CAIRN: Cross-Room 3D Scene Understanding with Topology-Aware Large Multimodal Models

He Liang[1]* **Chenyang Ma**[1] **Yiming Zhang**[3] **Sangyun Shin**[1]
Andrew Markham[1] **Niki Trigoni**[1] **Yuhang He**[2]†

[1]University of Oxford [2]Microsoft Research [3]Simon Fraser University
{he.liang, chenyang.ma, sangyun.shin, andrew.markham, niki.trigoni}@cs.ox.ac.uk
yuhanghe@microsoft.com yza440@sfu.ca

Project Page: https://oceansdepp.github.io/cairn_web/

Abstract

Existing 3D scene-grounded Large Language Models (3D-LLMs) focus on answering questions grounded in simplified single-room 3D scenes, lacking the ability to reason over real-world household environments containing multiple interconnected rooms and diverse object categories. We introduce **CAIRN**, a topology-aware 3D-LLM for multi-room 3D scene understanding. CAIRN aligns transformer attention with scene hierarchy, giving the model explicit awareness of object-level relations and room-level connectivity. It enriches object tokens with room-local relational context via a graph neural network, introduces learned room tokens for room-level abstraction, and applies a hierarchical attention mask with geometric bias to route information according to scene topology. CAIRN is developed on **CAIRN-MR**, a benchmark we introduce on HM3D for multi-room 3D scene understanding, covering grounding, captioning, and four question-answering tasks that progressively evaluate from intra-room perception to cross-room reasoning. Experiments show that CAIRN outperforms prior 3D-LLMs by a large margin across all CAIRN-MR tasks while remaining competitive on five single-room benchmarks.

1 Introduction

A capable home assistant agent should recall where an object was last seen in the kitchen and retrieve it from another room, or answer queries that relate objects across spaces (e.g., comparing furniture arrangements across bedrooms). This requires understanding the full 3D scene structure of a household and performing consistent reasoning over the entire multi-room environment.

There has been a rich history of prior work on developing agents that answer questions grounded in 3D scenes, using large-language models (LLMs) augmented with structured 3D scene representations (i.e., 3D-LLMs) as the reasoning backbone [28, 29, 31, 32, 57, 63]. However, these approaches largely restrict 3D scene understanding and spatial reasoning to a single room [8, 10, 20–22, 43, 60], representing scenes as flat sequences of object tokens without modeling higher-level room structure. In this work, we aim to move beyond the single-room paradigm and develop 3D-LLMs that can understand full household settings [42], where multiple interconnected rooms contain hundreds of objects. This unlocks more challenging tasks such as grounding across spatial contexts, tracing relationships between rooms, and reasoning over a scene's topological structure.

*Part of this work was done during an internship at Microsoft Research.
†Corresponding Author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

To enable the study of multi-room 3D scene understanding, we first introduce **CAIRN-MR**, a benchmark for structured multi-room 3D scene understanding built on HM3D [44], as existing 3D scene-language datasets [1, 5, 11, 37, 61] are mostly limited to single-room scans. Unlike prior benchmarks where reasoning is confined to a single room with a limited candidate set, CAIRN-MR shifts the problem to full-scene reasoning over multi-room environments, where models must search over larger spaces, handle similar object configurations, and resolve ambiguities across spatially separated regions. Crucially, the target region is not explicitly specified, requiring models to first localize the relevant room(s) from implicit object-level spatial constraints before performing downstream reasoning. Motivated by these challenges, CAIRN-MR preserves grounding and captioning tasks from prior benchmarks for backward compatibility, while further introducing four question answering tasks that progressively evaluate multi-room reasoning under increasingly challenging settings, ranging from room-grounded perception to cross-room comparison.

Developing 3D-LLMs for multi-room 3D scene understanding introduces challenges beyond the single-room setting. First, existing 3D-LLMs represent a scene as a flat set of object tokens with self-attention over all object pairs [8, 21, 22, 58, 60]. While this design is adequate for single-room scenes, it is poorly suited to multi-room environments, where interactions are primarily intra-room, and cross-room dependencies are sparse and structured by room connectivity. Second, multi-room scenes contain geometric and topological priors, such as relative object poses, distances, and room adjacency, that should be explicitly modeled, but are not exploited by standard self-attention. To address these challenges, we propose **CAIRN**, a topology-aware 3D-LLM that explicitly models the hierarchical structure and topology of multi-room 3D scenes. CAIRN first encodes each scene as a hierarchical 3D scene graph, capturing both object-level relations and room-level connectivity. After tokenizing this graph into scene tokens that preserve both object- and room-level information, CAIRN uses masked attention to handle the sparse and structured interactions in multi-room scenes by constraining information flow according to scene topology. It further incorporates geometric bias into attention to encode explicit spatial priors, such as relative poses, distances, and room adjacency. In summary, our contributions are as follows:

- We present the first study of 3D-LLMs for multi-room scene understanding. We introduce **CAIRN-MR**, a benchmark for multi-room 3D scene understanding that covers grounding, captioning, and four question-answering task types spanning intra-room perception to cross-room comparison.
- We propose **CAIRN**, a topology-aware 3D-LLM for multi-room understanding that uses masked attention to align information flow with scene topology and route information across rooms, and uses geometric bias to inject explicit spatial priors into attention.
- Extensive experiments show that CAIRN substantially outperforms prior methods on CAIRN-MR across most tasks, while remaining effective on five single-room 3D scene understanding benchmarks without compromising fine-grained object-level understanding. We further validate the quality of CAIRN-MR through human calibration and statistical baselines.

2 Related Work

3D-LLMs and Benchmarks. Most 3D-LLMs are developed on single-room benchmarks from ScanNet [1, 5, 11, 13, 37, 61], and thus represent scenes as flat sets of object tokens with dense self-attention. Some adopt object-level tokenization with unique identifiers [10, 20, 21], while others improve scene reasoning through training strategies [8, 14, 22], data [26, 43, 52], or input design [18, 33, 34, 48, 53, 64]. Some incorporate spatial relations via graph networks [58, 60]. However, none models rooms as semantic units or routes information by scene topology. While some recent efforts extend to larger scenes [64], they often construct queries with globally unique targets, effectively reducing the ambiguity inherent in multi-room environments. These settings primarily evaluate large-scale perception rather than structured multi-room reasoning, and do not capture key structural properties such as implicit room hierarchy, recurring object categories across rooms, and layout ambiguity. To our knowledge, CAIRN is the first 3D-LLM to introduce room-level abstraction and topology-aware attention, together with a multi-room benchmark for evaluation.

Multi-Room Embodied Tasks. Simulators that provide multi-room household environments [27, 40, 42] enable challenging multi-room tasks such as human-robot collaboration [4, 41], long-horizon robot task planning [17, 35, 36, 45], and navigation [51, 55]. These works typically rely on ground-truth maps or pre-built scene graphs. Complementary efforts construct hierarchical 3D scene graphs

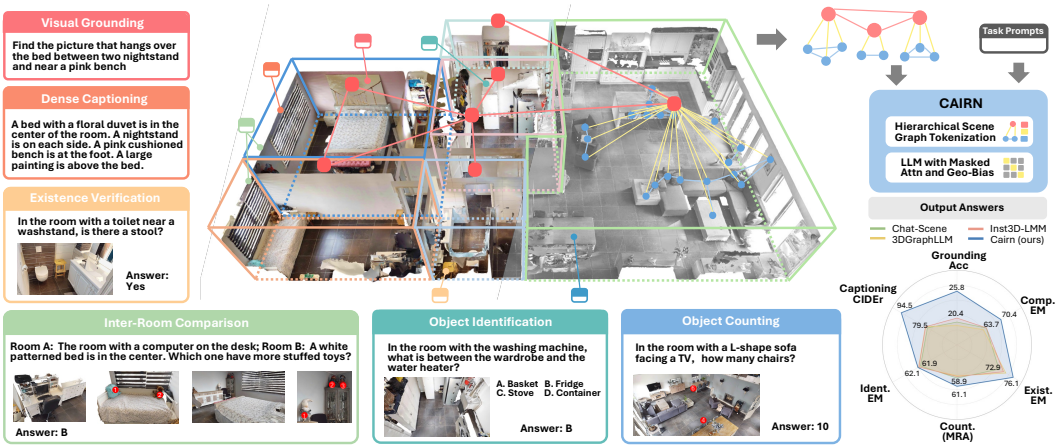


Figure 1: **3D-LLMs for multi-room scene understanding.** We introduce **CAIRN**, a topology-aware 3D-LLM for multi-room scene understanding, along with **CAIRN-MR**, a multi-room 3D scene understanding benchmark with diverse tasks. By representing each scene as a hierarchical scene graph and aligning attention with scene topology through structured masking and geometric bias, CAIRN achieves substantial gains over prior 3D-LLMs.

from raw data [15, 23, 50], but do not study language-grounded reasoning over such representations. Our work bridges these two directions by building hierarchical representations from raw 3D data and enabling language-grounded multi-room scene understanding over them.

Attention with Structural Priors. Standard self-attention models dense pairwise interactions without structural awareness. Many works revise this by injecting domain-specific priors. Sparse attention methods [2, 12, 59] restrict attention to local windows or global tokens for efficiency in long sequences. Others inject structural priors as additive biases, such as ALiBi [39], which penalizes attention by linear distance, and Graphormer-style models [6, 30, 56], which encode shortest-path distances and node centrality. We instead design topology-aware attention for multi-room scenes, combining masks that constrain information flow with geometric biases that encode spatial relations.

3 CAIRN-MR: Multi-Room 3D Scene Understanding Benchmark

Most existing 3D scene understanding benchmarks confine reasoning to single-room scenes, yet real household environments contain multiple interconnected rooms with distinct spatial structure. To advance 3D-LLMs toward structured multi-room scene understanding, we introduce **CAIRN-MR**, the first benchmark covering six cross-room reasoning tasks (Fig. 2). These tasks capture challenges overlooked in single-room settings: objects of the same category may appear across different rooms, requiring disambiguation over spatially separated regions; reasoning must span both object-level details and room-level structure; and scene information grows substantially with the number of rooms, demanding more scalable understanding methods.

3.1 Multi-Room Reasoning and Task Design

CAIRN-MR presents three progressively more challenging forms of multi-room reasoning. It remains compatible with standard grounding and captioning tasks, while introducing question answering tasks that directly test room hierarchy, implicit room localization, and cross-room ambiguity.

1) Object Visual Grounding. Given a referring expression, the model localizes the target object among all objects in the scene. Unlike prior settings where the search is confined to a single region, disambiguation here must span the entire scene, and recurring object categories with similar layouts across rooms make multiple candidates plausible matches.

2) Dense Captioning. Given a target object, the model generates a referring expression that uniquely identifies it. Unlike single-room settings, the expression must distinguish the target from all same-category instances across the entire scene, not just those within one room. This requires the model to reason about which attributes or spatial relations remain discriminative at scene scale.



Figure 2: **CAIRN-MR benchmark for multi-room 3D scene understanding.** The benchmark includes grounding, captioning, and question answering tasks covering room localization, intra-room reasoning, and cross-room comparison. Colored boxes denote parsed room regions. Right: benchmark scale and task composition.

3) Object Identification. Given a spatial description that implicitly specifies a room, together with an additional cue about a target object inside it, the model predicts the target category from four candidates. This requires two-stage reasoning: the model must first localize the room from the spatial description, and then identify the target object within that room. Errors in room localization therefore directly affect the final prediction.

4) Counting Question Answering. The model counts all instances of a queried category within the room implicitly specified by the spatial description. The main challenge is restricting the count to the correct room: same-category instances often appear in multiple rooms, so accurate counting depends on correct room localization.

5) Existence Verification Question Answering. The model determines whether a queried category is present in the room specified by the spatial description. Negative cases are particularly challenging, because the queried category may still appear elsewhere in the scene. To answer correctly, the model must reason over the localized room rather than relying on scene-level presence.

6) Inter-room Comparison Question Answering. Given two spatial descriptions that specify two different rooms, the model determines which room contains more instances of a queried category. This task introduces explicit cross-room reasoning: both rooms must be localized and counted separately, and an error in either room is sufficient to invalidate the comparison.

3.2 Benchmark Construction and Statistics

Tasks in CAIRN-MR require the model to infer the relevant room from object-level spatial descriptions rather than predefined room labels. This places a strong requirement on the referring expressions: they must be spatially discriminative enough to uniquely identify objects across the full scene and structurally rich enough to implicitly specify which room is being referenced. To achieve this, we construct the benchmark in three stages: scene assembly, expression generation, and task instantiation. We build on HM3D [44] residential environments and the object-level relational annotations from SceneVerse [24]. In the first stage, we assemble connected multi-room layouts from proximate rooms in each HM3D building, ensuring mutual reachability. In the second stage, we construct referring expressions from SceneVerse spatial relations in three forms–multi-reference, multi-hop chains, and branched chains–and retain those that are sufficiently informative to uniquely identify a single object across all rooms. A subset of expressions is further augmented with VLM-derived appearance modifiers (e.g., chair $\rightarrow$ wooden chair) to increase linguistic diversity. In the third stage, the validated expressions are used to instantiate all six tasks, including grounding, captioning, and four QA types via LLM rewriting and rule-based templates. Importantly, these rules are instantiated over diverse real-world scanned environments rather than fixed scene configurations. The resulting variations in object arrangements, recurring categories, and room layouts introduce substantial cross-room ambiguity and reasoning challenges. Full details are provided in Appendix B.

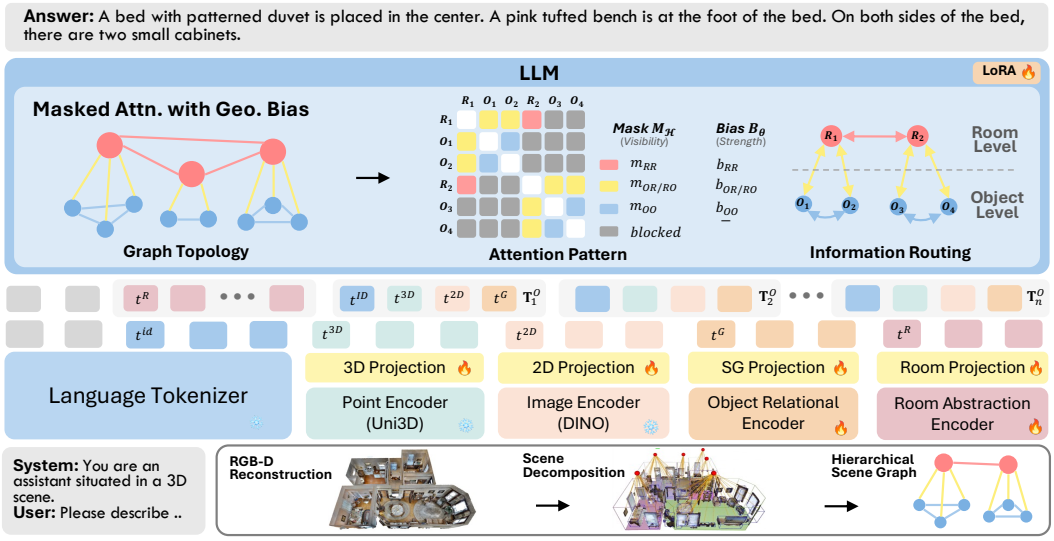


Figure 3: **Overview of CAIRN.** Given a 3D scene, CAIRN constructs a hierarchical scene graph capturing object relations and room adjacencies (bottom), tokenizes it into object and room tokens (middle), and feeds them to an LLM with hierarchical masked attention and geometric bias (top). The mask routes information along scene topology, while learned bias terms inject spatial priors into attention logits, enabling topology-aware reasoning across rooms.

The resulting benchmark contains 673 scenes, with 479 for training and 194 for validation. Splits are partitioned at the building level with no scene overlap. On average, each scene contains 4.5 rooms and approximately 115 object instances. As summarized in Fig. 2, CAIRN-MR provides 238K task annotations spanning grounding, captioning, and question answering.

4 CAIRN: Hierarchical Topology-Aware 3D-LLM

Our goal is to enable 3D-LLMs to perform multi-room 3D scene understanding in full household environments. Our approach, **CAIRN**, explicitly models the hierarchical topology of multi-room 3D scenes, enabling 3D-LLMs to reason over both object-level and room-level structure, as illustrated in Fig. 3. We first describe how we represent a multi-room 3D scene as a hierarchical 3D scene graph, capturing both object relations and room adjacencies (Sec. 4.1). We then describe how this graph is encoded into scene tokens while preserving both object-level and room-level information (Sec. 4.2). Finally, we detail how CAIRN incorporates structured masked attention and geometric bias to make 3D-LLMs topology-aware by constraining information flow according to scene connectivity and injecting explicit spatial priors into attention (Sec. 4.3).

4.1 Multimodal Hierarchical Scene Graph Construction

Given a 3D point cloud $\mathcal{P}$ representing an entire multi-room scene, CAIRN constructs a two-layer hierarchical scene graph $\mathcal{G}_{\mathcal{H}}$. The lower layer encodes object-level structure $(\mathcal{V}_O, \mathcal{E}_{OO})$, while the upper layer captures room-level topology $(\mathcal{V}_R, \mathcal{E}_{RR})$. Cross-layer edges $\mathcal{E}_{OR}$ assign each object to its containing room, forming a graph as:

$$\mathcal{G}_{\mathcal{H}} = \big((\mathcal{V}_O, \mathcal{E}_{OO}), \mathcal{E}_{OR}, (\mathcal{V}_R, \mathcal{E}_{RR})\big). \tag{1}$$

In practice, we construct this graph using off-the-shelf instance segmentation and room partitioning methods, without relying on ground-truth scene annotations. Following Chat-Scene [21], we use Mask3D [46] to obtain 3D object segments without assigning semantic category labels. Each segment defines an object node $\mathcal{O}_i \in \mathcal{V}_O$, with geometric attributes such as its 3D center and bounding-box dimensions derived from the segmented points. Edges $\mathcal{E}_{OO}$ connect each object to its top-K_g nearest neighbors within the same room, with geometric edge attributes defined by the relative 3D displacement between object centers.

For the room-level graph, we follow HOV-SG [50] to obtain room partitions through bird's-eye-view occupancy analysis, without assigning semantic room labels. Each partition defines a room node

$\mathcal{R}_r \in \mathcal{V}_R$, where $\mathcal{V}_R = \{\mathcal{R}_r\}_{r=1}^{M}$, and spatially adjacent rooms are connected by edges $\mathcal{E}_{RR}$. Object-room edges $\mathcal{E}_{OR}$ are established by assigning each object to its estimated containing room. The resulting hierarchical graph captures local object arrangements and global room topology, providing the structural foundation for subsequent multimodal tokenization and LLM reasoning.

4.2 Hierarchical Scene Graph Tokenization

To encode the constructed hierarchical scene graph into the LLM token space, we transform object nodes into multimodal object tokens and construct room tokens by integrating room geometry with information aggregated from objects within each room. Together, these tokens capture both fine-grained object details and higher-level room context.

Object-Level Tokenization. For each object node $\mathcal{O}_i \in \mathcal{V}_O$, we assign a unique textual identifier (e.g., `<OBJ_01>`), embedded into the LLM space as t_i^{id}. We then extract 3D and 2D object features using pretrained Uni3D [65] and DINO [38], respectively, and project them into the LLM space to obtain t_i^{3D} and t_i^{2D}.

To capture relational context beyond individual object features, we augment the geometric edge attributes with pretrained semantic relation features from VL-SAT [49]. We then apply a message-passing neural network (MPNN) with L_g layers over $\mathcal{E}_{OO}$, where each object updates its representation by aggregating information from connected neighbors. The resulting representation is projected into the LLM space to obtain the graph token t_i^G. The MPNN is trained end-to-end with the full model using next-token prediction loss, without separate pretraining or relation-level supervision. See Appendix C.2 for more details.

Together, these representations form a fixed four-token block for each object node $\mathcal{O}_i$:

$$\mathbf{T}_i^O = [t_i^{\text{id}},\, t_i^{3D},\, t_i^{2D},\, t_i^G], \qquad t_i^{\text{id}}, t_i^{3D}, t_i^{2D}, t_i^G \in \mathbb{R}^d, \tag{2}$$

where d denotes the LLM hidden dimension.

Room-Level Tokenization. To capture room-level context beyond geometric partitions, we construct room tokens by combining room geometry with aggregated information from objects within each room. Specifically, for each room $\mathcal{R}_r$, we employ $K_r = 4$ learned queries $\{q_{r,k}\}_{k=1}^{K_r}$ that cross-attend to its object representations:

$$t_{r,k}^R = W_R \left[\text{CrossAttn}\left(q_{r,k}, \{\mathbf{T}_i^O\}_{i\in\mathcal{O}^{(r)}}\right); \mathbf{g}_r\right], \tag{3}$$

where $\mathcal{O}^{(r)}$ denotes the set of objects assigned to room $\mathcal{R}_r$, and $\mathbf{g}_r$ denotes its geometric descriptor. The resulting room tokens $\mathbf{T}_r^R = [t_{r,1}^R, \ldots, t_{r,K_r}^R]$ summarize room-level content and subsequently support cross-room information exchange through topology-aware attention, as detailed in the next section.

4.3 Hierarchical Masked Attention with Geometric Bias

While hierarchical tokenization provides object- and room-level representations of the scene graph $\mathcal{G}_\mathcal{H}$ (Sec. 4.1), these representations alone do not ensure that the LLM respects the underlying scene topology. In particular, standard self-attention treats all scene tokens as fully connected, regardless of their hierarchical relationships. To address this, we introduce topology-aware attention that explicitly incorporates scene structure into information flow. Specifically, a structured mask constrains token interactions according to graph connectivity, while a geometric bias modulates the permitted interactions based on spatial relationships.

Let $\mathbf{T} = [\mathbf{T}_{\text{lang}}; \mathbf{T}_{\text{scene}}] \in \mathbb{R}^{L\times d}$ denote the LLM input, where $\mathbf{T}_{\text{lang}}$ contains prompt tokens and $\mathbf{T}_{\text{scene}}$ contains object and room tokens. The CAIRN attention operator is defined as

$$\text{Attn}(X) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d}} + M_\mathcal{H} + B_\theta\right) V. \tag{4}$$

The two additive terms play complementary roles: $M_\mathcal{H}$ enforces which token pairs may interact, while B_θ modulates their interaction strength based on geometry. The mask is applied only among object and room tokens (prompt tokens retain standard attention), with allowed pairs defined as

$$\mathcal{S}_\mathcal{H} = \mathcal{S}_{\text{OO}} \cup \mathcal{S}_{\text{OR}} \cup \mathcal{S}_{\text{RR}}. \tag{5}$$

Each subset corresponds to edges in $\mathcal{G}_{\mathcal{H}}$ (Eq. (1)) and defines bidirectional token interactions. This induces a hierarchical routing pattern: object tokens interact only within their room, and cross-room communication is mediated through room tokens, which carry inter-room information flow.

To modulate the strength of admitted interactions, the geometric bias adds a learned offset to the attention logits:

$$B_\theta(u, v) = b_{\tau(u,v)}(\mathbf{g}_{uv}), \tag{6}$$

where $\tau(u, v) \in \{\mathrm{OO}, \mathrm{OR}, \mathrm{RR}\}$ denotes the interaction type and b_τ is a type-specific MLP. The descriptor $\mathbf{g}_{uv}$ captures the spatial relation at each level: relative position and distance for OO pairs, object-to-room location for OR pairs, and relative layout for RR pairs.

B_θ is applied to the first L_b layers, while later layers retain only the structural sparsity induced by $M_{\mathcal{H}}$. This induces a block-sparse interaction pattern (Fig. 3), reducing the number of permitted object-object interactions from $O(N^2)$ to $O\big(\sum_r |\mathcal{O}^{(r)}|^2\big)$. Together, $M_{\mathcal{H}}$ and B_θ turn attention into a structural operator over the scene hierarchy, where topology determines token visibility and geometry determines interaction strength.

5 Experiments

We evaluate CAIRN through five questions: 1) How is the quality of our constructed CAIRN-MR benchmark? 2) Does the hierarchical design of CAIRN 3D-LLM improve performance on cross-room reasoning tasks? 3) Does CAIRN remain effective on single-room scene understanding? 4) Which components of CAIRN are responsible for the observed gains, and for which task types? 5) How robust is CAIRN to errors in the predicted hierarchical scene graph?

5.1 Implementation Details

We use Qwen3-8B [47] as the LLM backbone, frozen and adapted via LoRA [19] (rank 16, all attention and MLP projections); input embedding and output head are unfrozen for object identifier tokens. The graph encoder uses L_g=2 message-passing layers, each room is summarized by K_r=4 learned queries, and geometric bias B_θ is applied to the first L_b=4 transformer layers. Training has two stages on 8×H200 GPUs (∼12 hours total): Stage 1 trains for 3 epochs on ScanNet tasks [1, 5, 11, 37, 61]; Stage 2 trains for 3 epochs on 30% ScanNet and 100% CAIRN-MR. Full hyperparameters and training details are provided in Appendix C.1.

5.2 Experiment Setup

Datasets. For single-room evaluation, we use five ScanNet-based benchmarks: ScanRefer [5] and Multi3DRefer [61] (grounding), Scan2Cap [11] (captioning), ScanQA [1] (VQA), and SQA3D [37] (situated QA). For multi-room evaluation, we use CAIRN-MR (Sec. 3.1). Splits are partitioned at the building level with no scene overlap.

Metrics. We follow standard protocols for ScanNet benchmarks: Acc@0.25/0.5 for ScanRefer, F1@0.25/0.5 for Multi3DRefer, CIDEr@0.5 and BLEU-4@0.5 for Scan2Cap, CIDEr and BLEU-4 for ScanQA, and exact match for SQA3D. For CAIRN-MR, grounding and captioning follow ScanRefer and Scan2Cap protocols. QA tasks use rule-based answer extraction, with malformed outputs scored as incorrect: Identification and Comparison are evaluated by option selection, Existence by binary accuracy, and Counting by Mean Relative Accuracy (MRA) [54].

5.3 Benchmark Quality

We evaluate the quality of CAIRN-MR's QA tasks along three dimensions: annotation reliability, task difficulty, and answer distribution, using three baselines (Tab. 1): Random (selects answers uniformly), Frequency (predicts the most frequent answer), and Human (3 annotators on a subset). We focus on QA tasks as they provide discrete, verifiable outputs for controlled calibration.

Annotation Quality. Human performance is consistently high across tasks (86.8%–93.5%), indicating that the annotations are reliable and that the tasks are well-defined. This suggests that performance gaps primarily reflect task difficulty rather than annotation noise.

Task Difficulty. Human performance substantially exceeds random and frequency baselines across all tasks, indicating that the tasks are non-trivial and require genuine spatial reasoning beyond surface statistics.

Table 1: **Calibration of QA tasks in CAIRN-MR.** Performance of random, frequency, and human baselines across tasks.

	Ident.	Count.		Exist.	Comp.
	EM↑	EM↑	MRA↑	EM↑	EM↑
Random	25.0	11.1	19.9	50.0	50.0
Frequency	25.3	54.7	62.6	50.0	48.9
Human	86.8	82.3	94.5	93.5	91.4

Answer Distribution Balance. Random and frequency baselines are closely aligned on Identification, Existence, and Comparison, suggesting balanced answer distributions without dominant labels. Counting differs due to skew toward small integers (e.g., 2 accounts for 49.8%), allowing strong frequency performance (62.6% MRA) from numerical priors rather than reasoning. This skew is intrinsic to indoor scenes [1] and serves as a reference for interpreting the metric.

5.4 Performance on Multi-Room Scene Understanding

Tab. 2 compares CAIRN with three representative baselines and a room-aware variant of the strongest baseline on CAIRN-MR, all trained under the same two-stage protocol with Qwen3-8B. Compared with the original baselines, CAIRN achieves the largest gains on grounding and captioning: +5.4 Acc@0.25/+5.4 Acc@0.5 and +14.9 CIDEr (roughly 19% relative). These improvements reflect more accurate cross-room disambiguation enabled by topology-aware modeling of room-level structure. We further equip Inst3D-LMM with the same room tokens while retaining dense attention. Despite improvements over the original baseline, it still trails CAIRN by 4.0 Acc@0.25 and 12.8 CIDEr, highlighting the importance of topology-aware information routing beyond room representations alone.

Across QA tasks, gains scale with reliance on room-level structure. Tasks requiring cross-room reasoning (Comparison, Existence) show the largest improvements (+6.7, +3.4 EM), as they depend on reasoning across room boundaries. Counting, which enumerates objects within a localized room, shows moderate gains (+2.2 MRA), benefiting from clearer room-level scope. Identification, which only requires recognizing an object within an already localized room, remains comparable across methods (62.1 vs. 61.9 EM). This trend, from cross-room reasoning to within-room enumeration to object recognition, shows that CAIRN's gains scale with the level of room-level abstraction required. The improvements are consistent across backbones with strong performance even for smaller models, showing that CAIRN remains effective under reduced model capacity. (Tab. 2).

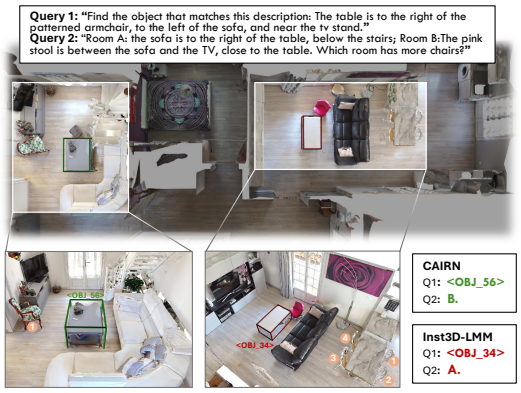


Figure 4: **Qualitative results on a multi-room scene** with visually similar living rooms. Green denotes CAIRN; red denotes the best baseline.

5.5 Performance on Single-Room Setting

Tab. 3 reports results on five ScanNet-based single-room benchmarks. Since these scenes contain a single room, the hierarchical structure reduces to a single level, leaving graph tokens and geometric bias as the effective components. CAIRN matches or exceeds prior methods across all benchmarks, achieving the best or second-best results on 7 of 9 metrics. These results indicate that CAIRN's room-level components do not interfere with single-room performance. The gains are primarily driven by graph tokens and geometric bias, which encode local relational structure independent of room hierarchy, yielding improved grounding performance while maintaining competitive results across tasks.

Table 2: **CAIRN-MR multi-room performance comparison.** Results on grounding, captioning, and question answering tasks. Best results are in **bold**.

Method	LLM	Grounding		Captioning		Question Answering			
		A@ 0.25↑	A@ 0.5↑	C↑	B-4↑	Ident. EM↑	Count. MRA↑	Exist. EM↑	Comp. EM↑
Chat-Scene [21]	Qwen3-8B	19.6	19.4	78.3	21.2	61.3	58.9	69.5	63.6
3DGraphLLM [60]	Qwen3-8B	18.9	18.7	79.6	21.1	61.9	58.3	72.7	62.3
Inst3D-LMM [58]	Qwen3-8B	20.4	20.0	79.5	21.3	61.8	58.5	71.6	63.7
Inst3D + Room Tokens	Qwen3-8B	21.8	21.3	81.7	21.1	61.5	59.4	73.4	65.1
CAIRN (ours)	Llama-3.2-3B	24.2	23.8	91.0	22.2	60.9	59.8	75.5	70.0
	Llama-3.1-8B	24.6	24.1	89.5	21.9	**62.5**	59.4	75.9	69.8
	Qwen3-4B	24.3	23.9	89.7	22.0	61.9	60.8	75.7	68.2
	Qwen3-8B	**25.8**	**25.4**	**94.5**	**22.8**	62.1	**61.1**	**76.1**	**70.4**

Table 3: **Single-room performance comparison.** Results on grounding, captioning, and question answering tasks. Best in **bold**; second best underlined.

	Method	LLM	ScanRefer		Multi3DRefer		Scan2Cap		ScanQA		SQA3D
			A@0.25↑	A@0.5↑	F1@0.25↑	F1@0.5↑	C@0.5↑	B-4@0.5↑	C↑	B-4↑	EM↑
Expert Models	3DVG-Trans [62]	×	45.9	34.5	-	-	-	-	-	-	-
	ViL3DRel [7]	×	47.9	37.7	-	-	-	-	-	-	-
	M3DRef-CLIP [61]	×	51.9	44.7	42.8	38.4	-	-	-	-	-
	3D-VLP [25]	×	51.4	39.5	-	-	54.9	32.3	67.0	11.1	-
	3D-VisTA [66]	×	50.6	45.8	-	-	66.9	34.0	72.9	13.1	48.5
	Vote2Cap-DETR++ [9]	×	-	-	-	-	67.6	37.1	-	-	-
	PQ3D [67]	×	-	51.2	50.1	-	80.3	36.0	87.8	-	47.1
LLM-based Models	3D-LLM [18]	Flamingo	21.2	-	-	-	-	-	59.2	7.2	-
	Scene-LLM [14]	Llama-2-7B	-	-	-	-	-	-	80.0	12.0	54.2
	LL3DA [8]	OPT-1.3B	-	-	-	-	65.2	36.8	76.8	13.5	-
	LEO [22]	Vicuna-7B-v1.1	-	-	-	-	72.4	38.2	**101.4**	13.2	50.0
	Grounded 3D-LLM [10]	Tiny-Vicuna-1B	47.9	44.1	45.2	40.6	70.6	35.5	72.7	13.4	-
	GPT4Scene-HD [43]	Qwen2-VL-7B	50.9	46.4	53.7	50.0	74.4	37.9	89.9	**15.9**	57.2
	Chat-Scene [21]	Vicuna-7B-v1.5	55.5	50.2	57.1	52.4	77.1	36.3	87.7	14.3	54.6
		Qwen3-8B	57.5	51.9	61.3	56.7	75.4	34.5	90.3	13.1	55.8
	3DGraphLLM [60]	Qwen3-8B	58.4	53.1	62.9	58.1	78.1	35.2	91.2	13.1	56.1
	Inst3D-LMM [58]	Vicuna-7B-v1.5	57.8	51.6	58.3	53.5	79.7	**38.3**	88.6	14.9	-
		Qwen3-8B	58.9	52.7	62.4	57.9	**80.9**	38.0	91.5	13.6	-
	CAIRN (ours)	Qwen3-8B	**59.6**	**54.1**	**63.8**	**58.8**	80.7	36.2	92.1	13.5	**58.1**

5.6 Ablation Studies

Tab. 4 presents an ablation on CAIRN-MR to analyze component contributions. Starting from the full model, we remove topology-aware attention (hierarchical mask and geometric bias) and structured representations (room and graph tokens), resulting in a flat baseline.

Effectiveness of Topology-Aware Attention. Topology-aware attention is the primary driver of performance gains. Removing the hierarchical mask causes the largest drops across all metrics (e.g., -5.5 EM on comparison, -6.7 CIDEr on captioning, -2.9 Acc@0.25 on grounding), as it enforces topology-aware routing of cross-room information. Without it, cross-room information flows become unconstrained, diluting room-level distinctions even when room tokens are present. The geometric bias provides consistent but smaller gains (e.g., $+1.3$ Acc@0.25, $+1.6$ EM) by injecting structure into attention.

Effectiveness of Structured Representations. Structured representations provide complementary gains by encoding global and local context. Room tokens aggregate room-level information into compact summaries, improving captioning ($+3.4$ CIDEr) and comparison ($+2.1$ EM), while graph tokens encode local object relations, contributing gains across captioning ($+4.0$ CIDEr), existence ($+3.5$ EM), and grounding ($+1.2$ Acc@0.25), particularly in tasks where local context supports spatial reasoning.

Gains Increase with Room-Level Reasoning. Performance gains increase with the degree of room-level reasoning required. Identification remains largely unaffected (within 1 point), as it primarily

Table 4: **Ablation of CAIRN.** We start from the full model and progressively remove components: geometric attention bias, hierarchical attention mask, room-level tokens, and graph tokens.

Variant	GNN	Room	Mask	Bias	Grounding		Captioning		Question Answering			
					A@ 0.25↑	A@ 0.5↑	C↑	B-4↑	Ident. EM↑	Count. MRA↑	Exist. EM↑	Comp. EM↑
CAIRN (full)	✓	✓	✓	✓	**25.8**	**25.4**	**94.5**	**22.8**	**62.1**	**61.1**	**76.1**	**70.4**
– geometric bias	✓	✓	✓	–	24.5	24.1	92.4	22.4	62.1	60.6	74.5	69.8
– hierarchical mask	✓	✓	–	–	21.6	21.1	85.7	21.8	61.8	59.3	72.6	64.3
– room tokens	✓	–	–	–	20.8	20.6	82.3	21.5	62.3	57.9	73.0	62.2
– graph tokens (= flat)	–	–	–	–	19.6	19.4	78.3	21.2	61.3	58.6	69.5	63.6

relies on recognizing objects within a localized room. Counting shows moderate degradation ($\sim$3 points), reflecting its dependence on aggregating instances within a correctly identified room. In contrast, Existence and Comparison exhibit the largest drops ($\sim$7–8 points), as they require reasoning across rooms and are more sensitive to disruptions in cross-room information flow. This pattern mirrors the graded trend observed in Tab. 2.

5.7 Robustness to Scene Graph Corruption

We stress-test CAIRN's robustness by injecting increasingly severe noise into the predicted hierarchical scene graphs. For detection-level errors, we drop 10/20/30% of objects, either retaining answer-relevant instances (context dropout) or sampling uniformly across all objects (unrestricted dropout). For room-structure errors, we randomly reassign 10/20/30% of objects to incorrect rooms, shift room boundaries to misassign 25/50/100% of nearby objects, add or remove 1/2/3 room connections, and merge 1/2 pairs of rooms. Tab. 5 reports the highest tested severity for each corruption.

The results reveal a clear hierarchy of robustness. CAIRN is nearly unaffected by adjacency errors and degrades only moderately under boundary shifts. Random object-room reassignment primarily affects grounding, while room merging causes larger drops in counting and existence, whose answers depend on correct room membership. Unrestricted dropout substantially reduces grounding accuracy because target objects may be removed entirely, whereas context dropout causes only moderate degradation.

Table 5: **Robustness to scene graph corruption.**

Corruption	Ground. A@0.25↑	Ident. EM↑	Count. MRA↑	Exist. EM↑	Comp. EM↑
Clean graph	26.3	63.0	62.4	76.2	70.3
Context dropout	25.2	60.8	63.1	70.3	66.5
Object dropout	18.0	60.8	63.2	71.6	66.7
Obj. reassignment	21.6	60.8	62.3	71.6	66.7
Boundary shift	24.8	62.8	61.8	71.2	68.9
Adjacency flips	26.1	63.2	61.6	76.1	69.1
Room merging	24.7	62.5	57.2	66.5	66.5

6 Conclusion

We introduce CAIRN, a topology-aware 3D-LLM for multi-room scene understanding, together with CAIRN-MR, a benchmark requiring both object- and room-level reasoning. By modeling hierarchical scene structure, CAIRN aligns information flow with scene topology via masked attention and encodes spatial priors through geometric bias. Experiments show that CAIRN consistently outperforms prior methods across grounding, captioning, and question answering on CAIRN-MR, while generalizing effectively to single-room benchmarks. This suggests that topology-aware modeling enables structured 3D reasoning across complex multi-room environments.

Limitations and Future Work. Further scaling model capacity and training data may yield additional gains. Extending CAIRN-MR to more complex, multi-level environments would broaden the scope of structured reasoning. Another direction is to incorporate explicit reasoning processes for more interpretable multi-step reasoning over scene structure, potentially improving compositional generalization across more complex scene configurations.

Acknowledgments and Disclosure of Funding

We thank our colleagues at the University of Oxford and Microsoft Research for their constructive feedback, insightful discussions, and continued support. We also thank the anonymous reviewers and area chairs for their valuable comments, which helped improve the quality and clarity of this paper. Apart from the support disclosed above and that provided by the authors' affiliated institutions, no additional funding was received for this work. The authors declare no additional competing financial interests.

References

[1] 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.

[2] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[3] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL `https://arxiv.org/abs/2511.16719`.

[4] Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Mottaghi, Priyam Parashar, et al. Partnr: A benchmark for planning and reasoning in embodied multi-agent tasks. *arXiv preprint arXiv:2411.00081*, 2024.

[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] Dexiong Chen, Leslie O'Bray, and Karsten Borgwardt. Structure-aware transformer for graph representation learning. In *International conference on machine learning*, pages 3469–3489. PMLR, 2022.

[7] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Language conditioned spatial relation reasoning for 3d object grounding. *Advances in Neural Information Processing Systems*, 35:20522–20535, 2022.

[8] 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. *arXiv preprint arXiv:2311.18651*, 2023.

[9] Sijin Chen, Hongyuan Zhu, Mingsheng Li, Xin Chen, Peng Guo, Yinjie Lei, YU Gang, Taihao Li, and Tao Chen. Vote2cap-detr++: Decoupling localization and describing for end-to-end 3d dense captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[10] Yilun Chen, Shuai Yang, Haifeng Huang, Tai Wang, Ruiyuan Lyu, Runsen Xu, Dahua Lin, and Jiangmiao Pang. Grounded 3d-llm with referent tokens. *arXiv preprint arXiv:2405.10370*, 2024.

[11] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3193–3203, 2021.

[12] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[13] 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.

[14] 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.

[15] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5021–5028. IEEE, 2024.

[16] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[17] Daniel Honerkamp, Martin Büchner, Fabien Despinoy, Tim Welschehold, and Abhinav Valada. Language-grounded dynamic scene graphs for interactive object search with mobile manipulation. *IEEE Robotics and Automation Letters*, 9(10):8298–8305, 2024.

[18] 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. *arXiv preprint arXiv:2307.12981*, 2023.

[19] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[20] Haifeng Huang, Zehan Wang, Rongjie Huang, Luping Liu, Xize Cheng, Yang Zhao, Tao Jin, and Zhou Zhao. Chat-3d v2: Bridging 3d scene and large language models with object identifiers. *arXiv preprint arXiv:2312.08168*, 2023.

[21] 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.

[22] 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.

[23] N. Hughes, Y. Chang, and L. Carlone. Hydra: A Real-time Spatial Perception System for 3D Scene Graph Construction and Optimization. In *Robotics: Science and Systems (RSS)*, 2022.

[24] Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. In *European Conference on Computer Vision*, pages 289–310. Springer, 2024.

[25] 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.

[26] Weitai Kang, Haifeng Huang, Yuzhang Shang, Mubarak Shah, and Yan Yan. Robin3d: Improving 3d large language model via robust instruction tuning. *arXiv preprint arXiv:2410.00255*, 2024.

[27] Mukul Khanna*, Yongsen Mao*, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X. Chang, and Manolis Savva. Habitat Synthetic Scenes Dataset (HSSD-200): An Analysis of 3D Scene Scale and Realism Tradeoffs for ObjectGoal Navigation. *arXiv preprint*, 2023.

[28] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023.

[29] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *Science China Information Sciences*, 68(10): 200102, 2025.

[30] Chuang Liu, Zelin Yao, Yibing Zhan, Xueqi Ma, Shirui Pan, and Wenbin Hu. Gradformer: Graph transformer with exponential decay. *arXiv preprint arXiv:2404.15729*, 2024.

[31] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.

[32] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

[33] K. Lu, C. Ma, C. Hori, and D. Romeres. KitchenVLA: Iterative vision-language corrections for robotic execution of human tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation Workshop on Safely Leveraging Vision-Language Foundation Models in Robotics (SafeLVMs@ICRA)*, 2025.

[34] Chenyang Ma, Kai Lu, Ta-Ying Cheng, Niki Trigoni, and Andrew Markham. Spatialpin: Enhancing spatial reasoning capabilities of vision-language models through prompting and interacting 3d priors. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[35] Chenyang Ma, Kai Lu, Ruta Desai, Xavier Puig, Andrew Markham, and Niki Trigoni. Coopera: Continual open-ended human-robot assistance. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[36] Chenyang Ma, Guangyu Yang, Kai Lu, Shitong Xu, Bill Byrne, Niki Trigoni, and Andrew Markham. Cyclevla: Proactive self-correcting vision-language-action models via subtask backtracking and minimum bayes risk decoding. *arXiv preprint arXiv:2601.02295*, 2026.

[37] 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.

[38] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[39] Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.

[40] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8494–8502, 2018.

[41] Xavier Puig, Tianmin Shu, Shuang Li, Zilin Wang, Yuan-Hong Liao, Joshua B Tenenbaum, Sanja Fidler, and Antonio Torralba. Watch-and-help: A challenge for social perception and human-ai collaboration. *arXiv preprint arXiv:2010.09890*, 2020.

[42] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023.

[43] Zhangyang Qi, Zhixiong Zhang, Ye Fang, Jiaqi Wang, and Hengshuang Zhao. Gpt4scene: Understand 3d scenes from videos with vision-language models. *arXiv preprint arXiv:2501.01428*, 2025.

[44] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021.

[45] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. *arXiv preprint arXiv:2307.06135*, 2023.

[46] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8216–8223. IEEE, 2023.

[47] Qwen Team. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

[48] Hugues Thomas, Chen Chen, and Jian Zhang. Pts3d-llm: Studying the impact of token structure for 3d scene understanding with large language models. *arXiv preprint arXiv:2506.05689*, 2025.

[49] Ziqin Wang, Bowen Cheng, Lichen Zhao, Dong Xu, Yang Tang, and Lu Sheng. Vl-sat: Visual-linguistic semantics assisted training for 3d semantic scene graph prediction in point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21560–21569, 2023.

[50] Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. In *Proceedings of Robotics: Science and Systems*, 2024.

[51] Pengying Wu, Yao Mu, Kangjie Zhou, Ji Ma, Junting Chen, and Chang Liu. Camon: Cooperative agents for multi-object navigation with llm-based conversations. *arXiv preprint arXiv:2407.00632*, 2024.

[52] Haomiao Xiong, Yunzhi Zhuge, Jiawen Zhu, Lu Zhang, and Huchuan Lu. 3ur-llm: An end-to-end multimodal large language model for 3d scene understanding. *IEEE Transactions on Multimedia*, 27: 2899–2911, 2025. doi: 10.1109/TMM.2025.3557620.

[53] Jintang Xue, Ganning Zhao, Jie-En Yao, Hong-En Chen, Yue Hu, Meida Chen, Suya You, and C-C Jay Kuo. Descrip3d: Enhancing large language model-based 3d scene understanding with object-level text descriptions. *arXiv preprint arXiv:2507.14555*, 2025.

[54] Jihan Yang, Shusheng Yang, Anjali Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in Space: How Multimodal Large Language Models See, Remember and Recall Spaces. In *CVPR*, 2025.

[55] Sriram Yenamandra, Arun Ramachandran, Karmesh Yadav, Austin Wang, Mukul Khanna, Theophile Gervet, Tsung-Yen Yang, Vidhi Jain, Alexander William Clegg, John Turner, et al. Homerobot: Open-vocabulary mobile manipulation. *arXiv preprint arXiv:2306.11565*, 2023.

[56] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.

[57] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704*, 2023.

[58] Hanxun Yu, Wentong Li, Song Wang, Junbo Chen, and Jianke Zhu. Inst3d-lmm: Instance-aware 3d scene understanding with multi-modal instruction tuning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14147–14157, 2025.

[59] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

[60] Tatiana Zemskova and Dmitry Yudin. 3dgraphllm: Combining semantic graphs and large language models for 3d scene understanding. *arXiv preprint arXiv:2412.18450*, 2024.

[61] Yiming Zhang, ZeMing Gong, and Angel X Chang. Multi3drefer: Grounding text description to multiple 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15225–15236, 2023.

[62] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3dvg-transformer: Relation modeling for visual grounding on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2928–2937, 2021.

[63] Yang Zhao, Zhijie Lin, Daquan Zhou, Zilong Huang, Jiashi Feng, and Bingyi Kang. Bubogpt: Enabling visual grounding in multi-modal llms. *arXiv preprint arXiv:2307.08581*, 2023.

[64] Hongyan Zhi, Peihao Chen, Junyan Li, Shuailei Ma, Xinyu Sun, Tianhang Xiang, Yinjie Lei, Mingkui Tan, and Chuang Gan. Lscenellm: Enhancing large 3d scene understanding using adaptive visual preferences. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3761–3771, 2025.

[65] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. *arXiv preprint arXiv:2310.06773*, 2023.

[66] 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.

[67] Ziyu Zhu, Zhuofan Zhang, Xiaojian Ma, Xuesong Niu, Yixin Chen, Baoxiong Jia, Zhidong Deng, Siyuan Huang, and Qing Li. Unifying 3d vision-language understanding via promptable queries. In *European Conference on Computer Vision*, pages 188–206. Springer, 2024.

Appendix for CAIRN

A Overview

This appendix provides supplementary materials supporting the main paper. Sec. B details the construction of the CAIRN-MR benchmark, including multi-room scene assembly from HM3D [44], the procedure for generating and validating spatial referring expressions, appearance-based noun augmentation via vision-language models, and the instantiation of grounding, captioning, and four QA task formats. Sec. C provides additional implementation details, including model hyperparameters and architectural details for object-level tokenization. Sec. D presents additional experiments examining multimodal complementarity under graph corruption, room-token capacity, text-serialized scene graph representations, performance across varying scene complexity, and computational efficiency.

B Benchmark Construction Details

B.1 Multi-Room Scene Assembly

CAIRN-MR is built on HM3D [44], a large-scale collection of realistic residential environments. We adopt the per-room annotations provided by SceneVerse [24], which include object identities, semantic labels, instance segmentations, and pairwise spatial relations for each room in the HM3D scenes. To construct multi-room scenes, we use the ground-truth room positions provided in HM3D. For each scene, we select a seed room whose spatial extent exceeds $30\,\mathrm{m}^2$, then compute Euclidean distances from its center to all other rooms and select the 3–5 nearest rooms to form a multi-room layout. To ensure spatial connectivity, we obtain the ground-truth occupancy map between sampled rooms via Habitat simulator [42] and discard any sample that contains an isolated room unreachable from the rest. Since different seed rooms within the same base scene yield different multi-room combinations, a single HM3D scene can produce multiple distinct samples. The resulting benchmark comprises 479 training samples from 116 base scenes and 194 validation samples from 45 base scenes, partitioned at the building level with no scene overlap.

B.2 Referring Expression Construction

We construct object-level referring expressions from the pairwise spatial relations provided by SceneVerse [24], which cover 11 relation types including support, containment, proximity, and directional positioning (see Tab. 6 for the full inventory). Each relation is a triplet (o_i, r, o_j) linking two objects within the same room. From these triplets, we construct referring expressions in three forms of increasing structural complexity.

1) Multi-reference. For a target object, we retrieve 2–3 relations that each directly involve the target, and combine them into a single expression via conjunction templates. SceneVerse's multi-object relations (e.g., aligned, in the middle of) naturally fall into this category. Example: "the table **between** the sofa **and** the bookshelf, **next to** the lamp."

2) Multi-hop chain. We identify a chain of three consecutive relations within the same room, where the target object appears at any position along the chain. The chain provides progressively more spatial context for disambiguation. Example: "the chair **next to** the desk **that supports** the monitor **near** the window."

Table 6: Spatial relation types used for referring expression construction.

Relation type	Example phrases
Support	resting on, placed on
Embedded	embedded into
Inside	inside
Hanging	hanging on, mounted on
Above / Below	above, below
Directional (near)	to the left of, behind
Directional (far)	far from
Proximity	close to, next to
Aligned	aligned with
In the middle of	in the middle of

3) Chain with branch. We first identify a relation chain of two or more hops (as in form 2), then connect the target object to any node on the chain via an additional relation. The chain serves as room-level context, while the branch relation localizes

the target. Example: "the lamp **above** the nightstand, where the nightstand **is next to** the bed **that faces** the window."

Prior to expression construction, we exclude objects with structurally ambiguous or overly generic labels (e.g., wall, floor, ceiling, object, furniture) and objects with fewer than 128 surface points in the reconstructed point cloud, ensuring that all participating instances are semantically specific and geometrically well-reconstructed. After construction, each expression is verified against the full multi-room scene to ensure it uniquely identifies the intended target. Specifically, for each expression targeting an object of class c with spatial constraints $\mathcal{C} = \{(r_k, c_k)\}_{k=1}^{K}$ (where each constraint specifies a relation type r_k and a reference object class c_k), we enumerate all objects of class c across the entire scene and check whether each candidate simultaneously satisfies all constraints in $\mathcal{C}$. An expression is retained only if exactly one object—the intended target—matches; all others are discarded. Since this verification is performed on the assembled multi-room scene, uniqueness is guaranteed at the full-scene level rather than within a single room. The verification procedure is formally described in Algorithm 1.

Algorithm 1: Full-scene uniqueness verification for a referring expression.

Input : Multi-room scene $\mathcal{S}$ with object set $\mathcal{O}$;
candidate expression e targeting object $o^\star \in \mathcal{O}$ of class c;
spatial constraints $\mathcal{C} = \{(r_k, c_k)\}_{k=1}^{K}$ parsed from e
Output : Boolean RETAIN$(e) \in \{\text{true}, \text{false}\}$

$\mathcal{M} \leftarrow \emptyset$ `// matches across the full scene`
foreach $o \in \mathcal{O}$ ***with*** `Class`$(o) = c$ **do**
  $ok \leftarrow$ true
  foreach $(r_k, c_k) \in \mathcal{C}$ **do**
    if $\nexists\, o' \in \mathcal{O}$ ***with*** `Class`$(o') = c_k$ ***and*** `Satisfies`(o, r_k, o') **then**
      $ok \leftarrow$ false
      break
    end
  end
  if ok **then** $\mathcal{M} \leftarrow \mathcal{M} \cup \{o\}$
end
if $|\mathcal{M}| = 1$ ***and*** $\mathcal{M} = \{o^\star\}$ **then**
  return true `// unique match: retain expression`
else
  return false `// ambiguous or wrong match: discard`
end

B.3 Appearance-Based Noun Augmentation

To increase linguistic diversity, we optionally augment object nouns with visually grounded appearance modifiers. This augmentation operates on a per-object basis and proceeds as follows.

For each object instance, we render the assembled multi-room scene and select multi-view RGB frames in which the object is visible, based on its 3D position and camera coverage. We then run SAM 3 [3] on each frame using the object's category label as the text prompt, and retain only frames in which the object is detected with confidence above 0.7. Objects with fewer than 3 qualifying frames are excluded from augmentation entirely, as insufficient high-confidence views indicate either poor reconstruction quality or heavy occlusion. For objects with more than 5 qualifying frames, we retain the 5 frames with the highest detection confidence to limit computational cost. The detected bounding boxes are used to crop tightly around the object in each selected frame, producing a set of 3–5 cropped views per eligible object.

For each eligible object, we independently query Qwen3-VL-32B [47] on each of its 3–5 cropped views. The model is provided with the object's category label and asked to identify a single clearly visible attribute from three types: color, material, or distinctive part. The model may also output "none" if no attribute is clearly identifiable. The full prompt template is shown in Fig. 5.

Prompt Template for Appearance Augmentation

```
You are given a cropped image of a {category}. Select ONE visual attribute
that is clearly visible:
- color (e.g., white, brown, black)
- material (e.g., wooden, metal, glass)
- shape (e.g., round, rectangular, L-shaped)
If none is clearly identifiable, output: none
Output format: [attribute] {category}
Examples: wooden chair, black table, round rug, none
```

Figure 5: Prompt template for querying Qwen3-VL-32B. `{category}` is replaced with the object's semantic label at runtime.

To suppress hallucination, we apply multi-view consensus: an attribute is accepted only if it appears in at least half of the queried views for that object. Objects for which no attribute achieves consensus retain their original noun without modification. Under this procedure, 21.6% of object nouns in the benchmark are augmented with an appearance modifier. The augmentation does not alter target identity or spatial constraints in the associated expressions.

B.4 Task Instantiation

All benchmark tasks are instantiated from the validated referring expression pool described in Sec. B.2. We partition the expression pool into two equal halves per scene using a fixed random seed: one half is used for grounding, the other for captioning.

Grounding. Each sample maps a referring expression to its target object identifier. Queries are drawn from three prompt templates, where `{description}` is replaced by the referring expression at runtime:

```
‘‘Find the object that matches this description: "{description}"’’
‘‘Which object in the scene is being described? "{description}"’’
‘‘Identify the object being referred to: "{description}"’’
```

The expected answer is the object identifier (e.g., `<OBJ041>`).

Captioning. Each sample maps an object identifier to a referring expression. Queries are drawn from four prompt templates, where `{id}` is replaced by the object identifier:

```
‘‘Explain the spatial position of <OBJ{id}>.’’
‘‘How does <OBJ{id}> relate spatially to the surrounding objects?’’
‘‘Describe the location and spatial context of <OBJ{id}>.’’
‘‘Describe where <OBJ{id}> is relative to other objects in the scene.’’
```

The expected answer is the original referring expression.

QA-I (Object Identification). Each sample presents a room-identifying spatial description and asks the model to identify a target object's class from four options. To construct the question, we prompt Qwen3-14B [47] to rewrite the target object's referring expression into an interrogative form that masks the target class name. For example, "the chair between the sofa and the bookshelf, next to the lamp" becomes "what is between the sofa and the bookshelf, next to the lamp?"—a transformation that is difficult to achieve reliably with rule-based templates due to the varied syntactic structures of multi-reference expressions. The rewritten query is then concatenated with the room-identifying spatial description as a preamble, and three distractor classes are sampled from the same room; the four options are randomly shuffled to ensure uniform answer distribution across A/B/C/D. Generated questions are validated to ensure the target class name does not appear in any form within the question text. Up to 3 questions are generated per target object. The full prompt template is shown in Fig. 6.

Prompt Template for QA-I Question Generation

```
Rewrite the following object description into a question that asks ‘‘what’’
or ‘‘which object’’ matches the description.  Do NOT include the target class
name anywhere in the question.

Target description:  {target_utterance}
Target class (to be masked):  {target_class}

Rules:
1) Output one line only, starting with ‘‘Q:’’ and ending with ‘‘?’’
2) Replace the target class with ‘‘what’’ or ‘‘which object’’
3) Keep all spatial constraints from the description
4) If you cannot comply, output:  INVALID

Example 1:
Input:  the chair is behind the lamp and in front of the cabinet
Class:  chair
Output:  Q: What is behind the lamp and in front of the cabinet?

Example 2:
Input:  the lamp next to the desk that supports the monitor near the window
Class:  lamp
Output:  Q: What is next to the desk that supports the monitor near the
window?
```

Figure 6: Prompt template for generating QA-I questions. The target class name is masked in the output to form an identification question.

QA-II (Counting). Each sample asks the model to count instances of a specified class within the room identified by the spatial description. Only classes with 2–10 instances in the target room are included. The spatial description is wrapped into a room phrase and combined with one of five templates, where `{room}` denotes the room phrase and `{class}` the pluralized category name:

```
‘‘In {room}, how many {class} are there?’’
‘‘In {room}, what is the total number of {class}?’’
‘‘How many {class} can be found in {room}?’’
‘‘How many {class} are there in {room}?’’
‘‘In {room}, how many {class} do you see?’’
```

The expected answer is the integer count.

QA-III (Existence Verification). Each sample asks whether a specified class is present in the room identified by the spatial description. For positive samples, the queried class must be present in the localized room but not mentioned in the spatial description, preventing trivial lookup. For negative samples, the queried class is drawn exclusively from adjacent rooms and must not appear in the localized room, ensuring non-trivial distractors. Positive and negative samples are balanced 1:1 per scene by truncating to the minority count. Five prompt templates are used, with singular and plural variants selected based on the queried class, where `{room}` denotes the room phrase, `{art}` the appropriate article, and `{class}` the category name:

```
‘‘In {room}, is there {art} {class}?’’
‘‘Can you find {art} {class} in {room}?’’
‘‘Does {room} contain {art} {class}?’’
‘‘Is {art} {class} present in {room}?’’
‘‘In {room}, is {art} {class} present?’’
```

The expected answer is “Yes.” or “No.”.

QA-IV (Inter-Room Comparison). Each sample provides spatial descriptions for two distinct rooms and asks which contains more instances of a specified class. Room pairs are drawn from all combinations within a scene, and the queried class must appear in both rooms with differing counts. To ensure label balance, the assignment of rooms to options (A) and (B) is randomly swapped with

probability 0.5 at generation time. Three templates are used, where `{room_A}` and `{room_B}` denote the two spatial descriptions and `{class}` the pluralized category name:

```
‘‘Room A: "{room_A}".  Room B: "{room_B}".
Which room has more {class}?’’
‘‘Room A: "{room_A}".  Room B: "{room_B}".
In which room can you find more {class}?  ’’
‘‘Room A: "{room_A}".  Room B: "{room_B}".
Comparing both rooms, which has a greater number of {class}?’’
```

The expected answer is “A.” or “B.”.

C Implementation Details

C.1 Hyperparameters and Training

We use AdamW with base learning rate 5×10^{-6}, betas $(0.9, 0.999)$, and weight decay 0.02. The learning rate follows a cosine decay schedule with linear warmup over the first 10% of epoch 1 and a min-LR multiplier of 0.01. We use an aggressive max-grad-norm of 0.01, empirically chosen to stabilize training of the bias and graph modules under the small base learning rate; values in the more common range $[1.0, 5.0]$ produced unstable loss curves on the hierarchical components in our preliminary experiments. The per-GPU batch size is 16 over 8 GPUs, giving a global effective batch size of 128. Both Stage 1 and Stage 2 train for 3 epochs in bf16 precision. We use random seed 42 for all reported numbers.

The LLM is adapted via LoRA [19] with rank 16, alpha 16, and dropout 0.05, applied to all attention and MLP projections (`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`). In addition to LoRA parameters, the input token embeddings and corresponding LM-head rows are unfrozen for the introduced object identifier tokens (`<OBJ_001>`, . . ., `<OBJ_300>`) to enable end-to-end learning of grounding outputs.

All experiments are conducted on 8× NVIDIA H200 GPUs (141 GB HBM3e per card), with steady-state training memory of approximately 80 GB per GPU. The two-stage training takes approximately 12 wall-clock hours per backbone: Stage 1 (3 epochs on ScanNet-based tasks) takes ∼4 hours, and Stage 2 (3 epochs on CAIRN-MR plus 30% ScanNet replay) takes ∼8 hours.

C.2 Architectural Details

Object Token Block. Each object is represented as a 4-token block $[t^{id}, t^{3D}, t^{2D}, t^{G}]$. The 3D and 2D tokens, t^{3D} and t^{2D}, are obtained by projecting pretrained features from Uni3D [65] (applied to per-object point-cloud segments) and DINOv2 [38] (applied to multi-view rendered images of the object), respectively. All four tokens are projected to the LLM hidden dimension via 2-layer MLPs. The maximum number of objects per scene is capped at 300; task instances targeting objects beyond this cap are excluded from training and evaluation. Empirically, this cap retains over 96% of task instances in CAIRN-MR and all task instances in the ScanNet-based benchmarks.

Message Passing for Graph Tokens. The graph token t_i^G in the object token block encodes relational context beyond per-object geometry and appearance, complementing the 3D and 2D tokens with information aggregated from neighboring objects. We instantiate the message-passing scheme outlined in Sec. 4.2 as follows.

For each object $\mathcal{O}_i$, we construct a room-local subgraph by connecting it to its K_g nearest in-room neighbors under the edge set $\mathcal{E}_{OO}$ defined in Sec. 4.3. Node features $\mathbf{h}_i^{(0)}$ are initialized by concatenating the per-object Uni3D point-cloud feature with a small geometric descriptor (center coordinates and bounding-box size). Edge features $\mathbf{e}_{ij}$ are constructed by concatenating a geometric component encoding relative position and Euclidean distance between $\mathcal{O}_i$ and $\mathcal{O}_j$, with a pretrained VL-SAT [49] relation embedding capturing semantic spatial relations between the pair (e.g., above, next to). Both edge components are precomputed and held fixed during training.

We then perform L_g layers of message passing:

$$\mathbf{h}_i^{(\ell+1)} = \mathbf{h}_i^{(\ell)} + \Phi\Big(\mathbf{h}_i^{(\ell)},\, \mathrm{AGG}\Big(\{\Psi(\mathbf{h}_j^{(\ell)},\, \mathbf{e}_{ij}) \mid j \in \mathcal{N}(i)\}\Big)\Big), \tag{7}$$

where Ψ computes a message from each neighbor's embedding and the edge attribute, AGG denotes mean aggregation, and Φ fuses the node's own embedding with the aggregated message. The final node embedding $\mathbf{h}_i^{(L_g)}$ is projected into the LLM hidden space by a 2-layer MLP to form the graph token t_i^G.

We use $L_g = 2$ message-passing layers with mean aggregation, following the standard depth in GraphSAGE-style architectures [16], where two layers suffice to propagate information across the immediate spatial neighborhood while avoiding the over-smoothing observed at greater depths.

D Additional Experiments

D.1 Multimodal Complementarity under Graph Corruption

Beyond evaluating robustness to scene graph corruption in Sec. 5.7, we further examine whether multimodal object features can compensate for inaccurate graph structure through controlled experiments under the same corruption settings.

Specifically, we independently train two architectural variants to isolate the contributions of different representations: **Feature-only**, which retains the 2D/3D object features but removes relational graph tokens, and **Graph-only**, which retains graph representations but removes the explicit 2D/3D object feature tokens. The full CAIRN model combines both representations. We evaluate all three variants under the same corruption settings and report grounding accuracy at the highest tested severity in Tab. 7.

As shown in Tab. 7, the full CAIRN model consistently outperforms both controlled variants across all tested corruption settings, demonstrating the benefit of combining multimodal object features with relational graph representations. The relative performance of Feature-only and Graph-only varies across corruption types. Under adjacency flips, Graph-only retains higher accuracy than Feature-only (23.5 vs. 19.8), whereas random object reassignment reduces Graph-only performance below Feature-only (18.1 vs. 19.8). These results suggest that relational representations remain informative under connectivity errors but are more sensitive to incorrect object-room membership.

The benefit of explicit multimodal features is particularly evident under unrestricted object dropout. Graph-only and Feature-only achieve comparable accuracy (14.6 vs. 14.4), whereas Full retains 18.0. This 3.4-point advantage over Graph-only suggests that explicit 2D/3D features provide complementary evidence when relational context is degraded. Nevertheless, the full model still suffers performance losses when objects are missing or incorrectly assigned, indicating that multimodal features can mitigate, but cannot fully eliminate, errors in the underlying scene structure.

Table 7: **Multimodal complementarity under graph corruption.** Grounding accuracy (Acc@0.25) at the highest tested corruption severity.

Corruption	Feature-only	Graph-only	Full
Clean predicted graph	19.6	23.6	**26.3**
Context dropout (30%)	20.3	20.9	**25.2**
Unrestricted dropout (30%)	14.4	14.6	**18.0**
Random reassignment (30%)	19.8	18.1	**21.6**
Boundary shift (100%)	19.8	21.7	**24.8**
Room merging ($\times 2$)	19.9	21.9	**24.7**
Adjacency flips ($\times 3$)	19.8	23.5	**26.1**

D.2 Comparison with Text-Serialized Scene Graphs

As an alternative to CAIRN's learned hierarchical representations, we construct a text-serialized graph baseline to assess the effectiveness of representing scene structure through natural language. As described in Sec. 4.1, the predicted hierarchical scene graph does not contain discrete semantic labels for individual objects. Instead, semantic information is encoded in continuous object features extracted by Uni3D. Therefore, constructing a semantic text representation requires an additional step to assign predicted category names to object proposals.

Specifically, building on the 2D/3D feature-only variant, we follow Uni3D's zero-shot classification protocol [65] to assign each object a predicted semantic category through feature-text similarity. We

Table 9: **Effect of room-token capacity.** Overall performance and crowded-scene performance difference (Δ) for different numbers of room queries. Crowded scenes contain $N \geq 143$ objects.

	Ground. (Acc@0.25)		Caption. (CIDEr)		Ident. (EM)		Count. (MRA)		Exist. (EM)		Comp. (EM)	
K_r	Avg.	Δ	Avg.	Δ	Avg.	Δ	Avg.	Δ	Avg.	Δ	Avg.	Δ
1	23.1	-7.1	90.9	-8.9	61.6	-0.7	59.9	-5.1	73.7	-1.1	67.4	-4.9
4	25.8	-5.3	**94.5**	-7.7	**62.1**	-0.2	61.1	-3.8	76.1	-0.6	**70.4**	-2.1
8	**26.6**	-5.5	93.2	-6.3	62.0	-0.1	**61.4**	-3.2	**76.4**	-0.2	69.9	-3.3

then serialize the predicted hierarchical scene graph into compact text descriptions, including object relations, object-room assignments, room connectivity, and the positions and sizes of object and room nodes. The resulting text replaces CAIRN's graph tokens, room tokens, and topology-aware attention mechanisms, while retaining the original 2D/3D object features. For a controlled comparison, the text-serialized variant uses the same LLM backbone, training data, and training objective as CAIRN. We compare this variant against Feature-only and full CAIRN across all six tasks in Tab. 8.

Table 8: **Comparison with text-serialized scene graphs.** Performance across six CAIRN-MR tasks.

Method	Ground. A@0.25↑	Caption. CIDEr↑	Ident. EM↑	Count. MRA↑	Exist. EM↑	Comp. EM↑
Feature-only	19.6	78.3	61.3	58.9	69.5	63.6
Text-graph	20.3	82.6	61.9	60.2	72.4	66.0
CAIRN	**25.8**	**94.5**	**62.1**	**61.1**	**76.1**	**70.4**

As shown in Tab. 8, the text-serialized graph consistently improves over the Feature-only variant across all six tasks, confirming that explicitly providing the predicted scene structure is beneficial. However, it remains below CAIRN on every metric, with particularly large gaps in grounding (20.3 vs. 25.8), captioning (82.6 vs. 94.5), existence (72.4 vs. 76.1), and comparison (66.0 vs. 70.4). These tasks require precise object localization and effective use of object-room and cross-room relationships, for which the text-serialized representation is less effective than CAIRN's learned hierarchical encoding. Overall, these results demonstrate the benefit of integrating scene structure through learned representations and topology-aware attention rather than relying solely on textual graph serialization.

D.3 Room-Token Capacity Analysis

To investigate whether a limited number of room tokens can adequately capture fine-grained object information in complex scenes, we analyze the effect of room-token capacity on downstream performance. As described in Sec. 4.2, CAIRN represents each room using $K_r = 4$ learned query tokens that aggregate information from its constituent objects through cross-attention, rather than compressing the entire room into a single representation. To assess the effect of query capacity, we additionally train and evaluate two variants with $K_r = 1$ and $K_r = 8$, alongside the default setting of $K_r = 4$.

We evaluate all three variants across the six CAIRN-MR tasks and further examine their performance on crowded scenes containing at least 143 objects ($N \geq 143$). In Tab. 9, Δ denotes the performance difference between crowded scenes and the overall average, measured in the corresponding metric units. Negative values indicate lower performance on crowded scenes.

As shown in Tab. 9, $K_r = 1$ consistently underperforms $K_r = 4$ across all six tasks, with particularly noticeable differences in grounding (23.1 vs. 25.8), captioning (90.9 vs. 94.5), and comparison (67.4 vs. 70.5). The single-query variant also exhibits larger performance degradation on crowded scenes across all tasks, suggesting that a single room query has insufficient capacity to preserve fine-grained object information as scene complexity increases.

In contrast, $K_r = 4$ and $K_r = 8$ achieve comparable performance across all tasks, with differences within 1.3 points and no consistent advantage from increasing the number of room queries. Although $K_r = 8$ slightly improves grounding, counting, and existence, $K_r = 4$ performs better on captioning, identification, and comparison. These findings suggest that additional capacity beyond $K_r = 4$ provides limited benefit at the scale of CAIRN-MR. We therefore adopt $K_r = 4$ as the smallest tested setting that avoids the single-query capacity bottleneck.

Table 10: **Performance across varying scene complexity.** Comparison between Inst3D-LMM + Room Tokens and CAIRN across different room-count and object-count bins. Each entry reports baseline / CAIRN, with higher values indicating better performance.

	Number of rooms		Number of objects		
Task (Metric)	2–3	4–5	$N \leq 81$	$82 \leq N \leq 127$	$N > 127$
Grounding (Acc@0.25)	22.6 / 27.5	19.4 / 24.6	39.6 / 42.7	25.1 / 28.3	17.8 / 22.4
Captioning (CIDEr)	83.2 / 97.8	76.9 / 92.1	91.4 / 106.0	83.1 / 99.4	75.5 / 92.0
Identification (EM)	61.4 / 61.8	62.5 / 61.1	63.9 / 64.4	62.3 / 61.8	60.2 / 60.9
Counting (MRA)	59.3 / 61.1	59.6 / 61.0	66.2 / 67.8	57.7 / 60.1	53.1 / 57.8
Existence (EM)	74.2 / 75.7	72.9 / 75.0	75.4 / 78.0	73.6 / 76.9	69.8 / 73.1
Comparison (EM)	70.4 / 75.6	62.3 / 69.8	68.8 / 73.9	64.8 / 69.9	62.7 / 67.5

D.4 Scaling with Scene Complexity

To better understand how scene complexity affects downstream performance, we conduct a systematic analysis on the full CAIRN-MR validation set, comprising 194 multi-room scenes from 45 buildings. We compare CAIRN against the strongest room-aware baseline, Inst3D-LMM + Room Tokens, using identical scene-complexity partitions. Specifically, we analyze three dimensions: the number of rooms M (2–3 vs. 4–5), the number of objects N ($N \leq 81$, $82 \leq N \leq 127$, and $N > 127$), and topological complexity measured by the cycle rank $\beta = |\mathcal{E}_{RR}| - M + 1$, where $\beta = 0$ indicates a tree-structured room layout and $\beta \geq 1$ indicates the presence of cycles. These analyses allow us to examine how increasing scene size and structural complexity affect different reasoning tasks.

As shown in Tab. 10, object count is the dominant observed factor affecting performance. Increasing the number of objects leads to degradation across all six tasks, particularly grounding, captioning, and counting. For example, CAIRN's grounding accuracy decreases from 42.7 in scenes with $N \leq 81$ to 22.4 in scenes with $N > 127$, while counting decreases from 67.8 to 57.8. These results indicate that increasing object density and the number of potential distractors remain substantial challenges for fine-grained scene understanding.

In contrast, increasing the number of rooms primarily affects tasks requiring room disambiguation and cross-room reasoning. Grounding and comparison show the clearest performance reductions when moving from 2–3 to 4–5 rooms, whereas identification, counting, and existence remain relatively stable. We additionally examine room-graph topology using cycle rank and find no clear performance dependence on whether the room layout is tree-structured or contains cycles, suggesting that topological complexity is not a dominant bottleneck within the evaluated scenes.

Across both room-count and object-count partitions, CAIRN outperforms the room-aware baseline in nearly every setting, with particularly clear advantages in grounding, captioning, and comparison. Moreover, CAIRN generally exhibits smaller performance reductions as scene complexity increases, although the trend is not uniform across all tasks. These findings suggest that CAIRN's hierarchical representations help maintain effective object-level and cross-room reasoning in more complex environments, while increasing object count remains a major source of difficulty.

D.5 Computational Efficiency and Memory

To evaluate the computational overhead of CAIRN, we profile the end-to-end inference pipeline and GPU memory consumption relative to Chat-Scene, which serves as a lightweight baseline using the same 2D/3D object features. We conduct profiling on a single NVIDIA H100 GPU with bf16 precision, batch size 1, and beam size 5 under identical decoding settings. Our measurements cover 10,000 validation queries across all task types and 194 multi-room scenes, containing up to 308 objects and 5 rooms.

End-to-End Computational Cost. We separately measure shared preprocessing and CAIRN-specific operations to identify the additional computational cost introduced by hierarchical scene modeling. Mask3D segmentation and Uni3D/DINOv2 feature extraction are shared between CAIRN and the baseline and are therefore excluded from the differential-cost comparison. Room decom-

Table 11: **End-to-end computational cost.** Runtime comparison between Chat-Scene and CAIRN on a single H100 GPU (bf16, batch size 1, beam size 5). Per-scene preprocessing is performed once and amortized across queries.

Stage	Chat-Scene	CAIRN	Additional cost
Room decomposition (HOV-SG)	–	48 ms/scene	+48 ms/scene (CPU)
Graph construction	–	0.6 ms/scene	+0.6 ms/scene
Structural encoders (GNN + room encoder + bias)	–	3.9 ms/query	+3.9 ms/query
LLM inference	254.6 ms/query	284.6 ms/query	+30.0 ms/query
Other (I/O)	9.1 ms/query	10.1 ms/query	+1.0 ms/query
Total per query	**263.7 ms**	**298.6 ms**	**+34.9 ms (+13%)**
Peak GPU memory	27.2 GB	27.8 GB	+0.6 GB (+2%)

position and graph construction are performed once per scene, while structural encoding and LLM inference are measured per query. Tab. 11 reports the median runtime over 10,000 queries.

As shown in Tab. 11, CAIRN introduces an additional 48.6 ms of preprocessing per scene, consisting of room decomposition and graph construction. Since these operations are performed only once per scene, their cost can be amortized across multiple queries. During inference, CAIRN adds 34.9 ms per query, corresponding to approximately 13% overhead relative to Chat-Scene. Notably, the topology-specific modules, including the GNN, room encoder, and geometric bias, account for only 3.9 ms per query. Most of the additional inference cost arises from the longer input sequence processed by the LLM. These results indicate that hierarchical scene modeling introduces relatively modest computational overhead compared with the underlying LLM inference.

GPU Memory Scaling. We further analyze how GPU memory consumption changes with the number of objects in a scene. Using the same profiling setup, we partition the 10,000 validation queries into four groups according to scene object count. Tab. 12 compares the peak GPU memory of Chat-Scene (Base) and full CAIRN (Full) across these groups.

Table 12: **GPU memory scaling with object count.** Peak inference memory on a single H100.

Objects per scene	Queries	Base	Full
$N \leq 81$	1,961	26.4 GB	26.8 GB
82–127	3,210	26.6 GB	27.1 GB
128–200	3,789	26.9 GB	27.4 GB
201–308	1,040	27.2 GB	27.8 GB

As the number of objects increases, CAIRN's peak GPU memory grows from 26.8 GB to 27.8 GB, an increase of only 1.0 GB across the evaluated scene sizes. Compared with Chat-Scene, CAIRN consistently introduces a small memory overhead of 0.4–0.6 GB across all object-count groups. This approximately constant overhead suggests that the additional GNN, room tokens, and hierarchical attention mechanisms do not substantially increase memory requirements as scene size grows. Memory consumption is dominated by the fixed 8B LLM backbone, and even the largest evaluated scenes require only 27.8 GB of GPU memory, fitting within the capacity of a single 40 GB GPU for inference.

Overall, CAIRN achieves substantial improvements in multi-room scene understanding with limited additional inference time and memory consumption, demonstrating the practical feasibility of incorporating hierarchical scene representations into existing 3D-LLMs.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the contributions, including the proposed CAIRN model, the CAIRN-MR benchmark, and the experimental findings. These claims are supported by both quantitative results (Sec. 5) and ablation analyses (Sec. 5.6), and are consistent with the scope of evaluation.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated discussion of limitations and future work (Sec. 6), covering aspects such as scaling to larger models and extending to more complex multi-level environments.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [No]

Justification: The paper does not include theoretical results or formal proofs, as it focuses on model design and empirical evaluation.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper describes the model architecture (Sec. 4), dataset construction (Sec. 3), training setup, and evaluation protocols (Sec. 5) in sufficient detail to allow reproduction of the main results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Code and data are not released at submission time to preserve anonymity. We plan to release the CAIRN-MR benchmark and implementation upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies training details, evaluation metrics, backbone models, and dataset construction procedures (Sec. 3, Sec. 4), with additional details provided in the Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The results are reported as single-run metrics without error bars. Due to the large-scale evaluation and consistent trends across tasks and backbones, we focus on comparative performance rather than statistical variance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides details on compute resources, including GPU type (8×H200), training duration (approximately 12 hours), and training setup in Sec. 5, which are sufficient to estimate the computational requirements for reproducing the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research complies with the NeurIPS Code of Ethics. It does not involve sensitive personal data or harmful applications, and follows standard practices in dataset usage and evaluation.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: The paper focuses on foundational research in 3D scene understanding and does not include a dedicated discussion of broader societal impacts. Potential impacts are indirect and not specific to deployment.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release models or datasets that pose a high risk of misuse, and therefore no specific safeguards are required.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets and models used (e.g., HM3D and prior benchmarks) are properly cited, and their usage follows the original terms and conditions.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces the CAIRN-MR benchmark. Its construction process, statistics, and task design are described in Sec. 3 and Appendix B, with plans for release upon acceptance.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human subject studies.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The work does not involve human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs are used as backbone models for our method and the baselines, and to assist CAIRN-MR construction (e.g., question rewriting and appearance augmentation). Details and prompts are in Appendix B.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.