# 3DMEM: SYNERGIZING TEXTUAL AND SPATIAL MEMORIES FOR CONVERSATIONAL 3D SCENE UNDERSTANDING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Existing 3D vision-language models are primarily developed for single-turn interactions, whereas natural dialogue requires a model to retain references and incorporate new constraints across turns. We introduce Conversational 3D Scene Understanding and ScanFlow, a benchmark for multi-turn question answering and grounding. Our framework, 3DMem, maintains two complementary recurrent states: a cumulative textual summary and a continuous spatial field aligned with scene superpoints. The spatial state is updated from the current causal prefix and previous predicted region, then read through language-side embeddings and residual modulation of grounding features. We construct continuous spatial supervision directly from point-cloud geometry, separating memory-region targets from exact instance-mask targets. Two supervised stages learn memory prediction and task responses using predicted memory trajectories, followed by IoU-aware optimization of the textual-memory policy with supervised grounding updates. This formulation connects evolving dialogue context to an explicit scene-indexed representation without generating discrete spatial buckets.

## 1 INTRODUCTION

3D scene understanding has evolved from closed-set geometric analysis to open-vocabulary, cross-modal reasoning. Early work focused on tasks like 3D object detection and semantic segmentation using fixed category sets. The rise of vision-language models introduced natural language supervision to 3D scenes, enabling tasks such as 3D visual question answering (VQA), language-grounded object localization, and open-vocabulary perception (Hong et al., 2023; Zhu et al., 2023; Azuma et al., 2022; Zhang et al., 2023). These methods align point cloud or multi-view representations with language semantics, allowing models to localize targets and reason in 3D space using text. Recently, the strong reasoning and instruction-following abilities of Large Language Models (LLMs) have shifted focus toward interactive 3D scene understanding. This paradigm uses language to query, analyze, and reason about complex 3D environments.

Existing 3D vision-language benchmarks, such as ScanQA (Azuma et al., 2022), primarily use a single-turn interaction paradigm that maps one instruction to one prediction. To avoid ambiguity, users must provide complete geometric, spatial, and appearance constraints in a single query. This differs from natural human dialogue, where users typically refine target descriptions across multiple turns, moving from coarse to precise references. Since current benchmarks focus on single-turn tasks, context does not transfer across turns, making it difficult to evaluate multi-turn 3D scene understanding. To address this, we introduce the task of **Conversational 3D Scene Understanding**, where a model takes a static 3D scene and a continuous sequence of user queries as input, and outputs corresponding textual responses or 3D grounding masks at each dialogue turn. By explicitly requiring the resolution of cross-turn coreferences and the accumulation of conversational context, this task effectively fills the evaluation gap for progressive target refinement. To support this task, we construct **ScanFlow**, a large-scale dataset comprising 17.3K dialogue turns organized into 3,524 multi-turn dialogues across 679 scenes. Built upon a rigorous human annotation pipeline, ScanFlow faithfully captures the natural human cognitive process of step-by-step spatial referencing, providing a comprehensive testbed for both multi-turn question answering and multi-turn 3D grounding tasks.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Current 3D-LLMs also lack mechanisms for multi-turn interactions. A simple workaround to accommodate multi-turn dialogues is to directly concatenate the dialogue history at the input. However, this approach fails to explicitly maintain a scene state that updates during the conversation. As a result, the model must reconstruct the alignment with the scene's geometric features at each turn. This makes it difficult to transfer spatial references from previous turns. As the dialogue continues, the model often misinterprets pronouns (e.g., "it" or "that") or ignores new constraints, leading to incorrect answers. The underlying issue is the lack of a cross-turn spatial-semantic memory (Jin et al., 2025; Ding et al., 2026). In multi-turn interactions, scene geometry is static, but language semantics constantly change. Without a mechanism to track semantic updates against fixed geometric features, models lose track of prior references, resulting in ambiguity and hallucinations. Therefore, a core question arises: *How can we establish a continuously updatable association mechanism within 3D-LLMs that allows the model to progressively accumulate linguistic semantics while maintaining stable correspondences with the static geometric features of the scene?*

To address this challenge, we propose **3DMem**, a dual-memory 3D-LLM that maintains a textual summary and a continuous spatial state across dialogue turns. Text memory records the referent and accumulated constraints, while spatial memory assigns an intensity to each superpoint in the scene. Its prediction depends on the current question, accumulated text, observable scene features, and the previous predicted region. We expose this state to the answer LLM through a compact embedding readout and to the grounding decoder through feature modulation. Continuous supervision is computed from raw point-cloud geometry, and ground-truth targets remain on the supervision side. This design provides an explicit interface for carrying spatial context across turns rather than requiring the LLM to express geometry as discrete labels.

Our main contributions are as follows: (1).We formally define **Conversational 3D Scene Understanding** to establish a multi-turn evaluation paradigm for continuous interactions and context accumulation in 3D scenes. (2).We construct **ScanFlow**, a 3D scene understanding benchmark designed for natural, multi-turn dialogues to evaluate models in continuous interactions. (3).We propose **3DMem**, a dual-memory 3D-LLM with recurrent superpoint-aligned spatial states and separate language and grounding readout paths. (4).We formulate geometry-derived continuous memory supervision and a training pipeline combining predicted-memory SFT with IoU-aware textual-memory policy optimization.

## 2 Related Works

**3D Vision-Language Understanding** Recently, researchers have integrated large language models (LLMs) with 3D representations to build multimodal models capable of instruction following and 3D reasoning. Current studies mainly follow two technical routes. The first route develops dedicated 3D-LLMs by integrating 3D encoders with LLMs. While some methods utilize rendered 2D features (Hong et al., 2023) or point cloud encoders (Chen et al., 2024a; Fu et al., 2025) for scene representation, others enhance fine-grained perception by incorporating object-centric features via 3D detectors (Chen et al., 2024b; Huang et al., 2023; 2024b;a). Additionally, researchers have improved scene understanding through voxel aggregation (Zhu et al., 2025) and specialized data generation strategies (Kang et al., 2025). The second route adapts Video LLMs by projecting 3D scenes into 2D sequences. To enhance spatial reasoning, current methods utilize 3D positional encodings for video-based alignment (Zheng et al., 2025), distill 3D priors from pre-trained foundation models (Zhu et al., 2023; Zhi et al., 2025), or establish cross-frame correspondences via visual prompts (Yu et al., 2023; Li et al., 2024; Jin et al., 2026a). Despite their progress, both routes primarily focus on single-turn tasks, leaving multi-turn interaction largely unexplored. Traditional methods align 3D geometry with language for tasks like VQA and grounding via cross-modal fusion (Azuma et al., 2022; Zhang et al., 2023; Wang et al., 2023b; Qiao et al., 2020; Parelli et al., 2023; Delitzas et al., 2023; Zhu et al., 2024). However, these works typically adopt a single-turn paradigm, which fails to capture the progressive constraints and coreferences inherent in natural multi-turn dialogues.

**Memory Mechanisms** To mitigate forgetting in long-context interactions and improve multi-turn consistency, memory mechanisms and state tracking have been extensively studied in LLMs and multimodal LLMs (MLLMs). In the text domain, existing methods primarily follow two directions. The first extends implicit context by optimizing KV caches or introducing linear attention mechanisms (Yu et al., 2023; Liu et al., 2026) to handle longer histories. The second manages explicit memory

banks, such as employing retrieval-style multimodal alignment (Wang et al., 2023a; Radford et al., 2021) for external knowledge or designing hierarchical memory architectures (Jin et al., 2026b) for long-term history retention. Additionally, dialogue-centric disambiguation and state maintenance (Taioli et al., 2025; Majumdar et al., 2023; Dai et al., 2024) are widely used to preserve user intent during interactions. In the 2D vision-language domain, visual dialogue and video understanding tasks rely on memory to ensure consistency across frames or turns. These methods typically use visual feature caches, recurrent networks, or object-centric representations (Yu et al., 2023; Ma et al., 2024; Yan et al., 2023; Zhang et al., 2024) to associate dynamic visual entities with dialogue history. Despite their success in text and 2D tasks, applying these mechanisms to 3D scenes remains challenging. Textual memory cannot capture 3D topology, and 2D visual caches are often incompatible with dense 3D representations. The core challenge arises from the asymmetry in 3D dialogues: while the scene geometry is static, user semantic constraints and spatial references evolve dynamically. Existing mechanisms struggle to associate shifting linguistic states with fixed 3D spatial features.

# 3 BENCHMARK

## 3.1 DATASET CONSTRUCTION

To synthesize multi-turn instruction-following data with physical anchors, we use a three-stage construction pipeline that aligns annotated 3D entities with multi-view observations. Geometric filtering supports this alignment but does not by itself guarantee the semantic correctness of every generated dialogue.

**Multi-granularity Filtering and Metadata Extraction.** Spatial hallucinations can arise when generated descriptions lack sufficient visual evidence. We use visibility filtering to select annotated targets with visual support. By projecting ScanNet point clouds onto image planes and checking depth consistency, the pipeline filters occluded candidates and retains targets exceeding a visibility threshold. From the selected viewpoint, we extract $\mathcal{M}_{obj} = \langle \text{Label}, \text{Center}_{3D}, \mathcal{B}_{3D}, \mathcal{B}^*_{2D} \rangle$, where $\mathcal{B}^*_{2D}$ is the target's 2D bounding box in that frame. This tuple supplies annotation-derived geometric anchors for offline dialogue synthesis; it is not a guarantee against errors in generated language.

**Cognition-Guided Dialogue Synthesis.** We utilize a multimodal large language model (Qwen3-VL-Max (Bai et al., 2025; Yang et al., 2025)) as the core synthesis engine. The LLM's input is strictly constrained to three dimensions: a representative high-resolution 2D view, the validated 3D metadata $\mathcal{M}_{obj}$, and task-specific prompts guided by cognitive strategies (the capability hierarchy is detailed in Table 1). This design forces the model to end-to-end simulate continuous dialogue streams, incorporating target tracking and spatial disambiguation within a standardized JSON format.

**Refinement and Human Verification.** To address potential logical drift in automated generation, we introduce a multi-stage refinement mechanism. Geometrically, we reconstruct predicted 3D bounding boxes from the responses and discard outliers with extremely low IoU against ground-truth annotations. Semantically, we apply a Minimum Bayes Risk (MBR) decoding strategy to retain stable outputs that align with common-sense consensus by maximizing semantic similarity among candidates. Finally, all dialogues undergo dual-independent manual 3D visual verification to correct implicit coreference errors and ensure high data fidelity.

Target metadata and reference responses are used offline for synthesis, filtering, and supervision. They are not supplied to the student at inference. Benchmark construction, continuous training-label construction, and policy rewards are separate procedures.

## 3.2 CAPABILITY HIERARCHY AND STATISTICS

Existing 3D vision-language tasks lack a systematic classification of cognitive capabilities. Therefore, we refer to Bloom's Taxonomy to deconstruct the 3D scene understanding task into three hierarchical capability dimensions: Perception, Understanding, and Interaction (detailed in Table 1). This taxonomy guides our data synthesis to ensure comprehensive coverage of the MLLM capability spectrum. The benchmark comprises 3,524 multi-turn interaction dialogues (totaling 17.3K dialogue turns) across 679 distinct 3D scenes. The overall distribution across task semantics is highly balanced: explicit grounding (20.6%), direct localization (20.3%), identity reasoning (15.6%), spatial reference (15.2%), visual reference (14.8%), and interactive disambiguation (13.5%). In terms of interaction

Table 1: Hierarchy of 3D multimodal understanding capabilities and task breakdown based on Bloom's Taxonomy.

| Level | Capability | Grp. | Task Definition & Requirements |
|---|---|---|---|
| Perception | Perception & Alignment | 1 | **Attribute Description**: Extract visual features (e.g., material, color). |
| | | 3 | **Semantic Interpretation**: Understand functional attributes and semantic roles. |
| | | 5 | **Coordinate Retrieval**: Precisely align text labels with 3D coordinates. |
| Understanding | Spatial & Logical Understanding | 2 | **Causal Inference**: Infer identities and behavioral causes from visual cues. |
| | | 4 | **Spatial Parsing**: Analyze global topological relations beyond a single view. |
| Interaction | Interaction & Tracking | 6 | **Interactive Disambiguation**: Simulates a "query–clarify–ground" loop, tracks dialogue states, and uses spatial memory to resolve references and localize targets. |

complexity, each sample features an average of 4.92 dialogue turns, with human instructions and model responses averaging 10.35 and 19.34 words, respectively.

### 3.3 Comparison with Existing Benchmarks

As shown in Table 2, existing benchmarks exhibit distinct limitations: they typically isolate visual QA (e.g., ScanQA) from 3D object grounding (e.g., ScanRefer) and rely heavily on single-turn static interactions. In contrast, our benchmark deeply couples QA with precise grounding within a continuous interaction loop. By introducing a cognitive hierarchy and multi-turn disambiguation mechanisms, our dataset advances the evaluation of 3D multimodal models from static single-step perception to dynamic deep cognition and reasoning.

### 3.4 Evaluation Metrics

**Basic Metrics:** For the 3D multi-turn QA task, we adopt standard NLP evaluation metrics, including BLEU-4, ROUGE-L, METEOR, and CIDEr. For the 3D spatial grounding task, we compute the Intersection over Union (IoU) between the predicted 3D bounding box and the ground truth, reporting Acc@0.25 and Acc@0.50 to objectively assess geometric localization precision.

**Autoregressive Protocol.** Model-generated history or recurrent memory is carried between turns without injecting reference answers. Grounding uses a freely generated `[SEG]` trigger. Missing triggers and invalid masks count as zero IoU in the all-sample mIoU and accuracy denominators; valid-mask-only mIoU is a separate diagnostic. Predicted masks are converted to AABBs in the same coordinate frame as the reference boxes. Comparisons must use the same dialogue IDs and scene split. Forced-trigger localization is reserved for separate mechanism diagnostics.

**LLM-as-a-Judge:** To overcome the limitations of traditional text metrics in evaluating complex reasoning, we introduce a comprehensive 5-point evaluation framework based on large language models (Bai et al., 2025; Yang et al., 2025; Wang et al., 2025). Utilizing Chain-of-Thought (CoT) for fact-checking, the judge focuses on three dimensions: 1) **Fine-grained Attribute Accuracy** (factual consistency of space, material, and shape); 2) **Multi-turn Context Coherence** (ability to track historical references); and 3) **Spatial Hallucination Rate**. See Appendix for detailed rubrics.

## 4 3DMem

### 4.1 Architecture Overview

3DMem maintains a compact dialogue state that connects evolving linguistic references to a static 3D scene. Its state at turn $t$ is $S_t = (T_t, m_t)$: $T_t$ is a cumulative textual summary, and $m_t \in [0,1]^K$ is a continuous spatial field aligned with the scene's $K$ superpoints. A Memory LoRA generates $T_t$, while a separate recurrent spatial head predicts $m_t$ from the scene, the current causal prefix, and the preceding spatial state. The Answer LoRA reads the textual summary and continuous spatial

Table 2: Comparison with existing 3D vision language benchmarks. QA denotes question answering, Grounding denotes 3D object grounding, Hier. Und. denotes hierarchical understanding, and Multi-turn denotes multi-turn interaction. Questions denotes dialogue turns.

| Dataset | Source | Questions | Scenes | QA | Grounding | Hier. Und. | Multi-turn |
|---|---|---|---|---|---|---|---|
| ScanQA | CVPR 2022 | 41K | 800 | ✓ | × | × | × |
| SQA3D | ICLR 2023 | 33.4K | 650 | ✓ | × | × | × |
| Multi3DRefer | ICCV 2023 | 61.9K | 800 | × | ✓ | × | × |
| ScanRefer | ECCV 2020 | 52K | 800 | × | ✓ | × | × |
| Nr3D | ECCV 2020 | 41.5K | 707 | × | ✓ | × | × |
| Sr3D | ECCV 2020 | 83.5K | 707 | × | ✓ | × | × |
| **Ours** | **-** | **17.3K** | **679** | ✓ | ✓ | ✓ | ✓ |

embeddings to generate a response; the mask decoder additionally receives spatially modulated superpoint features. Scene features are cached across turns, whereas both memory states are updated at every turn. Fig. 1 summarizes these connections.

**Base Model.** We initialize from 3D-LLaVA (Deng et al., 2025), which extends LLaVA-1.5-7B to point-cloud understanding. We retain its language backbone, 3D vision encoder, superpoint transformer, and grounding decoder. The language and 3D backbone weights remain frozen. Adaptation uses two LoRA modules (Hu et al., 2022), textual-memory vocabulary parameters, a continuous spatial head with its readout projections, and the grounding modules. The trainable subsets differ between stages, as described below.

**3D Scene Encoding.** The point-cloud encoder and superpoint transformer produce a visual sequence $V \in \mathbb{R}^{100\times D}$ for the LLM, scene-aligned state features $F \in \mathbb{R}^{K\times 1024}$, and grounding features $G \in \mathbb{R}^{K\times 32}$, where $D = 4096$. Each superpoint also has an observable center $c_k$ and point count $n_k$. The superpoint indexing is shared across dialogue turns. We cache these static features and recompute only the memory states and their readouts; the continuous memory does not alter the raw point cloud or feed tokens back into the superpoint transformer.

**Tokenizer and Vocabulary.** Structured text memory uses the  wrapper and field tokens, while the inherited `[SEG]` token triggers grounding. Added vocabulary rows are initialized from the mean of the existing embedding and output-head parameters. Spatial memory is represented by continuous tensors rather than XML or height and distance tokens. Legacy spatial tokens can remain in the checkpoint vocabulary for compatibility, but are not part of the revised spatial interface.

**Dual LoRA.** Separate Memory and Answer LoRA modules specialize the shared backbone for structured textual summaries and task responses, respectively. We switch adapters and their associated output-head weights between these operations. The spatial head is a separate deterministic module; it does not generate text tokens.

### 4.2 Synergistic Dual Memory Mechanism

At turn $t$, the memory predictor reads $V$, the current question $Q_t$, the preceding cumulative text memory $T_{t-1}$, and the preceding predicted spatial state $m_{t-1}$. Current answers, target-instance annotations, and exact target geometry are absent from this input. Earlier dialogue is carried through the recurrent state rather than a buffer containing only the most recent raw question–answer pair. We initialize $T_0$ to an empty summary and $m_0$ to a zero vector with an invalid-prior flag.

**Cumulative Text Memory.** The Memory LoRA produces a compact `<TextMem>` block containing `Turn`, `Target`, `Status`, and optional `Facts` and `Summary` content. The target describes the current referent; facts and summary retain relevant attributes and constraints from earlier turns. Status uses the vocabulary {`searching`, `proposed`, `confirmed`, `grounded`} without enforcing monotonic transitions. We commit the first complete generated block as $T_t$; if its wrapper cannot be parsed, we retain $T_{t-1}$. This fallback preserves an textual context but does not verify its semantic correctness.

**Recurrent Continuous Spatial Memory.** We represent spatial memory as a scene-indexed field $m_t \in [0, 1]^K$. Its entries encode target-region intensity, including geometric proximity, rather than discrete spatial labels or calibrated occupancy probabilities. We first obtain a query embedding $q_t$

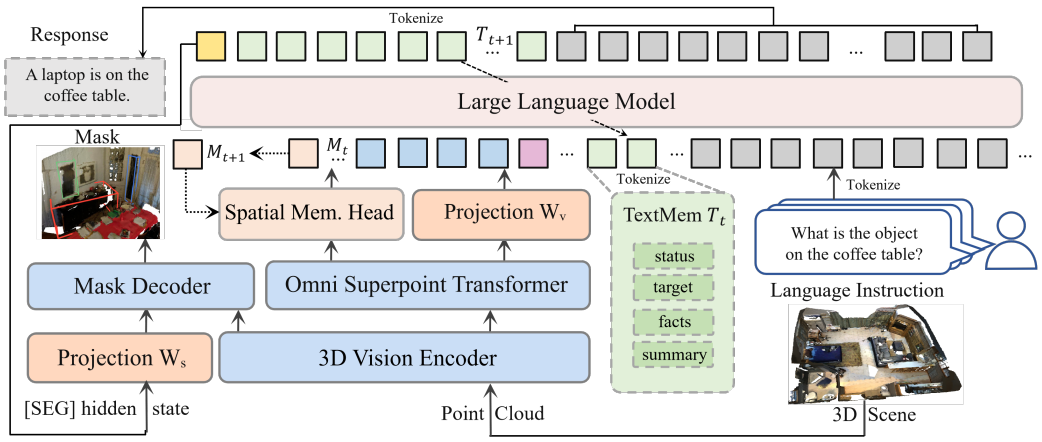



Figure 1: Overview of 3DMem. Scene features support recurrent spatial-memory prediction and visual embeddings for the LLM. The LLM consumes textual memory and spatial-memory embeddings to produce task responses, with a segmentation query supporting mask prediction. Textual and spatial states are carried forward across dialogue turns.

from the last hidden state of the causal prefix $(V, Q_t, T_{t-1})$ with the LoRA adapters disabled. For superpoint $k$, the spatial head computes

$$u_{t,k} = P_F(F_k) + P_C(c_k) + P_Q(q_t) + P_M(v_{t-1} m_{t-1,k}) + e(v_{t-1}), \qquad m_{t,k} = \sigma\big(f_{\text{region}}(u_{t,k})\big), \tag{1}$$

where $P_F, P_C, P_Q, P_M$ are learned projections and $e$ embeds prior validity $v_{t-1} \in \{0, 1\}$. This direct recurrent prediction allows the current question and accumulated text to revise the preceding region. The same vector coordinates refer to the same superpoints throughout a dialogue.

**Two Spatial Readout Paths.** Learned latent queries attend to the scene features augmented with $m_t$, producing four spatial embeddings $Z_t \in \mathbb{R}^{4\times D}$ for the Answer LLM. A second projection supplies a residual to the grounding features:

$$Z_t = \text{Readout}_\phi(F, C, m_t), \qquad G'_{t,k} = G_k + P_G(m_{t,k}). \tag{2}$$

The embeddings are inserted after the answer prompt and before its generated response; $G'_t$ is consumed by the mask decoder. At the next turn, we carry $m_t$ forward and compute a new readout. The embeddings themselves are not appended to a growing memory-token history.

**Answer and Mask Generation.** The Answer LoRA generates $A_t$ conditioned on $(V, Q_t, T_t, Z_t)$. For grounding, an actually generated `[SEG]` activates the mask decoder. Following the causal shift used during training, we project the last-layer hidden state immediately preceding this token into a grounding query. The decoder reads $G'_t$ together with cached point-cloud hidden features to predict a superpoint mask $\hat{h}_t$, which is mapped back to points. This output mask is distinct from the continuous memory field $m_t$. We retain $(T_t, m_t)$ as the state for the next turn.

### 4.3 Multi-Task Training Paradigm

Training consists of Memory SFT, Answer/Grounding SFT, and a subsequent GRPO phase. The first stage learns the recurrent memory representation, the second learns to consume predicted memory, and the third optimizes sampled textual policies while retaining supervised grounding updates.

**Stage 1: Memory SFT.** We train the Memory LoRA, textual-memory embedding and output-head parameters, and the continuous spatial module. TextMem is supervised with token cross-entropy. Spatial supervision is enabled only for causally observable prefixes, using a continuous geometric target $y_t$ and an exact instance-center target $c_t^*$. With $b_t \in \{0, 1\}$ denoting this supervision mask,

$$\begin{aligned}\mathcal{L}_{\text{mem}} = \mathcal{L}_{\text{text}} + \lambda_r b_t \Big[ &\mathcal{L}_{\text{BCE}}^{\text{point-weighted}}(m_t, y_t) + \lambda_d \mathcal{L}_{\text{Dice}}(m_t, y_t) \\ &+ \lambda_c \,\text{SmoothL1}(\bar{c}(m_t), c_t^*)\Big],\end{aligned} \tag{3}$$

where $\bar{c}(m) = \sum_k n_k m_k c_k / \sum_k n_k m_k$. Exact centers are supervision only. We oversample eligible dialogues to increase spatial training coverage without assigning final-target labels to ambiguous early prefixes. Geometric label construction is detailed in the Appendix.

**Stage 2: Answer/Grounding SFT.** We first run Stage 1 autoregressively to cache predicted TextMem trajectories. The Answer LoRA is then trained with these predicted summaries and the corresponding recurrent spatial states. We freeze the memory modules and train the Answer LoRA, segmentation bridge, alignment projection, and mask decoder. Language cross-entropy supervises task responses, including the `[SEG]` trigger on grounding turns; mask BCE and Dice supervise the exact instance mask $h_t^*$:

$$\mathcal{L}_{\text{task}} = \mathcal{L}_{\text{response}} + \lambda_{\text{seg}} \mathbb{1}_{\text{ground}} \left[ \mathcal{L}_{\text{BCE}}(\hat{h}_t, h_t^*) + \lambda_d \mathcal{L}_{\text{Dice}}(\hat{h}_t, h_t^*) \right]. \tag{4}$$

Grounding responses and masks can receive additional loss weights and sampling frequency. We retain the pretrained task output head rather than the head specialized for TextMem. Predicted memory, rather than a ground-truth memory substituted at the task input, matches the recurrent interface used at inference.

**Stage 3: IoU-Aware Policy Optimization.** We use the EasyR1 clipped policy-loss kernel within a 3D dialogue rollout. For each query, we sample a group of $G$ textual-memory trajectories and recursively compute the continuous state along each trajectory; for non-grounding queries, we additionally sample the Answer-policy response. On a grounding query, each sampled memory trajectory is evaluated with a supervised `[SEG]` continuation only for the IoU reward. The spatial module and its adapter-disabled query encoder remain frozen. Their outputs can still vary across trajectories because their textual inputs differ.

We assign rewards separately to the two actors. The format reward $r_i^{fmt} \in [0, 1]$ averages five checks: opening and closing TextMem tags, a non-empty target, a legal status, and a present facts field; legacy spatial XML incurs a penalty. The answer reward $r_i^{ans} \in [0, 1]$ averages ROUGE-L scores over sampled non-grounding responses. On the grounding turn, we evaluate a mask using a supervised `[SEG]` continuation and compute point-level mask IoU $r_i^{IoU}$. This training reward assesses localization conditioned on the sampled memory; it does not score autonomous trigger generation.

Let $\mathcal{N}_G(r_i) = (r_i - \mu_G(r))/(\sigma_G(r) + \epsilon)$ denote group normalization. Grounding credit is gated by successful TextMem commits along the trajectory, so $\tilde{r}_i^{IoU} = r_i^{IoU} \prod_t p_{i,t}$, where $p_{i,t}$ indicates a successful commit. The actor advantages are

$$A_i^{mem} = \text{clip}\left[ \mathcal{N}_G(r_i^{fmt}) + \lambda_{IoU} \mathbb{1}_{\text{ground}} \mathcal{N}_G(\tilde{r}_i^{IoU}), -5, 5 \right], \qquad A_i^{ans} = \mathcal{N}_G(r_i^{ans}). \tag{5}$$

Each sampled token receives the advantage of its own actor. With $\rho_j^a = \pi_\theta^a(u_j \mid u_{<j}, x)/\pi_{old}^a(u_j \mid u_{<j}, x)$, the policy objective is

$$\mathcal{L}_{\text{policy}}^a = -\mathbb{E}_{i,j}\left[ \min\left( \rho_j^a A_i^a, \text{clip}(\rho_j^a, 1 - \epsilon_{lo}, 1 + \epsilon_{hi}) A_i^a \right) \right], \qquad a \in \{mem, ans\}. \tag{6}$$

We regularize both policies toward their frozen SFT references using a KL penalty and an entropy term. The grounding turn does not contribute a sampled Answer-policy segment; instead, a supervised mask auxiliary loss updates the active Answer adapter and grounding modules. Thus IoU advantages optimize the textual-memory policy, while grounding parameters receive direct supervised gradients. The logged weighted reward sum is a monitoring statistic, not a shared advantage used by both actors.

## 5 Experiments

### 5.1 Training Data and Implementation Details

We reorganize training descriptions into multi-turn dialogues covering target confirmation, attributes, spatial relations, and final grounding. The training set contains 2,512 dialogues (12,361 turns), including 504 grounding episodes ending with `<ground>`. Text-memory labels provide linguistic supervision. Spatial-memory labels are continuous proximity fields computed from the raw target points; their losses are enabled only for eligible causal prefixes. Exact instance masks independently supervise grounding. Current answers, target IDs, target centers, and geometric labels are kept out

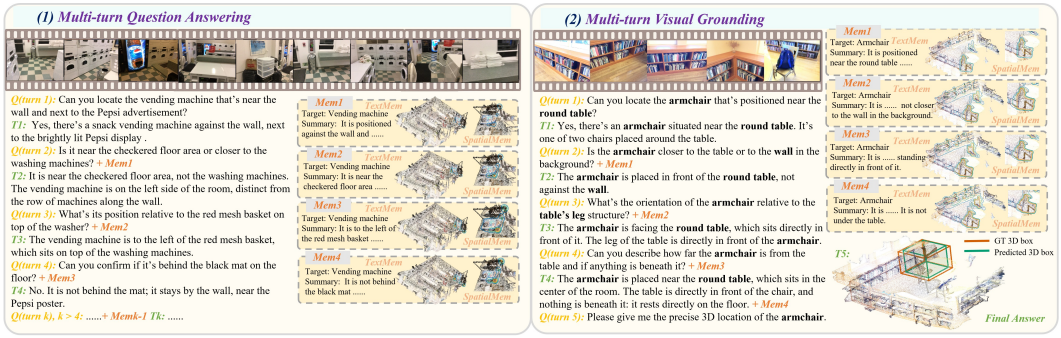



Figure 2: Qualitative comparison on multi-turn 3D question answering and visual grounding.

Table 3: Performance comparison under free generation: multi-turn 3D question answering and multi-turn 3D visual grounding. "–" marks a task that the method does not support.

| Model | Source | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | METEOR | ROUGE-L | CIDEr | LLM (Avg) | Acc@0.25 | Acc@0.50 | mIoU |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Video-3D-LLM | CVPR 2025 | 0.05 | 0.02 | 0.01 | 0.00 | 2.70 | 4.60 | 1.54 | 43.9 | 0.260 | 0.138 | 0.165 |
| 3DRS | NeurIPS 2025 | 0.01 | 0.00 | 0.00 | 0.00 | 1.68 | 4.21 | 1.06 | 57.1 | 0.246 | 0.125 | 0.153 |
| LL3DA | CVPR 2024 | 1.39 | 0.67 | 0.37 | 0.22 | 2.98 | 7.72 | 2.63 | 47.5 | 0.105 | 0.052 | 0.084 |
| LLaVA-3D | ICCV 2025 | 0.64 | 0.35 | 0.20 | 0.11 | 3.22 | 7.63 | 3.28 | 57.8 | – | – | – |
| 3D-LLaVA | CVPR 2025 | 1.38 | 0.69 | 0.38 | 0.21 | 3.51 | 8.96 | 3.07 | 48.9 | 0.456 | 0.209 | 0.259 |
| **3DMem (Ours)** | - | **30.40** | **17.85** | **11.30** | **7.45** | **16.19** | **25.30** | **39.70** | **60.2** | **0.500** | **0.233** | **0.281** |

of model-visible prompts. The Appendix describes label construction and the separation between offline supervision and student inputs.

The Memory LoRA uses rank $r = 64$ and scaling $\alpha = 64$; the Answer LoRA uses $r = 32$ and $\alpha = 64$. Stage 1 jointly trains textual memory and the continuous spatial-memory module for five epochs with gradient accumulation of four. Dialogues containing eligible continuous-region targets are sampled three times per epoch; learning rates are $10^{-5}$ for the Memory adapter, $10^{-3}$ for newly introduced tokens, and $3 \times 10^{-4}$ for the continuous module. Stage 2 consumes the Stage 1 predicted-memory cache and trains for one epoch on two GPUs with gradient accumulation of four and fourfold grounding-dialogue sampling. Its learning rates are $2 \times 10^{-5}$ for the Answer adapter and $10^{-5}$ for the grounding projection and mask decoder; the grounding language and mask objectives are each scaled by 10. Both SFT stages use a 5% warm-up and bfloat16 precision.

The policy phase uses AdamW with $\beta = (0.9, 0.95)$ and one grounding query plus one language query per update. We sample $G = 4$ trajectories per query and set $\lambda_{IoU} = 0.2$. Adapter learning rates are $5 \times 10^{-7}$ for Memory and $10^{-6}$ for Answer; grounding projection and decoder learning rates are $10^{-5}$ and $5 \times 10^{-7}$, respectively. Policy clipping uses $(\epsilon_{lo}, \epsilon_{hi}) = (0.2, 0.28)$ with separate KL scales for the two actors. Training rollouts retain at most the final five turns of dialogue and initialize state at the beginning of this window, whereas evaluation rolls through the complete dialogue.

### 5.2 Multi-turn QA Evaluation

We compare against five recent 3D-LLMs spanning both dedicated 3D architectures and adapted video LLMs: LL3DA (Chen et al., 2024a), LLaVA-3D (Zhu et al., 2025), 3D-LLaVA (Deng et al., 2025), 3DRS (Huang et al., 2025), and Video-3D-LLM (Zheng et al., 2025). We evaluate multi-turn question answering on the dialogue test set. As shown in the left block of Table 3, 3DMem achieves the best automatic language metrics across the board. Against the strongest per-metric baseline, it improves BLEU-1 from 1.39 to 30.40, BLEU-4 from 0.22 to 7.45, METEOR from 3.51 to 16.19, ROUGE-L from 8.96 to 25.30, and CIDEr from 3.28 to 39.70. These baselines are evaluated without task-specific SFT on ScanFlow; under standardized SFT (Table 5), margins narrow but 3DMem retains the best BLEU, CIDEr, LLM scores. The LLM score ranks models differently from lexical metrics because it penalizes hallucinations and rewards context tracking; 3DMem achieves the highest LLM score as well. Appendix A.5 analyzes judge calibration. The qualitative example in Fig. 2 further shows that competing methods confuse the target cabinet with nearby distractors or give unstable responses across turns. In contrast, 3DMem maintains compact text memory for user intent and spatial memory for anchors, preserving the referent and answering follow-ups.

Table 4: Inference-time ablation: continuous spatial memory, 120 scene-disjoint Group-5 dialogues. *w/o Spatial Readout* disconnects both language-side memory embeddings and grounding-feature modulation, whereas *w/o Region Recurrence* resets the previous-region prior at every turn.

| Setting | Enabled Component | | | Grounding Metric | | | | |
|---|---|---|---|---|---|---|---|---|
| | LLM Readout | Grounding Mod. | Region Recurrence | Acc@0.25 | Acc@0.50 | mIoU | [SEG] Rate | Valid Mask |
| w/o Spatial Readout | × | × | ✓ | 0.458 | 0.205 | 0.262 | 0.892 | 0.864 |
| w/o Region Recurrence | ✓ | ✓ | × | 0.483 | 0.217 | 0.272 | 0.858 | 0.858 |
| **Full** | ✓ | ✓ | ✓ | 0.500 | 0.233 | 0.281 | 0.975 | 0.880 |

Table 5: Comparison under standardized SFT on multi-turn 3D question answering and multi-turn 3D visual grounding. All baselines use unified LoRA and identical history concatenation.

| Model | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | METEOR | ROUGE-L | CIDEr | LLM (Avg) | Acc@0.25 | Acc@0.50 |
|---|---|---|---|---|---|---|---|---|---|---|
| Video-3D-LLM | 11.87 | 5.61 | 2.90 | 1.60 | 7.72 | 12.60 | 11.04 | 58.9 | 0.253 | 0.223 |
| 3DRS | 28.55 | 16.82 | 10.57 | 6.99 | 14.38 | **25.91** | 38.21 | 59.3 | 0.345 | 0.189 |
| LL3DA | 22.59 | 12.47 | 7.49 | 4.86 | **22.36** | 19.18 | 17.07 | 50.5 | 0.203 | 0.118 |
| 3D-LLaVA | 27.50 | 16.00 | 9.95 | 6.45 | 15.42 | 24.10 | 35.20 | 56.5 | 0.456 | 0.209 |
| **3DMem (Ours)** | **30.40** | **17.85** | **11.30** | **7.45** | 16.19 | 25.30 | **39.70** | **60.2** | **0.472** | **0.215** |

### 5.3 Multi-turn Visual Grounding Evaluation

The multi-turn visual grounding task requires continuous language understanding and accurate 3D target segmentation while user constraints accumulate over a static scene. The right block of Table 3 compares all methods under free generation, where 3DMem attains the best Acc@0.25, Acc@0.50, and mIoU. For the matched 3D-LLaVA and 3DMem runs, we use the same scene-disjoint dialogues, autoregressive histories, free `[SEG]` triggering, and all-sample denominator; missing triggers and invalid masks receive zero.

### 5.4 Analysis

**Ablation of Continuous Spatial Memory.** Table 4 applies paired inference-time interventions to the same checkpoint and the same 120 dialogue IDs. *w/o Spatial Readout* preserves recurrent region prediction but disconnects both downstream consumers: the spatial embeddings supplied to the LLM and the residual modulation of grounding features. *w/o Region Recurrence* retains both readout paths but resets the previous-region prior at each turn. These interventions are component-use diagnostics rather than separately retrained variants.

**Scope of the Diagnostic.** The ablation retains cumulative TextMem, the current question, and scene features in every setting, and changes only how the continuous spatial state is propagated or consumed. It therefore tests whether the trained checkpoint uses the spatial readout and cross-turn recurrence; it does not estimate the performance of independently optimized architectures.

**Comparison via Standardized Fine-Tuning** To disentangle architecture from training data, we re-evaluate all baselines under standardized SFT using the same multi-turn training set, a unified LoRA configuration ($r$=16, frozen backbone), and identical autoregressive history concatenation $[Q_1, \hat{A}_1, \ldots, Q_t]$. Table 5 reports both the language-generation and the localization comparison under this matched protocol.

## 6 Conclusion

We introduce Conversational 3D Scene Understanding and ScanFlow to study context accumulation in multi-turn 3D interactions. Our framework, 3DMem, maintains cumulative textual memory and a recurrent continuous spatial field aligned with scene superpoints. The spatial state is consumed through language embeddings and grounding-feature modulation. Geometry-derived supervision separates continuous memory targets from exact grounding masks, while predicted-memory SFT aligns task training with inference. A subsequent policy phase assigns localization-aware credit to textual-memory trajectories and retains supervised grounding updates. This framework provides an explicit, trainable connection between evolving dialogue context and static scene geometry.

## REFERENCES


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*, pp. 19129–19139, 2022.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Jierun Chen, Fangyun Wei, Jinjing Zhao, Sizhe Song, Bohuai Wu, Zhuoxuan Peng, S-H Gary Chan, and Hongyang Zhang. Revisiting referring expression comprehension evaluation in the era of large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 513–524, 2025.

Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. Ll3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26428–26438, 2024a.

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*, 2024b.

Yinpei Dai, Run Peng, Sikai Li, and Joyce Chai. Think, act, and ask: Open-world interactive personalized robot navigation. In *2024 IEEE international conference on robotics and automation (ICRA)*, pp. 3296–3303. IEEE, 2024.

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

Jiajun Deng, Tianyu He, Li Jiang, Tianyu Wang, Feras Dayoub, and Ian Reid. 3d-llava: Towards generalist 3d lmms with omni superpoint transformer. *arXiv preprint arXiv:2501.01163*, 2025.

Jiayu Ding, Haoran Tang, Hongbo Jin, Wei Gao, and Ge Li. 3d instruction ambiguity detection. *arXiv preprint arXiv:2601.05991*, 2026.

Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual reasoning. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2195–2206. IEEE, 2025.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

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.

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. *Advances in Neural Information Processing Systems*, 37:113991–114017, 2024a.

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. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024b.

Xiaohu Huang, Jingjing Wu, Qunyi Xie, and Kai Han. 3drs: Mllms need 3d-aware representation supervision for scene understanding. In *Advances in Neural Information Processing Systems*, 2025. NeurIPS 2025.

Hongbo Jin, Qingyuan Wang, Wenhao Zhang, Yang Liu, and Sijie Cheng. Videomem: Enhancing ultra-long video understanding via adaptive memory management. *arXiv preprint arXiv:2512.04540*, 2025.

Hongbo Jin, Jiayu Ding, Siyi Xie, Guibo Luo, and Ge Li. Vista: Mitigating semantic inertia in video-llms via training-free dynamic chain-of-thought routing. *arXiv preprint arXiv:2505.11830*, 2026a.

Hongbo Jin, Rongpeng Zhu, Jiayu Ding, Wenhao Zhang, and Ge Li. Himac: Hierarchical macro-micro learning for long-horizon llm agents. *arXiv preprint arXiv:2603.00977*, 2026b.

Weitai Kang, Haifeng Huang, Yuzhang Shang, Mubarak Shah, and Yan Yan. Robin3d: Improving 3d large language model via robust instruction tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3905–3915, 2025.

Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pp. 323–340. Springer, 2024.

Lulin Liu, Dayou Li, Yiqing Liang, Sicong Jiang, Hitesh Vijay, Hezhen Hu, Xuhai Xu, Zirui Liu, Srinivas Shakkottai, Manling Li, et al. Egotl: Egocentric think-aloud chains for long-horizon tasks. *arXiv preprint arXiv:2604.09535*, 2026.

Chuofan Ma, Yi Jiang, Jiannan Wu, Zehuan Yuan, and Xiaojuan Qi. Groma: Localized visual tokenization for grounding multimodal large language models. In *European Conference on Computer Vision*, pp. 417–435. Springer, 2024.

Arjun Majumdar, Fei Xia, Dhruv Batra, Leonidas Guibas, et al. Findthis: Language-driven object disambiguation in indoor environments. In *7th Annual Conference on Robot Learning*, 2023.

Maria Parelli, Alexandros Delitzas, Nikolas Hars, Georgios Vlassis, Sotirios Anagnostidis, Gregor Bachmann, and Thomas Hofmann. Clip-guided vision-language pre-training for question answering in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5607–5612, 2023.

Yanyuan Qiao, Chaorui Deng, and Qi Wu. Referring expression comprehension: A survey of methods and datasets. *IEEE Transactions on Multimedia*, 23:4426–4440, 2020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Francesco Taioli, Edoardo Zorzi, Gianni Franchi, Alberto Castellini, Alessandro Farinelli, Marco Cristani, and Yiming Wang. Collaborative instance object navigation: Leveraging uncertainty-awareness to minimize human-agent dialogues. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18781–18792, 2025.

Peng Wang, Shijie Wang, Junyang Lin, Shuai Bai, Xiaohuan Zhou, Jingren Zhou, Xinggang Wang, and Chang Zhou. One-peace: Exploring one general representation model toward unlimited modalities. *arXiv preprint arXiv:2305.11172*, 2023a.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Xingrui Wang, Wufei Ma, Zhuowan Li, Adam Kortylewski, and Alan L Yuille. 3d-aware visual question answering about parts, poses and occlusions. *Advances in Neural Information Processing Systems*, 36:58717–58735, 2023b.

Bin Yan, Yi Jiang, Jiannan Wu, Dong Wang, Ping Luo, Zehuan Yuan, and Huchuan Lu. Universal instance perception as object discovery and retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15325–15336, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. *Advances in Neural Information Processing Systems*, 36:76749–76771, 2023.

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*, pp. 15225–15236, 2023.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d llm: Learning position-aware video representation for 3d scene understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8995–9006, 2025.

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*, pp. 3761–3771, 2025.

Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d capabilities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4295–4305, 2025.

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*, pp. 2911–2921, 2023.

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*, pp. 188–206. Springer, 2024.

# A Additional Details

## A.1 Training Dataset Construction

We construct dual-memory supervision labels based on existing 3D vision-language benchmarks (Azuma et al., 2022; Zhang et al., 2023; Hong et al., 2023; Huang et al., 2024a). The training data comprises 2,512 multi-turn dialogues with a total of 12,361 turns, including 2,008 question-answering (QA) dialogues and 504 grounding dialogues. A complete multi-turn dialogue terminating with the `<ground>` token is defined as a grounding sample, accounting for 20.06% (504/2,512) of the dataset.

**Text Memory Generation.** We retain the offline textual-memory annotations from the dialogue construction pipeline and restrict student generation to a `<TextMem>` block with turn, target, status, facts, and summary content. Teacher-generated labels may use the reference response during offline annotation. These labels are training targets rather than model inputs: the student prefix contains scene features, the current question, and the preceding textual state. For task SFT, preceding and current summaries are taken from Stage-1 predicted trajectories. Reference responses are used only for supervised loss and reward computation. Teacher annotation does not itself establish that a final object is identifiable from every earlier prefix; spatial supervision is gated separately.

**Causal Prefix Eligibility.** An exact region target is used only when the available prefix supports the target under the training loader's causal-observability rules. Prefixes that do not meet these rules retain textual supervision but have their region and center losses masked out. We do not copy the final grounding target into every earlier turn. This eligibility mask is distinct from the generated TextMem status field. Manual prefix audits provide diagnostics for reference ambiguity; they do not supply geometry to the model or imply that every training prefix has been independently human-verified.

**Continuous Spatial Label Construction.** Let $P_o$ be the raw point set of target instance $o$, $\mathcal{P}_k$ the points assigned to superpoint $k$, and $D_o$ the diagonal length of the target's AABB. We construct exact occupancy and a continuous proximity field as

$$h_{o,k} = \frac{|\mathcal{P}_k \cap P_o|}{|\mathcal{P}_k|}, \qquad y_{o,k} = \max\left(h_{o,k}, \mathrm{clip}\left(1 - \frac{d(c_k, P_o)}{0.25 D_o}, 0, 1\right)\right), \tag{7}$$

where $d(c_k, P_o)$ is the nearest-point distance from the superpoint center to the target point set. Both distance and AABB diagonal are measured in the same coordinate units. Target occupancy is preserved, while nearby superpoints receive graded intensities. These values are recomputed from geometry rather than obtained by assigning numerical values to legacy buckets. The memory loss uses $y_o$; grounding supervision uses $h_o$. Center supervision is derived from exact target occupancy and superpoint geometry in the same transformed frame as the prediction, rather than from the soft neighborhood field. A scene is transformed once per dialogue so that coordinates and superpoint assignments remain aligned across turns.

**Input–Supervision Separation.** Scene coordinates and superpoint centers are observable inputs. Instance identities, target point sets, exact target centers, and the proximity fields above remain supervision-side quantities. The recurrent input uses a previous predicted region rather than a ground-truth region. Continuous supervision is trained with point-count-weighted BCE, soft Dice, and an exact-center loss on eligible prefixes. Task training uses predicted textual-memory trajectories and recurrent spatial outputs. At inference, the memory head receives no target annotations and does not perform a lookup into a ground-truth distance table.

## A.2 Extended Dataset Construction Details

Figure 3 summarizes the three-stage offline construction pipeline for ScanFlow.

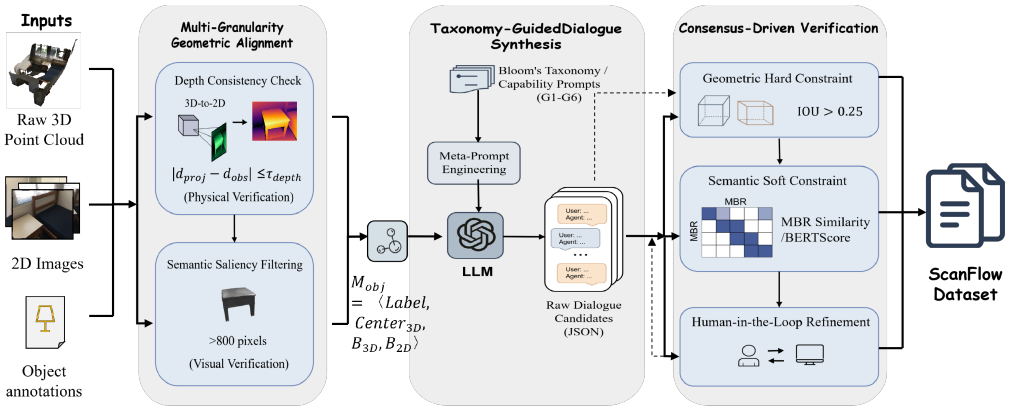



Figure 3: ScanFlow data construction pipeline. Phase I aligns annotated 3D targets with observable 2D evidence through depth-consistency and visual-saliency filtering. Phase II uses taxonomy-guided prompts and verified scene context to generate structured JSON dialogue candidates with an MLLM. Phase III filters candidates using a 3D-IoU constraint and MBR-based semantic agreement, followed by human-in-the-loop refinement. This offline benchmark-construction process is distinct from continuous spatial-memory label generation, and its target annotations are not exposed to evaluated models.

**Mathematical Formulation of Geometric and Visual Verification.** During the candidate pool construction phase, we apply rigorous geometric constraints to ensure physical visibility. Given camera intrinsics and extrinsics $\{K, T\}$, we project the 3D point cloud onto the image plane. We verify visibility by computing the residual between the projection depth $d_{proj}$ and the sensor observation depth $d_{obs}$. Objects satisfying $|d_{proj} - d_{obs}| > \tau_{depth}$ (where we set $\tau_{depth} = 0.2$ m) are deemed physically occluded and discarded. We also exclude sensor depth blind zones ($d_{obs} \leq 0$).

For visual saliency, we retain only objects exceeding a pixel threshold $\tau_{vis} = 800$. To reduce computational overhead while maximizing visual quality, we select the optimal observation frame $I^*$ for each target based on a maximum visibility criterion:

$$I^* = \arg\max_{I_t \in \mathcal{V}} N_{vis}(I_t), \tag{8}$$

where $N_{vis}(I_t)$ is the visible pixel count in frame $t$ from the video sequence $\mathcal{V}$.

**Semantic Refinement via MBR Decoding.** To filter geometrically inconsistent generated responses, we discard a response if its reconstructed 3D bounding box $\hat{\mathcal{B}}_{3D}$ yields an IoU less than 0.25 with the ground truth $\mathcal{B}_{3D}$. This check does not validate all statements in a dialogue. For semantic refinement, we apply Minimum Bayes Risk (MBR) decoding to select $R^*$ from candidates $\mathcal{H} = \{R_1, \ldots, R_N\}$:

$$R^* = \arg\max_{R_i \in \mathcal{H}} \frac{1}{N-1} \sum_{j \neq i} \mathcal{S}(R_i, R_j), \tag{9}$$

where $\mathcal{S}$ uses a DeBERTa-xlarge-based BERTScore to measure semantic similarity among candidates. This is an offline data-refinement score, not the GRPO answer reward, which uses ROUGE-L. Candidate agreement alone does not establish factual correctness.

### A.3 Prompt Group Definitions Based on Bloom's Taxonomy

Due to space constraints in the main text, we detail the specific design requirements and cognitive objectives of each Prompt Group here:

- **Group 1 (Attribute Description):** Focuses on the Perception level. Requires the model to extract fine-grained visual features of the target (e.g., material, color, shape).
- **Group 3 (Semantic Interpretation):** Focuses on the Perception level. Evaluates the model's understanding of an object's functional attributes and its semantic role in the scene.

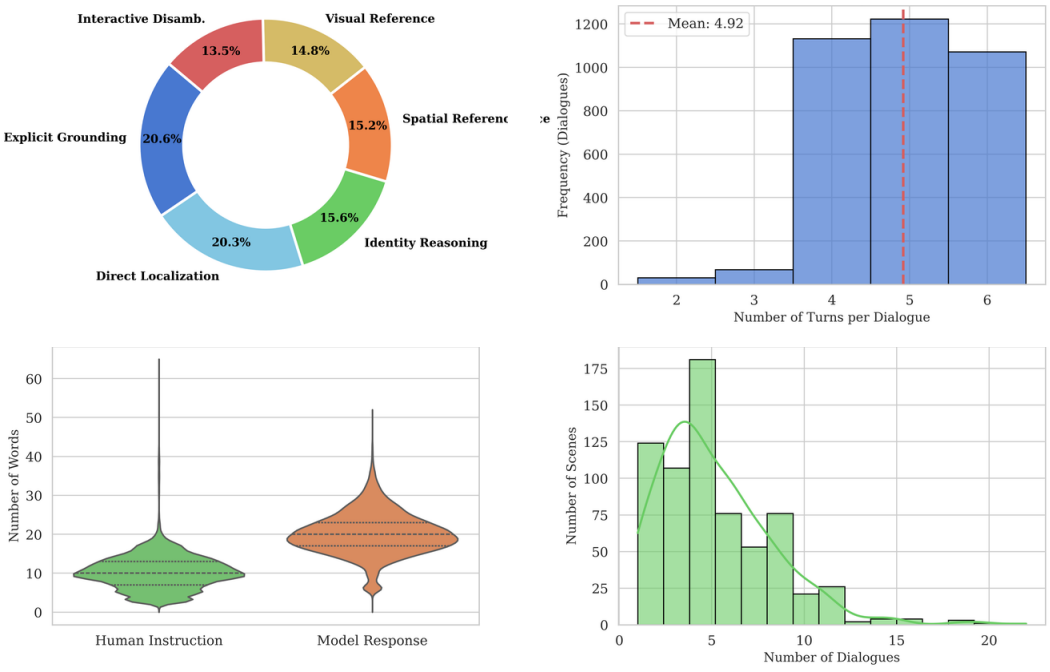



Figure 4: Statistical overview of ScanFlow: reasoning-task distribution (upper left), dialogue lengths (upper right), instruction and response lengths (lower left), and dialogues per scene (lower right).

- **Group 5 (Coordinate Retrieval):** A foundational alignment task requiring the model to precisely map text labels to underlying 3D spatial coordinates (bounding boxes).
- **Group 2 (Causal Inference):** Focuses on the Understanding level. The model must infer the identity of objects or the behavioral causes behind their presence based on local visual cues.
- **Group 4 (Spatial Parsing):** High-order spatial reasoning. Requires the model to transcend a single camera view and analyze global topological relationships and indoor layout logic.
- **Group 6 (Interactive Disambiguation):** Focuses on the Interaction level. Simulates a real-world "query–clarify–ground" loop. The model must track multi-turn dialogue states and use spatial memory to resolve references.

### A.4 ScanFlow Benchmark Statistics

The ScanFlow benchmark exhibits significant linguistic complexity and high information density. As illustrated in the lower-left panel of Fig. 4, human user queries average 10.35 words in length, while the reference model responses average 19.34 words, both exhibiting a pronounced long-tail distribution.

Beyond linguistic richness, the benchmark covers multiple dimensions: diverse reasoning tasks (upper left), scene diversity (lower right), and interaction complexity (upper right) in Fig. 4. Dialogues contain an average of 4.92 turns. These multi-turn queries require tracking references and constraints across the dialogue rather than considering each question in isolation.

### A.5 LLM-as-a-Judge Evaluation Details

**Motivation and Model Selection.** Standard lexical matching metrics (e.g., BLEU, ROUGE-L, CIDEr) often correlate poorly with human judgments in open-ended generation tasks. In multi-turn 3D dialogues, correct responses may use synonyms, partial information, or specific scene terminology absent from the ground truth. Therefore, we introduce an LLM-as-a-Judge evaluation protocol to complement pure text semantic evaluation.

We select Qwen3-Max (Bai et al., 2025; Yang et al., 2025) as our evaluation engine based on three considerations: (1) **Instruction following and structured output.** It demonstrates high stability in complex instruction following and formatted output tasks. (2) **Avoiding in-family scoring bias.** Its architecture differs significantly from all evaluated baseline models (which are based on Qwen-7B, LLaMA-3-8B, Flan-T5-XL, and T5 decoders). This minimizes the risk of scoring favoritism caused by similar underlying features. (3) **Deep reasoning capability.** It supports an extended-thinking mode for implicit Chain-of-Thought (CoT) reasoning, providing a transparent and auditable intermediate evaluation process.

**Prompt and Rubric Design.** To minimize scoring subjectivity and ambiguity, we design a behaviorally anchored 5-point rubric. Each score level is defined by specific, observable model behaviors rather than abstract descriptors: (1) **5 Points (Excellent):** Accurately identifies the specific entity or attribute, matches the detail level of the ground truth, and correctly retains context in multi-turn dialogues. (2) **4 Points (Good but concise):** The answer is correct and logically aligns with the context but is too brief (e.g., answering "wood" instead of "dark stained wood"). (3) **3 Points (Fair):** Correct at a macro-category level but overly broad (e.g., using "furniture" instead of "sofa"), or missing secondary 3D visual attributes. (4) **2 Points (Poor):** Contains explicit factual errors in core keywords or fails to provide the requested information. (5) **1 Point (Severe hallucination):** Forgets entities established in previous turns or exhibits severe hallucinations regarding 3D spatial topology.

During evaluation, the judge model is required to output a JSON object containing only two keys: "Reasoning" and "Score". This strictly enforces a 3-step CoT before generating the final numeric score: (1) **Keyword match and specificity**, which identifies core keywords and assesses their accuracy and detail; (2) **Context tracking**, which verifies whether the model retains and applies the dialogue history; and (3) **Final conclusion**, which cites the specific rubric anchor determining the score. This structured output forces explicit intermediate analysis, reduces anchoring bias, and makes failure modes auditable.

**Input Construction.** For each evaluated turn, the judge model receives four fields: Dialogue History, Current Question, Ground Truth, and Prediction. The Dialogue History consists of previous QA pairs derived from the model's own autoregressive predictions, reflecting actual error accumulation and propagation.

**Task Isolation.** Group 5 tasks, where ground truths are 3D bounding box coordinates, are explicitly excluded from the LLM text scoring system. Direct comparison of numerical coordinates by LLMs lacks geometric meaning, which could cause cross-modal hallucinations. Performance on this task is separately evaluated using objective 3D IoU metrics.

**Calibration and Consistency.** To verify the stability of the rubric, we uniformly sample 200 dialogue entries across all four baseline models and the remaining task groups. We run the evaluation twice using independent API calls with identical prompts. Results show an absolute score consistency of 89.5%. For samples with score disagreements, the Mean Absolute Deviation (MAD) is only 0.61 out of 5 points. This confirms the high discriminability and stability of the rubric anchors.

Furthermore, all baseline models are evaluated under identical judge configurations, including model weights, prompt templates, and decoding temperatures. Therefore, any residual judge variance is uniformly shared across all experimental conditions, preventing systematic bias toward any specific baseline.

**Scalability and Fault Tolerance.** To handle the massive evaluation volume, our scoring system operates across 8 concurrent API threads. The system includes automatic retry logic (up to 3 times) and checkpoint-based resumption. This enables fault-tolerant evaluation of approximately 55,000 interaction turns across all four models. For rare failure cases (e.g., API timeouts or JSON parsing errors), the system assigns a score of 0 and logs them separately. The overall evaluation failure rate is maintained below 0.3% for all models, ensuring statistical reliability.

Table 6: Error type breakdown per group within the 120 sampled **QA failure cases** (low-score text responses only; grounding failures are discussed separately in §C). XML-related failures (degradation + truncation + empty) dominate Groups 3 and 4, while factual hallucination dominates Groups 1 and 6.

| Group | $n$ | Factual | XML Degrad. | Truncated | Empty |
|---|---|---|---|---|---|
| G1 Attribute | 28 | 28 (100%) | 0 (0%) | 0 (0%) | 0 (0%) |
| G2 Causal | 13 | 10 (76.9%) | 2 (15.4%) | 0 (0%) | 1 (7.7%) |
| G3 Semantic | 44 | 4 (9.1%) | 39 (88.6%) | 0 (0%) | 1 (2.3%) |
| G4 Spatial | 10 | 0 (0%) | 3 (30%) | 6 (60%) | 1 (10%) |
| G6 Disambiguation | 25 | 20 (80%) | 2 (8%) | 1 (4%) | 2 (8%) |

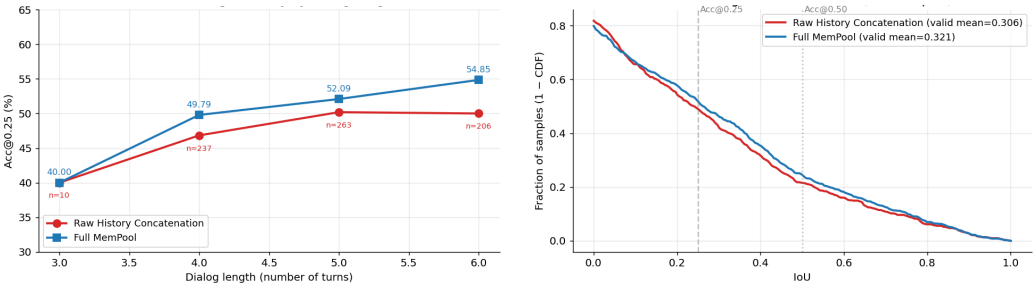



Figure 5: Historical analysis of spatial reference stability in multi-turn grounding with the original memory interface. Left: grounding accuracy by dialogue length. Right: IoU survival curves comparing Full MemPool with raw-history concatenation.

### A.6 XML Parse Failure Analysis

Table 6 reports the error type breakdown within the 120 sampled low-score failure cases used for the left panel of Fig. 6. All main-paper scores include these samples at their actual penalised values.

## B Mechanism Analysis

The following curves summarize the original discrete-memory experiments and are retained as historical diagnostics. They do not establish the performance of the revised continuous-memory model.

The left panel of Fig. 5 reports historical Group-5 grounding accuracy by dialogue length. Both methods start from 40.00% Acc@0.25 at three turns. Full MemPool reaches 49.79%, 52.09%, and 54.85% at four, five, and six turns, respectively, while raw-history concatenation obtains 46.84%, 50.19%, and 50.00%. The right panel shows IoU survival curves over 716 grounding samples. The original analysis reports valid mean IoU of 0.306 for raw history and 0.321 for Full MemPool. These observations motivate matched long-dialogue evaluation of the revised continuous state; they do not measure the revised model's performance or isolate the cause of the original gap.

## C Error Analysis

The sampled failures and empirical distributions in this section refer to the original model. We distinguish these observations from the revised implementation and from possible mitigation strategies.

**Quantitative Error Distribution.** We randomly sample 240 failure cases from the ScanFlow evaluation set: 120 low-scoring QA responses and 120 zero-IoU grounding predictions. The left panel of Fig. 6 shows that failure patterns differ sharply across groups. Group 1 (Attribute) failures are entirely factual hallucinations (28/28), and Group 6 (Disambiguation) follows a similar pattern (20/25, 80%). Group 3 (Semantic) is dominated by XML memory degradation (39/44, 88.6%), with almost no factual errors, indicating that semantic relational queries are conceptually tractable but impose stronger demands on structured memory formatting. Group 4 (Spatial) failures are predominantly output truncations (6/10, 60%), with XML degradation and empty outputs accounting

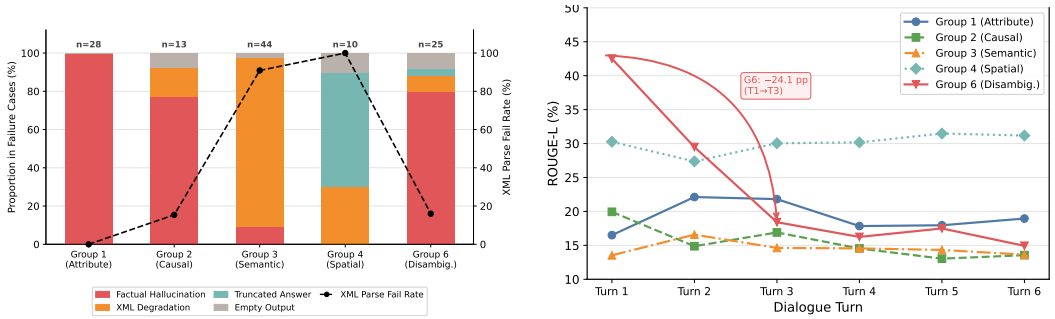



Figure 6: Historical error analysis of the original 3DMem on 240 sampled failure cases. Left: error type distribution per group. Factual hallucination dominates attribute and disambiguation queries, while XML memory degradation accounts for 88.6% of semantic group failures. All reported metrics include these degraded samples at their actual penalised scores. Right: per-turn ROUGE-L trend. Group 6 (Disambiguation) drops 24.1 pp from Turn 1 to Turn 3, reflecting early memory corruption that cascades across subsequent turns.

for the remainder. No factual hallucinations are annotated in this sampled subset; this absence does not establish that the model's spatial knowledge is correct. All XML-degraded and truncated samples are *included at their actual low scores* in every reported metric—they are never excluded.

**E1: Factual Hallucination.** The model generates responses that contradict observable visual properties of the scene, such as misidentifying the surface finish of a transparent object. This error concentrates in Groups 1 and 6, where fine-grained perceptual precision (material, reflectance, transparency) is required. The baseline exhibits the same failure pattern. This motivates examining shared perceptual limitations, but does not isolate the visual encoder as the root cause.

**E2: XML Memory Degradation.** Malformed XML output prevents the parser from extracting cross-turn context, corrupting the memory state for all subsequent turns in the same dialogue. This cascading effect is most severe in Group 3 (88.6% of failures), where the original structured interface was vulnerable to formatting errors. The revised model retains a structured TextMem summary, but represents spatial memory as a continuous tensor without generating spatial XML.

**Mitigation strategies.** Three practical remedies can reduce parse failures without architectural redesign: (1) **Constrained decoding** enforces valid XML tag sequences at generation time via finite-state automata; (2) **post-hoc repair** attempts to recover corrupted `<TextMem>` blocks before the next turn; (3) **format-robust fine-tuning** trains the model on partially-corrupted memory templates to improve resilience to self-generated noise. These are possible extensions; the current implementation retains the previous textual state when no complete block can be extracted. The continuous spatial field has no XML parsing step, although errors in the textual prefix can still affect its recurrent prediction.

**E3: Output Truncation and Empty Responses.** Group 4 (Spatial) shows a 60% truncation rate among sampled failures. This motivates checking token budgets and stopping behavior rather than attributing the failures to a particular module without a controlled test. The revised spatial field is predicted numerically, so it does not consume a spatial-XML generation budget; TextMem and task responses still require decoding.

**Per-turn Performance Decay.** The right panel of Fig. 6 reveals distinct degradation patterns across groups. Group 6 (Disambiguation) drops 24.1 pp from Turn 1 (42.5%) to Turn 3 (18.4%), as early memory corruption eliminates the accumulated disambiguation context. Groups 2 and 3 show a steady decline across turns, consistent with compounding memory formatting and reasoning errors. Group 4 (Spatial) remains stable at 27–31% ROUGE-L throughout all six turns. This pattern characterizes the original evaluation and must be rechecked for the revised memory interface before making a long-context robustness claim.

**Grounding Failure Modes.** Among the 120 zero-IoU grounding failures, the majority occur in scenes with multiple same-category instances (chairs, tables, shelves), where the legacy discrete SPATIALMEM representation may provide insufficient discriminative signal for fine-grained instance-level

disambiguation. This is a hypothesis, not an isolated causal explanation. The revised continuous field assigns separate intensities to scene superpoints; whether this improves same-category disambiguation requires an evaluation of the revised model.

# D QUALITATIVE CASES

Fig. 7 shows three historical examples. Left: in scene0228_00 (Group 1), Turn 4 asks about the texture of transparent drink bottles. The reference specifies a smooth, reflective finish. The baseline describes smoothness but omits the finish; the original 3DMem instead predicts a matte, non-reflective finish. Both responses fail to capture the reference attribute. Middle: in scene0213_00 (Group 4), Turn 6 asks what lies beneath a gray couch. The reference is a neutral-toned, ribbed carpet. The baseline predicts a brown ottoman, while the original 3DMem produces the stub "ottoman" after an XML parse failure at Turn 1 that persists through the six-turn dialogue. Right: in scene0492_00 (Group 5), six turns culminate in localizing a bathroom stall door. The reference location is $(4.87, 0.41, 0.97)$ in the benchmark's coordinate frame. The original 3DMem selects the door with IoU 0.93, whereas the baseline selects a far-left wall region with IoU 0.00.

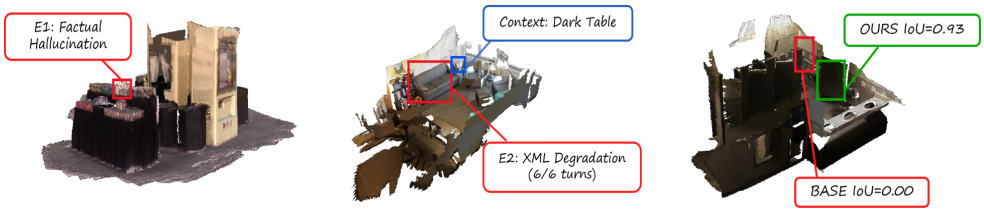



Figure 7: Historical examples of attribute error (left), memory-parsing failure (middle), and successful grounding (right). Case details are given in the text.

# E ETHICS, AI USE AND REPRODUCIBILITY

## E.1 ETHICS STATEMENT

ScanFlow is a benchmark of multi-turn dialogues grounded in static indoor 3D scenes. It contains no human subjects and no personally identifiable information, and it introduces no user-generated content beyond the RGB-D scans already distributed by the source datasets. Dialogue text refers to everyday indoor objects only; we do not annotate people, personal attributes, or private spaces. Human verification was performed by trained annotators under the authors' supervision, who inspected rendered 3D views together with reference annotations and corrected coreference and attribute errors. The benchmark is intended for research use, and we do not endorse its use in surveillance, biometric, or safety-critical decision systems. Because the task deliberately involves resolving referential ambiguity, failures can take the form of confident but incorrect localization; we therefore report the error statistics of Section C rather than presenting model output as a verified geometric measurement.

## E.2 DATA LICENSES

ScanFlow dialogues are synthesized over ScanNet scenes and the instance annotations distributed with public 3D vision-language benchmarks (Azuma et al., 2022; Zhang et al., 2023; Huang et al., 2024a). We follow the license and terms of use of each source. Raw point clouds, RGB-D frames, and camera parameters are not redistributed by us and must be obtained from the original providers.

## E.3 AI USE STATEMENT

We used generative AI tools for three purposes. First, a multimodal LLM (Qwen3-VL-Max (Bai et al., 2025; Yang et al., 2025)) served as the offline dialogue synthesis engine of the ScanFlow construction pipeline described in Section 3.1 and Figure 3. Second, an LLM-as-a-judge protocol was used as a complementary evaluation signal, with its rubric and model configuration documented

in Appendix A.5. Third, language models were used only for limited copy-editing of the manuscript text. Generative AI was not used to produce, select, or alter experimental results, tables, or figures, and no reported result is synthetic. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with AI assistance.

### E.4 Reproducibility statement

All components required to reproduce our results are specified in the paper. The dual-memory architecture, its readout paths, and the training objectives are given in Section 4; the continuous spatial label of Eq. equation 7, its construction, and the separation between student inputs and supervision-side targets are detailed in the appendix. Training-data composition, LoRA ranks, learning rates, sampling ratios, and the two-stage and policy schedules appear in Section 5.1, including the number of sampled trajectories, the clipping and KL coefficients, and the evaluation checkpoint. The autoregressive evaluation protocol, including how missing `[SEG]` triggers and invalid masks are scored, is specified in Section 3.4 and Appendix A.6. Dataset composition, the cognitive taxonomy, and benchmark statistics are reported in Section 3.2 and the appendix. Baseline configurations, including whether each method received task-specific SFT on ScanFlow (Table 5), follow the protocols stated in Section 5. We will release training and evaluation code, model adapters, and ScanFlow annotations. Results come from a single training run per configuration, and both the policy phase and the LLM judge are stochastic, so small run-to-run variation is expected.