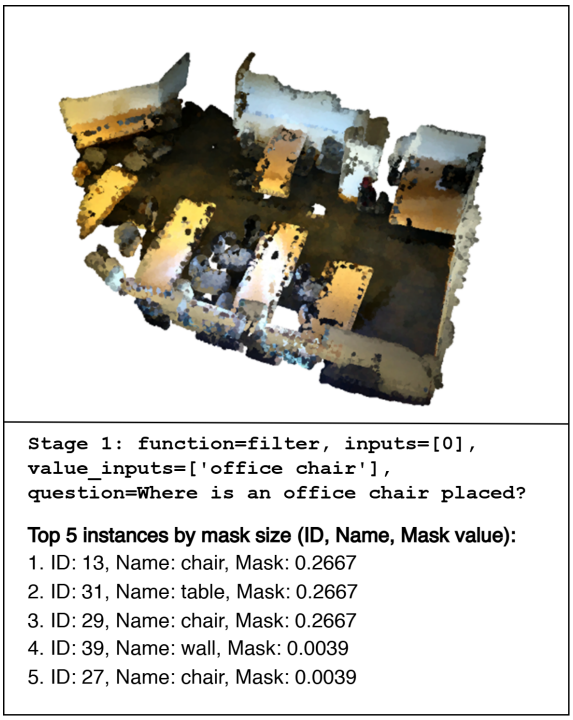

Figure 4. Visualization of intermediate NMN outputs. This fighre shows the mask output of the `FilterModule` when filtering for "office chair", which contains many other objects

a closer examination of intermediate module outputs reveals that non-negligible errors persist throughout the reasoning pipeline. In particular, the visualization of attention maps and object filtering masks indicates that certain modules—especially those responsible for spatial grounding and attribute filtering—produce suboptimal attention distributions under complex question structures. As a result, these intermediate inaccuracies can propagate downstream, ultimately limiting final answer quality.

Concretely, Figure 4 shows a representative example of an intermediate NMN output for the `FilterModule`. In this visualization, the module is tasked with isolating all objects of category *"office chair"*. Although the module correctly identifies several chairs in the scene (e.g., ID: 13 and ID: 29), it also erroneously assigns high mask values to non-chair objects such as a table (ID: 31) and even attends to portions of the wall (ID: 39). The "Top 5 instances by mask size" list (below the point-cloud rendering) confirms that the module's mask values for chairs (Mask = 0.2667) are tied with a table (Mask = 0.2667), and that some chairs receive only weak activations (e.g., ID: 27, Mask = 0.0039). This noisy filtering behavior indicates that, even when the final answer occasionally aligns with ground truth, the intermediate reasoning step can suffer from substantial ambiguities, especially in cluttered 3D scenes where geometric and semantic boundaries overlap.

Despite these intermediate errors, ProMo3D's overall design—i.e., coupling an LLM-synthesized program with explicit neural modules—nevertheless provides a more transparent reasoning trace than monolithic architectures. In particular, one can pinpoint precisely which module introduced the largest deviation from the expected attention distribution, and thereby focus future improvements on that module's architecture or training regimen. Moreover, our ablation study (Table 1) confirms that (i) using rich 3D object features is critical for capturing spatial relationships that 2D-only variants miss, (ii) increasing transformer depth in each module yields consistently better alignment between linguistic concepts and object sets, and (iii) the addition of auxiliary losses (e.g., object localization, object classification) regularizes intermediate representations. Nevertheless, the residual errors highlighted in Figure 4 indicate that more robust supervision—particularly for intermediate NMN outputs—is needed to further reduce error propagation and improve final answer accuracy.

5. Future Work

Despite encouraging results, there remains significant headroom for improving both accuracy and interpretability. We outline several complementary research directions.

5.1. Weak Supervision for Module Alignment

Hand-annotating every intermediate mask or relation is infeasible. Instead, we plan to derive *weak labels* from automatically generated heuristics—for example, projecting 3D bounding boxes into 2D and using off-the-shelf 2D detectors to approximate "visible object" masks, or leveraging CAD model metadata (size, material) to create noisy attribute labels. These signals can regularise corresponding `filter` and `query` modules through a multi-task loss while remaining cheap to obtain.

5.2. Semi-supervised Learning with Unlabeled QA

ScanQA offers thousands of unannotated question–answer pairs in the wild. We intend to:

1. Generate pseudo-programs via the current LLM prompt.
2. Run our NMN to obtain intermediate attention maps and answers.
3. Apply consistency regularization between multiple stochastic forward passes (e.g. dropout, point jittering).

This "self-training" loop can refine both the parser prompt and the module parameters, exploiting the abundance of unlabelled 3D scans.

5.3. Active Learning of Hard Cases

Because visual-spatial edge cases dominate our error taxonomy, an *active learning* loop can query annotators only for scenes/questions where model uncertainty (entropy of answer logits *and* disagreement across intermediate modules) is highest. Targeted annotation of a few hundred critical samples may outperform indiscriminate labelling of thousands, especially for spatial-relation supervision.

5.4. Knowledge Distillation

Large 2D-VLMs such as GPT-4V or Gemini Pro display superior attribute recognition. We can distil their predictions on rendered multi-view images into our 3D modules, enforcing cross-modal agreement losses that teach the NMN to mimic reliable 2D reasoning while retaining full-scene 3D context.

5.5. Reinforcement Learning for Program

Finally, the rigid LLM-generated DSL can be fine-tuned *without* updating LLM weights by treating program tokens as actions in a non-differentiable environment. Using the final EM or CIDEr as sparse rewards, we can employ policy-gradient (REINFORCE) or self-critical sequence training to bias the parser toward shorter, less ambiguous programs that maximise downstream answer correctness.

Beyond these directions, integrating cross-scene self-supervision (contrastive learning across different rooms), leveraging synthetic 3D scene generators for data augmentation, and exploring graph-structured modules for long-range object relations also offer promising avenues to push ProMo3D toward human-level 3D visual reasoning.

6. Conclusion

We presented **ProMo3D**, a neuro-symbolic framework that addresses key limitations in 3D Visual Question Answering by synergizing frozen LLM capabilities with structured neural reasoning. Our approach leverages pre-trained LLMs for on-the-fly program synthesis without fine-tuning, while employing specialized 3D Neural Module Networks to execute interpretable, step-by-step reasoning pathways.

Experimental evaluation on the challenging ScanQA benchmark demonstrates that ProMo3D achieves competitive performance (22.3% EM@1) while significantly outperforming existing task-specific methods in semantic quality metrics (+6.8 points METEOR over 3D-VisTA). Our systematic ablation studies validate the importance of rich feature processing and adequate model capacity for complex 3D reasoning tasks.

The key strength of ProMo3D lies in its ability to provide transparent reasoning traces without the computational burden of LLM fine-tuning, making it both interpretable and practical for real-world deployment. This work demonstrates that careful integration of symbolic and neural approaches can achieve strong performance while maintaining transparency and computational efficiency, opening promising directions toward interpretable and generalizable 3D understanding.

References

[1] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. *arXiv preprint arXiv:1601.01705*, 2016. 2, 3

[2] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39–48, 2016. 2

[3] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19129–19139, 2022. 1, 2, 3, 5, 6

[4] Dzmitry Bahdanau, Harm de Vries, Timothy J. O'Donnell, Shikhar Murty, Philippe Beaudoin, Yoshua Bengio, and Aaron C. Courville. CLOSURE: assessing systematic generalization of CLEVR models. arXiv preprint, arXiv:1912.05783v2, 2020. 3

[5] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European Conference on Computer Vision*, pages 202–221. Springer, 2020. 6

[6] Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. Ll3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26428–26438, 2024. 2, 6

[7] Wenhu Chen, Zhe Gan, Linjie Li, Yu Cheng, William Wang, and Jingjing Liu. Meta module network for compositional visual reasoning. In *Proc. of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV 2021)*, pages 655–664, 2021. 3

[8] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 5

[9] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 2

[10] Alexandros Delitzas, Maria Parelli, Nikolas Hars, Georgios Vlassis, Sotirios Anagnostidis, Gregor Bachmann, and Thomas Hofmann. Multi-clip: Contrastive vision-language pre-training for question answering tasks in 3d scenes. *arXiv preprint arXiv:2306.02329*, 2023. 2

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019. 4, 6

[12] Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual understanding and reasoning. *arXiv preprint arXiv:2403.11401*, 2024. 2, 6

[13] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023. 2, 6

[14] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *Proc. of the 16th IEEE International Conference on Computer Vision (ICCV 2017)*, pages 804–813, 2017. 3

[15] Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, et al. Chat-scene: Bridging 3d scene and large language models with object identifiers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2

[16] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023. 2, 6

[17] Zhao Jin, Munawar Hayat, Yuwei Yang, Yulan Guo, and Yinjie Lei. Context-aware alignment and mutual masking for 3d-language pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10984–10994, 2023. 6

[18] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. In *Proc. of the 16th IEEE International Conference on Computer Vision (ICCV 2017)*, 2017. 3

[19] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 2

[20] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. Mvbench: A comprehensive multi-modal video understanding benchmark. *arXiv preprint arXiv:2311.17005*, 2024. 6

[21] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022. 1, 2

[22] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 4, 6

[23] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 6

[24] Gemini Team. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2025. 2, 3

[25] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B Tenenbaum. Neural-symbolic

VQA: Disentangling Reasoning from Vision and Language Understanding. In *NeurIPS*, 2018. 2, 3

[26] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

[27] Lichen Zhao, Daigang Cai, Jing Zhang, Lu Sheng, Dong Xu, Rui Zheng, Yinjie Zhao, Lipeng Wang, and Xibo Fan. Towards explainable 3d grounded visual question answering: A new benchmark and strong baseline. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022. 6

[28] Huasong Zhong, Jingyuan Chen, Chen Shen, Hanwang Zhang, Jianqiang Huang, and Xian-Sheng Hua. Self-adaptive neural module transformer for visual question answering. *IEEE Transactions on Multimedia*, 23:1264–1273, 2021. 3

[29] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2911–2921, 2023. 1, 2, 6, 7