3D-IDE: 3D Implicit Depth Emergent

Chushan Zhang[1] Ruihan Lu[2] Jinguang Tong[1] Yikai Wang[3*] Hongdong Li[1*]

[1]School of Computing
Australian National University

[2]School of Electrical Engineering and Computer Science
The University of Queensland

[3]School of Artificial Intelligence
Beijing Normal University

Abstract

Leveraging 3D information within Multimodal Large Language Models (MLLMs) has recently shown significant advantages for indoor scene understanding. However, existing methods, including those using explicit ground-truth 3D positional encoding and those grafting external 3D foundation models for implicit geometry, struggle with the trade-off in 2D-3D representation fusion, leading to suboptimal deployment. To this end, we propose 3D-Implicit Depth Emergence, a method that reframes 3D perception as an emergent property derived from geometric self-supervision rather than explicit encoding. Our core insight is the Implicit Geometric Emergence Principle: by strategically leveraging privileged geometric supervision through mechanisms like a fine-grained geometry validator and global representation constraints, we construct an information bottleneck. This bottleneck forces the model to maximize the mutual information between visual features and 3D structures, allowing 3D awareness to emerge naturally within a unified visual representation. Unlike existing approaches, our method enables 3D perception to emerge implicitly, disentangling features in dense regions and, crucially, eliminating depth and pose dependencies during inference with zero latency overhead. This paradigm shift from external grafting to implicit emergence represents a fundamental rethinking of 3D knowledge integration in visual-language models. Extensive experiments demonstrate that our method surpasses SOTA on multiple 3D scene understanding benchmarks. Our approach achieves a 55% reduction in inference latency while maintaining strong performance across diverse downstream tasks, underscoring the effectiveness of meticulously designed auxiliary objectives for dependency-free 3D understanding. Source code can be found at `github.com/ChushanZhang/3D-IDE`.

1. Introduction

Multimodal large language models (MLLMs) have rapidly become a unifying interface for 3D scene understanding, thanks to their strong 2D priors and powerful reasoning ability [1, 28, 29]. Recent work adapts MLLMs to 3D through several routes: point cloud encoders that project 3D features into language-aligned spaces [10, 43], multi-view approaches that lift 2D features to 3D representations [19, 55], and video-based methods that capture spatio-temporal relationships in 3D scenes [30, 54]. Among these paradigms, video-based MLLMs are especially attractive because they naturally preserve scene continuity and leverage pretrained video understanding capabilities.

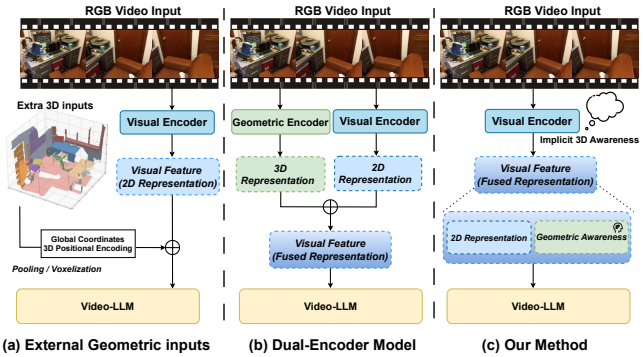


Figure 1. **Comparison of 3D-aware designs for video-LLMs.** **(a)** Explicit coordinate injection fuses 2D features with coarse 3D positional embeddings and requires 3D inputs at inference. **(b)** Dual encoders separately process RGB and geometry, then fuse their outputs, increasing complexity and latency. **(c)** 3D-IDE uses a single visual encoder trained so that 3D awareness emerges implicitly, enabling efficient RGB-only inference.

Within this video-based paradigm, existing methods mainly differ in how they inject geometric awareness. As illustrated in Fig. 1, one line of work explicitly encodes per-pixel 3D coordinates from depth and camera poses into visual tokens, requiring additional 3D inputs such as depth maps or camera poses at inference time [4, 24, 54]. Another line relies on a separate 3D foundation model whose features are fused with those of a 2D visual encoder, forming a dual-encoder design [7, 38, 46, 53]. Both strategies, however, have fundamental drawbacks. Explicit coordinate injection introduces a critical dependency on specialized 3D sensors and passes through aggressive downsampling and quantization, which collapses fine-grained geometry and lead to a "double information loss". Dual-encoder designs increase parameters and latency and create a repre-

sentation gap: 2D and 3D encoders are typically frozen and optimized under different objectives, so the language model is forced to act as a late-stage aligner instead of focusing on high-level 3D reasoning, ultimately limiting deployability.

This work asks a different question: can we learn a 3D-aware representation that uses only RGB video at inference, yet retains geometry strong enough for 3D grounding and reasoning? To answer this, we propose **3D-IDE** (3D-Implicit Depth Emergence), guided by the **Implicit Geometric Emergence Principle (IGEP)**. Rather than treating geometry as a mandatory input, we regard it as privileged supervision that is available only during training. A lightweight, training-only geometric validator and a global 3D teacher provide fine-grained and scene-level geometric signals that push the visual encoder to embed 3D structure directly in its tokens, without modifying the inference-time interface or introducing any additional inputs.

Concretely, the same visual tokens that condition the video-LLM are also required, during training, to support dense, uncertainty-aware depth prediction, localized cross-view consistency across neighboring frames, and alignment with a frozen 3D foundation model [38, 46]. Because the geometric head is deliberately low-capacity and discarded at test time, the model cannot rely on it as a 3D expert; instead, it must internalize geometric cues in the shared encoder. This implicit training pressure yields a single RGB-only representation that is geometrically informative, avoids explicit coordinate injection and separate 3D encoders, and adds no latency or extra inputs at deployment. Overall our main contributions are three-fold:

- We introduce the Implicit Geometric Emergence Principle, which views 3D awareness in video-MLLMs as an emergent effect of training-time geometric supervision rather than explicit 3D inputs or heavy 3D encoders.
- We realize this principle in **3D-IDE**, which uses a lightweight geometric validator and a global 3D teacher to impose uncertainty-aware depth, multi-view consistency, and scene-level constraints, while keeping inference strictly RGB-only.
- On standard 3D grounding, captioning, and QA benchmarks, 3D-IDE outperforms prior RGB-only video-MLLMs and remains competitive with methods using explicit 3D inputs, achieving up to 6.36% higher 3D grounding accuracy with 12.86% fewer parameters and 55.28% faster inference.

2. Related Work

MLLMs for 3D Scene Understanding. Adapting Multimodal Large Language Models (MLLMs) [1, 27, 28, 39] for 3D scene understanding has attracted significant interest. Early pioneering works focused on bridging the modality gap between 3D representations and language. These methods typically ingest point cloud data, which is processed by a specialized 3D encoder (like PointNet [34] or its variants [35]) before being projected into the MLLM's embedding space. Prominent examples in this category include PointLLM [42], 3D-LLM [19], Chat-3D [40], LL3DA [9], and Grounded 3D-LLM [12]. While effective for 3D-centric tasks, these approaches face two key challenges: the scarcity of large-scale, well-annotated 3D–text datasets and a fundamental disconnect from the rich 2D visual knowledge learned by MLLMs during large-scale pre-training. Recent empirical studies [16, 33] have shown that 2D pre-trained visual foundation models can effectively extract 3D spatial representations from 2D features, indicating that large-scale 2D pre-training inherently encodes structural priors of 3D scenes. Building on this insight, recent work has shifted toward video-based approaches [36, 54], which process 3D scenes as multi-view sequences or video frames to naturally preserve scene continuity, exploit pre-trained video understanding capabilities, and capture spatio-temporal relationships across frames.

3D-Aware Integration in Video-MLLMs. Within the video or multi-view paradigm, contemporary work incorporates 3D priors into MLLMs along two explicit routes and one implicit route. First, direct injection methods, exemplified by Video-3D LLM [54] and its variant 3DRS [24], treat 3D scenes as videos and augment patch-level visual tokens with global 3D coordinates computed from depth and camera poses, a strategy that is also adopted by LLaVA-3D [55] and GPT4Scene [36]. Second, explicit supervision or fusion methods refine these representations through geometric feature fusion: Vid-LLM [7] aligns MLLM visual features to those from pre-trained 3D foundation models such as VGGT [38] and FLARE [46], while VG-LLM [53] extracts priors such as inter-frame correspondences from RGB videos using a pre-trained 3D geometry encoder and fuses them with 2D tokens. In contrast, 3D-IDE advances an implicit route via the Implicit Geometric Emergence Principle, where geometry serves as a form of privileged training signal for an auxiliary validator, encouraging the MLLMs to internally emerge 3D-aware structure directly from monocular cues without relying on explicit 3D priors or any additional inference-time components.

3. Method

This section details our proposed 3D-IDE framework, which is designed to overcome the critical limitations of existing MLLMs in 3D scene understanding. In Sec. 3.1, we begin by formally revisiting and analyzing the distinct structural constraints of the two conventional paradigms, which motivates our approach. We then introduce the key principles of our proposed 3D-IDE framework in Sec. 3.2, centered on the Implicit Geometric Emergence Principle. Following this, we detail the specific architecture and com-

posite training objectives used to realize this principle in Sec. 3.3. The overall framework is illustrated in Fig. 3.

3.1. Preliminaries

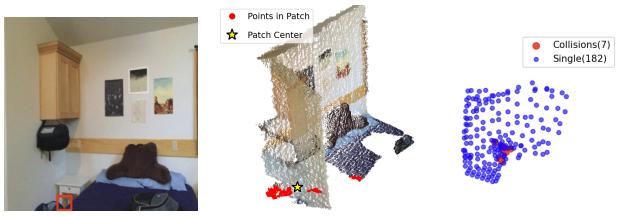
We first formalize the typical structure of 3D-aware video MLLMs and the two dominant paradigms used to inject geometric information. Let $\{I_t\}_{t=1}^{N}$ denote a set of multi-view images or video frames, and let f_E be a visual encoder. For each frame I_t, the encoder produces patch-level features

$$F_t = f_E(I_t) \in \mathbb{R}^{H' \times W' \times d},$$

where H', W' are the downsampled spatial dimensions and d is the feature dimension. Existing 3D-aware designs construct enriched features F_t^{3D} by combining F_t with 3D cues, typically from explicit inputs or external encoders.

Table 1. Performance gap between 3D-trained models with and without 3D inputs. On 3D VQA benchmarks, removing 3D signals reduces accuracy to the level of zero-shot 2D MLLMs.

Method	External 3D inputs	Scan2Cap	ScanRefer	Multi3DRefer
		C@0.5	Acc@0.25	F1@0.25
2D-MLLM (LLaVA-Next-Video) [48]	×	31.0	-	-
3D-MLLM (Video-3D LLM) [54]	×	31.5	53.7	46.0
3D-MLLM (Video-3D LLM) [54]	✓	83.8	58.1	58.0



(a) RGB input (b) Pooling loss (c) Voxelization loss

Figure 2. Illustration of the double information loss in explicit coordinate injection. **(a)** RGB frame with a 2D patch whose pixels are back-projected to point cloud. **(b)** Pooling collapses all patch points into one token, losing local structure. **(c)** Voxelization merges distinct 3D points into the same voxel, further degrading fine-grained geometry and harming downstream 3D reasoning.

Explicit 3D-Input. Methods in this family rely on explicit 3D inputs, such as depth maps. This reliance is a critical flaw, significantly limiting generalization and practical deployment in real-world scenarios where such data is unavailable. To quantify this dependency, we ablate the 3D ground-truth inputs during inference. As shown in Tab. 1, we observe a substantial performance degradation when 3D inputs are withheld. Despite being fine-tuned on 3D data, the model's performance collapses to a level comparable to a 2D MLLM (1st row) in a zero-shot setting. This confirms that existing models often use 3D inputs as a crutch, failing to develop robust, generalizable 3D understanding.

This method, used by [24, 54], requires a per-pixel 3D coordinate map $C_t \in \mathbb{R}^{H \times W \times 3}$ associated with each image I_t, computed by back-projecting the depth map using camera parameters. These coordinates are encoded via a positional encoding function $\phi(\cdot)$ and injected into the visual features:

$$F_t^{3D} = F_t + \phi(C_t) \tag{1}$$

This encoding process itself is highly problematic. As visualized in Fig. 2, the coordinates C_t are heavily downsampled and voxelized to align with patch-wise features. This coarse aggregation creates an information bottleneck, causing fine-grained geometric structures to be averaged out and rendered indistinguishable. This double information loss prevents the model from perceiving fine-grained geometry and exacerbates spatial ambiguity, especially when a single patch is geometrically complex or contains multiple objects.

External 3D-Encoder. A second family of approaches employs a dedicated geometric encoder E_{geo} to extract latent 3D features from the input frames. Given image features F_t from the visual encoder and geometric features $G_t = E_{\text{geo}}(I_t)$, the 3D-aware representation is obtained by a feature fusion $\oplus$, which is formulated as follows:

$$F_t^{3D} = F_t \oplus G_t, \tag{2}$$

Such geometric encoders are often large models (e.g., VGGT [38] has on the order of one billion parameters), which substantially increases the overall model size and inference cost and makes end-to-end optimization more demanding. In addition, E_{geo} is typically pre-trained and kept frozen under objectives different from those of the video MLLM, so the resulting feature spaces may be poorly aligned. As a consequence, the language model must implicitly learn to reconcile the 2D and 3D streams, limiting the effectiveness of late fusion for 3D reasoning.

LLMs with 3D visual tokens. Regardless of the method used to obtain F_t^{3D} (Eq. (1) or (2)), given a tokenized text instruction ψ, models in this dependency paradigm are optimized by minimizing the standard cross-entropy loss:

$$\mathcal{L}_{ce} = -\sum_{n=1}^{T} \log p_\theta(y_n | y_{<n}, \{F_t^{3D}\}_{t=1}^{N}, \psi), \tag{3}$$

where y_n is the n-th ground-truth output token, T is the total length of the target sequence, and θ denotes the set of all trainable parameters of the entire MLLM model f_θ. The analyses above highlight a critical trilemma in 3D-aware MLLMs: existing methods either (1) rely on unavailable ground-truth inputs, (2) destroy geometric information via coarse encoding, or (3) depend on massive, external 3D foundation models. These limitations motivate the need for a 3D representation that is simultaneously lightweight, information-preserving, and independent of ground-truth.

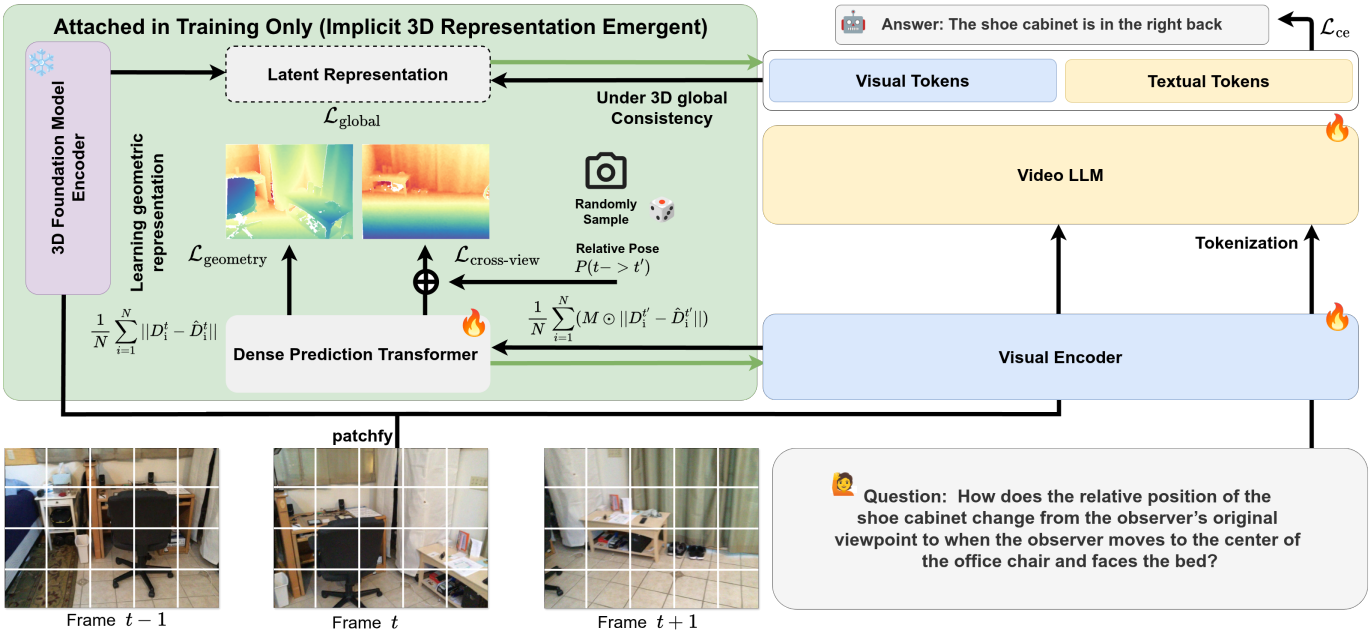


Figure 3. **The 3D-IDE framework**. Our approach avoids the "Double Information Loss" (see Fig. 2) inherent in explicit coordinate injection methods. Instead of injecting coarse, lossy coordinates, we use a privileged training module (green box) that is detached at inference for **zero latency**. This module forces the model to learn a fine-grained 3D representation implicitly via two parallel gradient signals (green arrows): a geometric gradient from a weak depth validator and a global gradient from a frozen foundation model guidance.

3.2. Implicit Geometric Emergence Principle

To address the trilemma of inference latency, external dependency, and representation misalignment, we propose the Implicit Geometric Emergence Principle. The core idea is to reframe 3D perception as an emergent property derived from privileged supervision rather than external inputs.

3.2.1. Geometric Emergence Learning

Building on the previous analysis, our objective is to obtain a 3D-aware representation that requires only RGB inputs at inference, yet retains sufficient geometric information for downstream reasoning. In 3D-IDE, we treat 3D awareness as an emergent property of the encoder features themselves. Formally, within our unified representation learning framework, we redefine the 3D-aware feature as:

$$F_t^{3D} \equiv F_t, \tag{4}$$

so that a single feature space must encode both 2D semantics and 3D geometry, rather than being an explicit concatenation or addition of separate 2D and 3D streams.

To induce such an implicit 3D representation, we regard geometry as privileged supervision that is only available during training. Let Y^{3D} denote latent 3D scene variables associated with the input sequence, such as depth, local surface structure, and multi-view consistency. Rather than providing Y^{3D} (or its substitutes) as explicit input to the MLLM, we encourage the encoder to learn features from which Y^{3D} can be recovered by low-capacity decoders attached only during training. This encourages the encoder to internalize geometric information while keeping the inference-time interface unmodified. Concretely, 3D-IDE attaches a lightweight, training-only privileged module to the encoder. Given the sequence of tokens F_t, an Auxiliary Geometric Validator f_P, implemented as a compact DPT-style decoder, predicts a dense depth map $\hat{D}_t$ and a pixel-wise uncertainty map $\hat{\Sigma}_t$ for each frame. A geometric supervision signal, denoted by $\mathcal{L}_{\text{geometry}}$, measures the discrepancy between $(\hat{D}_t, \hat{\Sigma}_t)$ and the available ground-truth depth, and is designed to encourage fine-grained and uncertainty-aware depth prediction.

The video setting provides additional structure through multi-view geometry. We exploit this structure in a localized and a global manner. At the local level, a cross-view term, denoted $\mathcal{L}_{\text{cross-view}}$, enforces consistency between depth predictions across neighboring frames under the known relative pose by warping and comparing outputs. This encourages the model to learn viewpoint-consistent geometry from sparsely sampled frame pairs, without incurring the cost of exhaustive pairwise supervision. However, enforcing cross-view constraints densely over all frame pairs is computationally prohibitive for long video sequences. To extend consistency beyond the sampled pairs in a lightweight manner, we therefore introduce a complementary global constraint, denoted $\mathcal{L}_{\text{global}}$. Concep-

tually, this constraint provides a compact scene-level regularization signal: it encourages the model to produce a global representation of the scene that is geometrically coherent, thereby propagating consistency across the entire sequence while maintaining low computational overhead.

3.2.2. Geometric Validator Capacity and Initialization

IGEP is designed so that most of the 3D reasoning capacity resides in the shared visual encoder. This makes the Auxiliary Geometric Validator f_P conceptually a readout module rather than a task-specific 3D expert. In 3D-IDE we therefore instantiate f_P as a low-capacity decoder trained from scratch, instead of a powerful pre-trained depth network.

Let $\mathcal{P}_{\text{weak}}$ denote a family of lightweight validators parameterized by θ_P. Given an encoder $f_E(\cdot;\theta_E)$, the auxiliary geometric supervision can be abstracted as

$$\min_{\theta_E,\theta_P} \mathcal{L}_{\text{geometry}}(f_P(f_E(I;\theta_E);\theta_P), D^{\text{gt}}),\ f_P \in \mathcal{P}_{\text{weak}}$$

Because f_P is restricted to $\mathcal{P}_{\text{weak}}$ and has no pre-training, it cannot absorb arbitrarily complex 3D reasoning. The geometric loss instead constrains the joint system (f_E, f_P) and biases the optimization toward encoder representations $F_t = f_E(I_t;\theta_E)$ in which geometry is organized in a form that such a simple validator can decode. In contrast, a high-capacity, pre-trained validator would carry strong task-specific priors and could account for much of the 3D reasoning on its own, weakening the pressure on the encoder. From the perspective of IGEP, we therefore favor a weak, from-scratch validator that allocates capacity and prior knowledge primarily to the shared encoder.

Our ablation in Tab. 3 compares two instantiations of f_P that share the same architecture but differ in initialization: a pre-trained depth model from vggt and a from-scratch variant. The two settings achieve very similar performance across ScanRefer and Multi3DRef, with the from-scratch variant being slightly but consistently higher. This suggests that a strong pre-trained depth head is not necessary for IGEP to be effective, and that a weak, from-scratch validator is sufficient while better matching the desired design philosophy of keeping geometry inside the shared encoder.

3.3. Architecture and Training Objective

Our implementation adapts a standard video-language architecture to incorporate our privileged learning framework. The core idea is a training-only module that enables the model to develop an implicit perception of 3D geometry. This module is entirely detached and discarded during inference, thus incurring zero latency overhead. The training is driven by a composite objective that primarily focuses on learning fine-grained geometry through our IGEP, while localized and global signals enforce multi-view consistency.

3D-Aware Visual Encoder. We employ a pretrained Vision Transformer backbone, SigLIP [37], which is fine-tuned end-to-end. Given a sequence of N video frames $\{I_t\}_{t=1}^{N}$, the encoder produces per-frame token features

$$F_t = f_E(I_t;\theta_E) \in \mathbb{R}^{M\times C}, \quad t = 1,\dots,N,$$

where M is the number of visual tokens and C is the channel dimension. As defined in Eq. (4), these tokens are directly treated as the 3D-aware features F_t^{3D}. After a linear projection into the language embedding space, the sequence $\{F_t^{3D}\}_{t=1}^{N}$ is used as a soft visual prompt.

Auxiliary Geometric Validator. To expose the encoder to fine-grained geometric supervision, we attach an Auxiliary Geometric Validator f_P to the visual tokens. For each frame t, the validator takes F_t as input and predicts a dense depth map and a pixel-wise uncertainty map,

$$(\hat{D}_t, \hat{\Sigma}_{D,t}) = f_P(F_t;\theta_P),$$

where $\hat{D}_t \in \mathbb{R}^{H\times W}$ and $\hat{\Sigma}_{D,t} \in \mathbb{R}^{H\times W}$ are aligned with the image resolution. The validator is implemented as a DPT-style decoder with limited capacity (train from strath). It is used only during training and removed at inference.

Video MLLM. We adopt a LLaVA-Next-Video [48] style architecture with Qwen2-7B [39] as the language backbone. The projected visual tokens derived from $\{F_t^{3D}\}$ are concatenated with textual tokens and fed into the LLM as a soft visual prompt. The language model remains standard: given a tokenized instruction ψ and target token sequence $\{y_n\}_{n=1}^{T}$, it is trained autoregressively with a cross-entropy objective, as detailed in Eq. (3).

Geometric Objective. Let D_t^{gt} denote the ground-truth depth map for frame t, and let Ω be the set of valid pixels across all frames. For each valid pixel $p \in \Omega$, we define a per-pixel loss ℓ_p that combines data fidelity, gradient consistency, and uncertainty regularization:

$$\begin{aligned}\ell_p =& \|\hat{\Sigma}_{D,p} \odot (\hat{D}_p - D_p^{\text{gt}})\| \\ &+ \|\hat{\Sigma}_{D,p} \odot (\nabla\hat{D}_p - \nabla D_p^{\text{gt}})\| \\ &- \alpha \log \hat{\Sigma}_{D,p}\end{aligned} \tag{5}$$

where ∇ denotes the spatial gradient operator, $\odot$ is the element-wise product, and α is a weighting hyperparameter. The geometric loss is the average of this per-pixel loss:

$$\mathcal{L}_{\text{geometry}} = \frac{1}{|\Omega|}\sum_{p\in\Omega} \ell_p \tag{6}$$

This loss encourages accurate and uncertainty-aware depth reconstruction and compels the encoder features F_t to carry fine-grained geometric information.

Localized Cross-View Consistency. To exploit temporal structure, we introduce a cross-view loss that enforces depth consistency between neighboring frames. For a reference frame t, we randomly sample a neighboring frame t' and use the known relative pose $P(t' \to t)$ to warp the predicted depth $\hat{D}_{t'}$ into the viewpoint of frame t, obtaining $\hat{D}_{t'\to t}$. Let $M_{t'\to t} \in \{0,1\}^{H\times W}$ be a mask indicating valid warps, and let $\Omega_{t'\to t}$ be the set of pixels where $M_{t'\to t,p} = 1$. The cross-view consistency loss is:

$$\mathcal{L}_{\text{cross-view}} = \frac{1}{|\Omega_{t'\to t}|} \sum_{p\in\Omega_{t'\to t}} \left\| \hat{D}_{t,p} - \hat{D}_{t'\to t,p} \right\|_1 \quad (7)$$

This encourages depth predictions to be stable under viewpoint changes and respect multi-view constraints, as [45].

Global Scene-Level Consistency. The global regularizer aligns the encoder's sequence-level representation with that of the frozen 3D foundation model f_G [38]. Given the teacher descriptor f_a and the projected encoder descriptor f_b defined above, we use a cosine-distance loss:

$$\mathcal{L}_{\text{global}} = 1 - \cos(f_a, f_b) = 1 - \frac{f_a^\top f_b}{\|f_a\|_2 \, \|f_b\|_2} \quad (8)$$

Because f_G is frozen and used only at training time, this loss provides lightweight scene-level guidance without introducing any additional inference latency.

Composite Training Objective. The total training objective is a sum of the primary $\mathcal{L}_{\text{ce}}$ and our auxiliary losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ce}} + \mathcal{L}_{\text{geometry}} + \mathcal{L}_{\text{cross-view}} + \mathcal{L}_{\text{global}} \quad (9)$$

At inference time, the validator f_P and the 3D foundation model f_G are removed, and the MLLM performs 3D-aware reasoning using only the unified encoder features $\{F_t^{3D}\}$ extracted from RGB video.

4. Experiments

In this section, we empirically evaluate 3D-IDE on standard 3D scene understanding benchmarks and analyze how each component of the framework contributes to performance. Throughout, we focus on settings where models receive only RGB inputs at inference, so that any gains stem from the learned 3D-aware representation rather than additional geometric inputs. Unless otherwise specified, our core baseline is a reproduction of Video-3D LLM [54] with its explicit geometric injection mechanism removed; we denote this baseline as *Video-3D LLM** in all tables. We first compare 3D-IDE with task-specific specialist models and 3D generalist MLLMs to assess its performance under fair, depth-free conditions and its competitiveness with methods

Figure 4. Qualitative results on three 3D vision-language tasks.

that rely on explicit 3D inputs. We then analyze the learned representation and efficiency of 3D-IDE through geometric correspondence, surface normal estimation, and inference-time measurements. Finally, we conduct ablation studies that examine the impact of geometric supervision, multi-view consistency, and validator initialization, and we provide qualitative visualizations that illustrate the emergent 3D understanding of our model.

4.1. Datasets and Evaluation Metrics

Datasets. We evaluate 3D-IDE on five standard 3D vision–language benchmarks, all derived from the ScanNet RGB-D dataset [14]. For 3D visual grounding, we use ScanRefer [5] for single-target localization and Multi3DRefer [47] for multi-target localization. For 3D dense captioning, we adopt Scan2Cap [13], which provides region-level descriptions for objects in ScanNet scenes. For 3D question answering, we use ScanQA [2], which requires reasoning about spatial relations in indoor environments. To further probe the learned geometric representation, we use

Table 2. Performance comparison on 3D scene understanding benchmarks. Specialists are single-task methods, while generalists are trained for multiple tasks. Bold indicates the best result. Our method belongs to the generalist group without 3D geometric.

Method	ScanRefer		Multi3DRefer		Scan2Cap		ScanQA		SQA3D
	Acc@0.25	Acc@0.5	F1@0.25	F1@0.5	C@0.5	B-4@0.5	C	EM	EM
Specialists									
ScanRefer [5]	37.3	24.3	–	–	–	–	–	–	–
MVT [23]	40.8	33.3	–	–	–	–	–	–	–
3DVG-Trans [50]	45.9	34.5	–	–	–	–	–	–	–
ViL3DRel [8]	47.9	37.7	–	–	–	–	–	–	–
M3DRef-CLIP [47]	51.9	44.7	42.8	–	38.4	–	–	–	–
Scan2Cap [13]	–	–	–	–	35.2	22.4	–	–	–
ScanQA [2]	–	–	–	–	–	–	64.9	21.1	47.2
3D-VisTA [57]	50.6	45.8	–	–	66.9	34.0	69.6	22.4	48.5
Generalists *(with 3D geometric inputs)*									
3D-LLM(Flamingo) [20]	21.2	–	–	–	–	–	59.2	20.4	–
3D-LLM(BLIP2-flant5) [20]	30.3	–	–	–	–	–	69.4	20.5	–
Chat-3D [41]	–	–	–	–	–	–	53.2	–	–
Chat-3D v2 [21]	42.5	38.4	45.1	41.6	63.9	31.8	87.6	–	54.7
LL3DA [10]	–	–	–	–	62.9	36.0	76.8	–	–
SceneLLM [18]	–	–	–	–	–	–	80.0	27.2	53.6
LEO [22]	–	–	–	–	72.4	38.2	101.4	21.5	50.0
Grounded 3D-LLM [11]	47.9	44.1	45.2	40.6	70.6	35.5	72.7	–	–
PQ3D [58]	57.0	51.2	–	50.1	80.3	36.0	–	–	47.1
ChatScene [21]	55.5	50.2	57.1	52.4	77.1	36.3	87.7	21.6	54.6
Inst3D-LLM [44]	57.8	51.6	58.3	53.5	79.7	38.3	88.6	24.6	–
3D-LLaVA [15]	51.2	40.6	–	–	78.8	36.9	92.6	–	54.5
LLaVA-3D [56]	54.1	42.4	–	–	79.2	41.1	91.7	27.0	55.6
Video-3D LLM [51]	58.1	51.7	58.0	52.7	83.8	41.3	102.1	30.1	58.6
3DRS[24]	62.9	56.1	60.4	54.9	86.1	41.6	104.8	30.3	60.6
Generalists *(without 3D geometric inputs)*									
*Video-3D LLM** [51]	53.7	47.8	46.0	42.4	31.5	29.9	99.7	29.5	58.6
VG LLM-4B [53]	53.5	47.5	–	–	78.6	40.9	–	–	57.0
VG LLM-8B [53]	57.6	50.9	–	–	80.0	**41.5**	–	–	57.9
VID-LLM [7]	50.1	46.7	47.2	42.9	**81.5**	40.6	101.9	27.6	57.3
ours	**60.9**	**54.5**	**59.8**	**54.9**	79.0	40.7	**102.1**	**29.8**	**59.2**

NAVI [25] for two low-level tasks: geometric correspondence and surface normal estimation.

Evaluation metrics. We follow the standard protocols for each benchmark. For 3D visual grounding, we report Acc@0.25 and Acc@0.5 on ScanRefer, and F1@0.25 and F1@0.5 on Multi3DRefer, all based on 3D IoU thresholds of 0.25 and 0.5. For 3D dense captioning (Scan2Cap), we report CIDEr and BLEU-4 on predictions whose proposal boxes have IoU ≥ 0.5 with the ground-truth region, denoted C@0.5 and B-4@0.5. For 3D question answering, we report Exact Match (EM) accuracy and CIDEr on ScanQA, and EM on SQA3D. For representation evaluation on NAVI, we follow the Probe3D protocol [17]: for surface normal estimation we report angular RMSE (lower is better) and mean accuracy (mAcc; higher is better) aggregated over standard angular error thresholds (11.25°, 22.5°, 30°), and for geometric correspondence we report correspondence recall at a 3D point error threshold of 2 cm (with full recall–threshold curves in the appendix).

Implementation Details. Our model, 3D-IDE, is trained end-to-end on 8 NVIDIA H100 GPUs with a batch size of 16. The training process takes around 23 hours for 1 epoch. We use the Adam optimizer [26] with an initial learning rate of $2e-6$ for the validator while keeping the learning rates of other modules consistent with the baseline [54], using a cosine decay schedule that decays the learning rate down to zero. Following Video-3D LLM, we uniformly sample 32 frames from each scene for both training and evaluation.

4.2. Main results

As summarized in Tab. 2, we compare 3D-IDE with task-specific specialist models that are individually fine-tuned for each benchmark and 3D generalist models that handle

diverse multiple tasks within a single framework. Among generalist models, we further distinguish those that require explicit 3D geometric inputs at inference from those that operate purely on RGB inputs. In the RGB-only generalist setting, 3D-IDE achieves the strongest overall results across all three task categories: it sets new state-of-the-art performance on all 3D grounding and QA metrics, and remains highly competitive on Scan2Cap, trailing the best captioning models by less than 3 CIDEr and 1 BLEU-4. At the same time, it is significantly more efficient than representative 3D generalist baselines that incorporate explicit geometry: compared to VG-LLM-8B [53], 3D-IDE reduces the parameter count by 12.86% and inference latency by 55.3% while maintaining comparable accuracy, as reported in Tab. 5. Moreover, 3D-IDE attains performance comparable to other generalist models that use ground-truth geometric inputs during inference, despite relying only on RGB video. Qualitative results on ScanRefer, Scan2Cap, and ScanQA further illustrate these gains and show that 3D-IDE produces both accurate localizations and descriptions that respect the underlying 3D scene context, as shown in Fig. 4

We further evaluate geometric awareness via geometric correspondence and surface normal estimation, as reported in Tab. 4. Compared to the RGB-only baseline without IGEP, 3D-IDE achieves higher correspondence recall under both overall and large-baseline settings, and improves both normal RMSE and accuracy. These results indicate that IGEP leads to stronger 3D-aware encoder features, which in turn improve performance on 3D vision–language tasks.

4.3. Ablation Study

We conduct an ablation study on ScanRefer and Multi3DRef to quantify the contribution of each IGEP component, as summarized in Tab. 3. Starting from an RGB-only baseline without auxiliary losses, adding only the global loss $\mathcal{L}_{\text{global}}$ brings modest but consistent gains on both datasets, indicating that aligning the encoder's sequence-level representation with a frozen 3D foundation model provides useful scene-level regularization. Introducing the geometric loss $\mathcal{L}_{\text{geometry}}$ further improves performance: with $\mathcal{L}_{\text{global}}$ fixed, variants with geometric supervision outperform the baseline by several points on both benchmarks. Comparing the two validator initializations, we observe that the from-scratch validator achieves performance that is very close to, and slightly higher than, its pre-trained counterpart across all reported metrics, showing that IGEP does not rely on a strong pre-trained depth head and that a weak, from-scratch validator is sufficient.

Enabling the cross-view loss $\mathcal{L}_{\text{cross-view}}$ on top of the global and geometric losses yields the best overall configuration, with monotonic improvements across all grounding metrics on both datasets. This demonstrates that localized multi-view consistency complements fine-grained geometric supervision and global sequence-level alignment. Overall, the ablations confirm that each component of IGEP contributes positively and that combining all of them produces the strongest 3D-aware encoder for downstream 3D vision–language tasks. Notably, the full IGEP configuration even surpasses our baseline that relies on Explicit 3D-Input, while using only RGB inputs at inference time.

Table 3. **Ablation Study.** Effect of the components in our model. ✓ denotes the component is enabled, × denotes it is disabled.

Components			ScanRefer		Multi3DRef	
Global.	Geometric.	Cross-view.	F1@0.25	F1@0.5	Acc@0.25	Acc@0.5
×	×	×	53.7	47.8	46.0	42.4
✓	×	×	56.9	50.8	55.6	51.1
✓	✓ $\text{pretrain}_{\text{vggt}}$	×	59.6	53.2	58.7	53.2
✓	✓ scratch	×	59.8	53.3	59.7	54.3
✓	✓ $\text{pretrain}_{\text{vggt}}$	✓	**60.5**	**54.1**	**59.7**	**54.5**
✓	✓ scratch	✓	**60.9**	**54.5**	**59.8**	**54.9**

Table 4. **Geometric Representation Evaluation.** We compare our method against the baseline on the tasks of Spatial Correspondences and Normal Prediction. Our method demonstrates superior performance by consistently outperforming the baseline across all four metrics. Detailed results are available in appendix.

Method	Spatial Correspondences		Normal Prediction	
	Recall@2cm ↑	θ_{90}^{120} ↑	RMSE ↓	mAcc↑
Baseline [54]	40.15	21.38	32.26	52.13
Baseline + ours	**42.27**	**23.06**	**31.36**	**53.49**

Table 5. **Inference Efficiency Comparison.** Our model achieves over 2× faster inference and higher generation throughput while using less GPU memory than VG-LLM under identical settings.

Method	Params(B)↓	Mean Time (s)↓	Tokens/s ↑	Peak Mem↓
VG LLM-8B [53]	9.25	3.60	4.32	21.10 GB
ours	**8.06**	**1.61**	**10.72**	**18.35** GB

5. Conclusion

We presented 3D-IDE, a framework that addresses the inference-time latency and data dependencies of 3D-aware MLLMs. Guided by the IGEP, a lightweight training-only validator encourages a unified RGB encoder to internalize 3D structure from video. As a result, 3D-IDE matches or surpasses state-of-the-art performance while requiring no 3D inputs or auxiliary encoders at inference and incurring no latency overhead. This shows that 3D knowledge can be learned implicitly within a single encoder, enabling more practical 3D understanding models.

Acknowledgement: YW is supported by the NSFC (No. 62576043). HL is supported in part by an ARC Grant (DP220100800). HL holds concurrent appointments with both ANU and Amazon; however, this research was conducted at ANU and is independent of Amazon.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1, 2

[2] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *CVPR*, 2022. License: Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. 6, 7, 2, 5

[3] Daigang Cai, Lichen Zhao, Jing Zhang, Lu Sheng, and Dong Xu. 3djcg: A unified framework for joint dense captioning and visual grounding on 3d point clouds. In *CVPR*, 2022. 4, 5

[4] Ziyun Cai, Jungong Han, Li Liu, and Ling Shao. Rgb-d datasets using microsoft kinect or similar sensors: a survey. *Multimedia Tools and Applications*, 76(3):4313–4355, 2017. 1

[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*, 2020. License: Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. 6, 7, 2, 4

[6] Dave Zhenyu Chen, Qirui Wu, Matthias Nießner, and Angel X. Chang. D^3net: A unified speaker-listener architecture for 3d dense captioning and visual grounding. In *ECCV*, 2022. 4, 5

[7] Haijier Chen, Bo Xu, Shoujian Zhang, Haoze Liu, Jiaxuan Lin, and Jingrong Wang. Vid-llm: A compact video-based 3d multimodal llm with reconstruction-reasoning synergy. *arXiv preprint arXiv:2509.24385*, 2025. 1, 2, 7

[8] Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Language conditioned spatial relation reasoning for 3d object grounding. In *NeurIPS*, 2022. 7, 4

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

[10] Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. Ll3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26428–26438, 2024. 1, 7, 5

[11] Yilun Chen, Shuai Yang, Haifeng Huang, Tai Wang, Ruiyuan Lyu, Runsen Xu, Dahua Lin, and Jiangmiao Pang. Grounded 3d-llm with referent tokens. 2024. 7, 4, 5

[12] Yilun Chen, Shuai Yang, Haifeng Huang, Tai Wang, Runsen Xu, Ruiyuan Lyu, Dahua Lin, and Jiangmiao Pang. Grounded 3d-llm with referent tokens. *arXiv preprint arXiv:2405.10370*, 2024. 2

[13] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *CVPR*, 2021. License: Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. 6, 7, 3

[14] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. License: ScanNet Terms of Use. 6

[15] Jiajun Deng, Tianyu He, Li Jiang, Tianyu Wang, Feras Dayoub, and Ian Reid. 3d-llava: Towards generalist 3d lmms with omni superpoint transformer. 2025. 7

[16] Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21795–21806, 2024. 2

[17] Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. In *CVPR*, 2024. 7, 1

[18] Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual understanding and reasoning. 2024. 7, 3, 5

[19] 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. 1, 2

[20] 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. In *NeurIPS*, 2023. 7, 4, 5

[21] 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. 2023. 7, 3, 4, 5

[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. 2024. 7, 5

[23] Shijia Huang, Yilun Chen, Jiaya Jia, and Liwei Wang. Multi-view transformer for 3d visual grounding. In *CVPR*, 2022. 7, 4

[24] Xiaohu Huang, Jingjing Wu, Qunyi Xie, and Kai Han. Mllms need 3d-aware representation supervision for scene understanding. *arXiv preprint arXiv:2506.01946*, 2025. 1, 2, 3, 7, 4, 5

[25] Varun Jampani, Kevis-Kokitsi Maninis, Andreas Engelhardt, Arjun Karpur, Karen Truong, Kyle Sargent, Stefan Popov,

André Araujo, Ricardo Martin Brualla, Kaushal Patel, et al. Navi: Category-agnostic image collections with high-quality 3d shape and pose annotations. *Advances in Neural Information Processing Systems*, 36:76061–76084, 2023. 7

[26] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[27] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 2

[28] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 1, 2

[29] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024. 1

[30] Zuyan Liu, Yuhao Dong, Ziwei Liu, Winston Hu, Jiwen Lu, and Yongming Rao. Oryx mllm: On-demand spatial-temporal understanding at arbitrary resolution. *arXiv preprint arXiv:2409.12961*, 2024. 1

[31] Zuyan Liu, Yuhao Dong, Ziwei Liu, Winston Hu, Jiwen Lu, and Yongming Rao. Oryx MLLM: on-demand spatial-temporal understanding at arbitrary resolution. 2024. 5

[32] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. SQA3D: situated question answering in 3d scenes. In *ICLR*, 2023. License: CC-BY-4.0. 3

[33] Yunze Man, Shuhong Zheng, Zhipeng Bao, Martial Hebert, Liangyan Gui, and Yu-Xiong Wang. Lexicon3d: Probing visual foundation models for complex 3d scene understanding. *Advances in Neural Information Processing Systems*, 37:76819–76847, 2024. 2

[34] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[35] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2

[36] 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. 2

[37] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 5

[38] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 1, 2, 3, 6

[39] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 2, 5

[40] Zehan Wang, Haifeng Huang, Yang Zhao, Ziang Zhang, and Zhou Zhao. Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes. *arXiv preprint arXiv:2308.08769*, 2023. 2

[41] Zehan Wang, Haifeng Huang, Yang Zhao, Ziang Zhang, and Zhou Zhao. Chat-3d: Data-efficiently tuning large language model for universal dialogue of 3d scenes. 2023. 7, 5

[42] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision*, pages 131–147. Springer, 2024. 2

[43] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision*, pages 131–147. Springer, 2024. 1

[44] Hanxun Yu, Wentong Li, Song Wang, Junbo Chen, and Jianke Zhu. Inst3d-lmm: Instance-aware 3d scene understanding with multi-modal instruction tuning. 2025. 7

[45] Chushan Zhang, Jinguang Tong, Tao Jun Lin, Chuong Nguyen, and Hongdong Li. Pmvc: Promoting multi-view consistency for 3d scene reconstruction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3678–3688, 2024. 6

[46] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. 2025. 1, 2

[47] Yiming Zhang, ZeMing Gong, and Angel X. Chang. Multi3drefer: Grounding text description to multiple 3d objects. In *ICCV*, 2023. License: MIT. 6, 7, 2, 4, 5

[48] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024. 3, 5

[49] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data, 2024. 5

[50] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3dvg-transformer: Relation modeling for visual grounding on point clouds. In *ICCV*, 2021. 7, 4

[51] Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d llm: Learning position-aware video representation for 3d scene understanding. 2024. 7, 1, 3, 4, 5

[52] Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. Towards learning a generalist model for embodied navigation. In *CVPR*, 2024. 5

[53] Duo Zheng, Shijia Huang, Yanyang Li, and Liwei Wang. Learning from videos for 3d world: Enhancing mllms with 3d vision geometry priors. *arXiv preprint arXiv:2505.24625*, 2025. 1, 2, 7, 8, 3

[54] 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*, pages 8995–9006, 2025. 1, 2, 3, 6, 7, 8, 4

[55] Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. *arXiv preprint arXiv:2409.18125*, 2024. 1, 2

[56] Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d-awareness. 2024. 7, 3, 4, 5

[57] 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 *ICCV*, 2023. 7, 3, 4, 5

[58] 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 *ECCV*, 2024. 7, 4, 5

3D-IDE: 3D Implicit Depth Emergent

Supplementary Material

6. Datasets Statistics

Training data. For fine-tuning, we adopt the same pool of 3D Understanding and Reasoning benchmarks as Video-3D LLM [51], namely ScanRefer, Multi3DRefer, Scan2Cap, ScanQA, and SQA3D. In total, this yields 223,128 training examples: SQA3D provides 79,445 samples (35.6% of the corpus), Multi3DRefer 43,838 (19.6%), ScanRefer and Scan2Cap 36,665 each (16.4% per dataset), and ScanQA 26,515 (11.9%). All datasets except SQA3D are built on 562 reconstructed scans, while SQA3D covers 518 scans. The average question length ranges from 13 to 38 words across datasets. Scan2Cap and ScanQA additionally offer answer sentences averaging 17.9 and 2.4 words, and SQA3D has relatively long questions (37.8 words on average) with very short answers (1.1 words).

Evaluation data. For evaluation, we use the official validation splits of ScanRefer, Multi3DRefer, Scan2Cap, and ScanQA, together with the test split of SQA3D. The combined evaluation suite contains 30,890 instances: 11,120 from Multi3DRefer (36.0%), 9,508 from ScanRefer (30.8%), 4,675 from ScanQA (15.1%), 3,519 from SQA3D (11.4%), and 2,068 from Scan2Cap (6.7%). The average question length in these splits varies between 13.0 and 36.3 words, while Scan2Cap and ScanQA provide answer texts with mean lengths of 18.7 and 2.4 words, respectively; SQA3D again features long questions (36.3 words) with very short answers (1.1 words).

7. Additional Ablative Analysis

Geometric Representation. As shown in Tabs. 11 and 12, both the pretrained-head and from-scratch-head variants improve encoder probing scores compared to training without any validator, confirming that geometric supervision is beneficial. Among them, the from-scratch validator yields the highest normal and correspondence accuracy, indicating that it encourages the encoder to internalize 3D structure more effectively than relying on a stronger pretrained head. This outcome is consistent with the design philosophy of IGEP: the geometric validator is intentionally kept weak and low-capacity so that it cannot absorb complex 3D reasoning on its own. To minimize the geometric loss, the visual encoder is instead pressured to internalize 3D structure within the shared tokens so that this low-capacity validator can decode it. Geometry therefore neither explicitly injected nor disentangled, but emerges under optimization pressure within a unified representation space. For surface normal estimation, we follow the Probe3D [17] protocol and train linear probing heads on features extracted from the frozen visual encoder. The reported geometry is thus derived entirely from the implicit RGB tokens; the training-only validator is not involved at evaluation time.

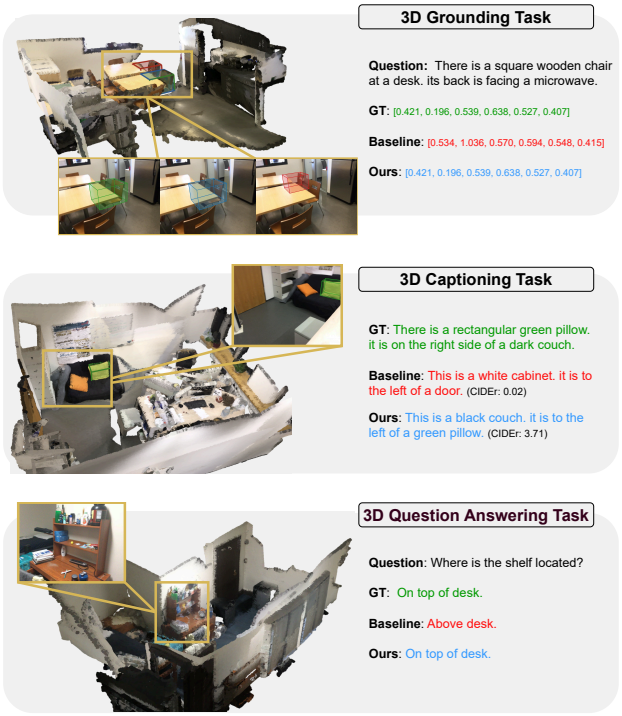


Figure 5. More qualitative results on three 3D vision-language tasks: language-guided object localization (top), region-level captioning (middle), and spatial question answering (bottom). In the grounding examples, green 3D bounding boxes denote the ground-truth targets, red boxes the predictions of the baseline, and blue boxes the predictions of our model. Our method better aligns with the targets and produces more accurate captions and answers.

Role of Global Supervision. A natural concern is that 3D-IDE might simply inherit 3D knowledge from the founda-

Table 6. **Additional Ablation Study.** Effect of $\mathcal{L}_{\text{global}}$ to speed-up training. ✓ denotes enabled, × denotes disabled.

Components			ScanRefer		Multi3DRef	
Global.	Geometric.	Cross-view.	Acc@0.25	Acc@0.5	F1@0.25	F1@0.5
×	×	×	53.7	47.8	46.0	42.4
✓	×	×	56.9	50.8	55.6	51.1
×	✓	✓	58.6	51.9	57.8	52.5
✓	✓ scratch	×	59.8	53.3	59.7	54.3
✓	✓ scratch	✓	**60.9**	**54.5**	**59.8**	**54.9**

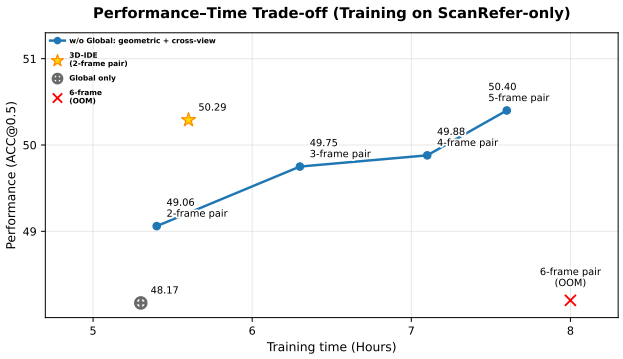


Figure 6. 3D-IDE retains performance w/o global supervision, at the cost of longer training and higher VRAM (OOM at 6-frame).

tion model f_G [38] through the global supervision. To disentangle this effect, we ablate $\mathcal{L}_{\text{global}}$ in Tab. 6. Using only local geometric and cross-view constraints, *i.e.*, $\mathcal{L}_{\text{geometry}}$ and $\mathcal{L}_{\text{cross-view}}$ with 2-frame warping, already yields a substantial gain over the baseline and even surpasses using $\mathcal{L}_{\text{global}}$ alone. This indicates that the principal source of 3D awareness arises from IGEP itself rather than being dictated by f_G. At the same time, combining local and global supervision achieves the best overall performance, while $\mathcal{L}_{\text{global}}$ is used at training and is entirely discarded at inference. In practice, the global term behaves as a training-time scene-level regularizer that approximates dense multi-view constraints whose pairwise complexity grows quadratically with sequence length, delivering an almost "free-lunch" improvement in 3D consistency without any inference latency. Importantly, f_G and the geometric validator operate on distinct parts of the architecture: f_G supervises only the final VLM hidden space via $\mathcal{L}_{\text{global}}$, whereas the geometric validator acts solely on the upstream shared encoder via $\mathcal{L}_{\text{geometry}}$ and $\mathcal{L}_{\text{cross-view}}$. This separation avoids direct coupling between the teacher and validator feature spaces, reducing the risk of misaligned supervision signals. As shown in Fig. 6, 3D-IDE retains strong performance even without global supervision, though at the cost of longer training and higher VRAM usage.

Extended Ablation Across All Benchmarks. To further validate the contribution of each component, Tab. 7 extends the ablation in the main paper to all five benchmarks by cumulatively adding each objective. Each component brings consistent improvements across tasks, and the full configuration achieves the best results on all five benchmarks.

8. Detailed Comparison

Here, we conduct a thorough comparison with other methods, covering all metrics across five benchmark tasks.

RGB-Only Inference: Impact of Removing 3D Inputs from 3DRS. A key claim of our work is that methods relying on ground-truth depth and camera pose at inference time suffer a severe performance drop when those inputs are withheld. Tab. 8 substantiates this by comparing 3DRS [24] in its original setting (with 3D geometric inputs) against its RGB-only variant (3DRS*, without 3D inputs) and our method, which is RGB-only by design. Removing 3D inputs causes a sharp drop in 3DRS performance on both ScanRefer and Multi3DRefer, whereas 3D-IDE remains strong and outperforms 3DRS* by a substantial margin on all grounding metrics. This confirms that 3DRS cannot be categorized as a generalist model without 3D geometric inputs, as it requires ground-truth depth and camera pose at inference to construct coordinate maps.

Components	ScanRefer Acc@0.25 ↑	Multi3DRefer F1@0.25 ↑	Scan2Cap CIDEr ↑	ScanQA CIDEr ↑	SQA3D EM ↑
Baseline	53.7	46.0	31.5	99.7	58.6
+ $\mathcal{L}_{\text{global}}$	56.9	55.6	77.7	100.0	57.8
+ $\mathcal{L}_{\text{geometry}}$	59.8	59.7	78.7	101.9	59.0
+ $\mathcal{L}_{\text{cross-view}}$ (Full)	**60.9**	**59.8**	**79.0**	**102.1**	**59.2**

Table 7. Extended ablation across all five benchmarks. Each row cumulatively adds one IGEP component. The full model consistently achieves the best results across all tasks.

Method	3D inputs	ScanRefer		Multi3DRefer	
		Acc@0.25	Acc@0.5	F1@0.25	F1@0.5
3DRS [24]	✓	62.9	56.1	60.4	54.9
3DRS* [24]	×	56.95	50.83	55.8	51.1
3D-IDE (Ours)	×	**60.9**	**54.5**	**59.8**	**54.9**

Table 8. Effect of removing 3D geometric inputs from 3DRS at inference. 3DRS* denotes the RGB-only variant. Our method closes most of the gap to the full 3DRS while using no 3D inputs.

ScanRefer. As shown in the detailed ScanRefer [5] results in Tab. 13, our method achieves strong overall performance, with clear improvements over the baseline on both Acc@0.25 and Acc@0.5, indicating better fine-grained localization of the target object.

Multi3DRefer. Following [47], we evaluate all question types, including zero-target (ZT), single-target (ST), and multi-target (MT) cases, with and without distractors. From Tab. 14, our approach consistently outperforms previous methods on the ST and MT splits under both distractor settings, demonstrating stronger robustness to spurious objects. Interestingly, the depth-free variants of Video 3D-LLM and 3DRS obtain higher ZT scores but substantially worse ST and MT results, indicating a bias toward predicting no target once geometric cues are removed.

ScanQA. On the ScanQA validation set [2], our method achieves better results than prior approaches on key metrics such as EM@1 and CIDEr, and is competitive on BLEU and METEOR, as shown in Tab. 15. These results highlight

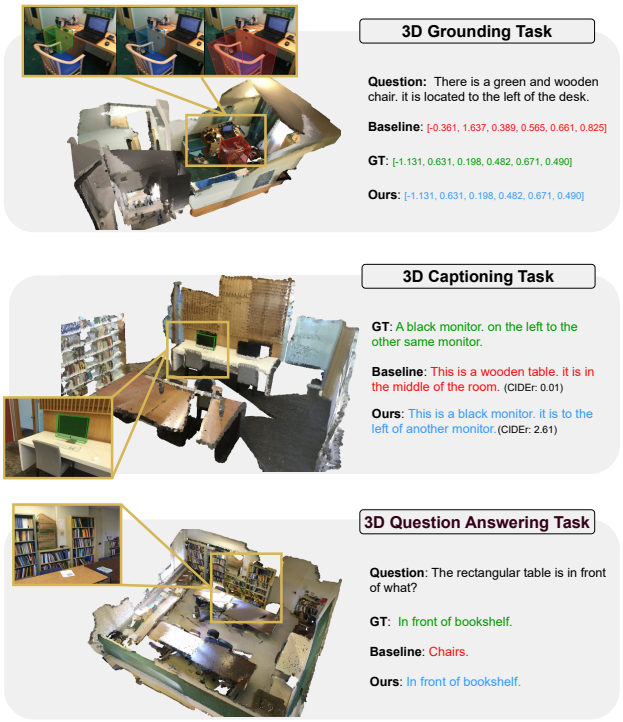


Figure 7. More qualitative results on three 3D vision-language tasks (continued). The three rows correspond to object localization, region-level captioning, and spatial question answering, respectively. Color coding is the same as in Figure 5. Our model remains effective across diverse scenes and linguistic queries.

the effectiveness of our model for 3D question answering.

SQA3D. As shown in Tab. 10, our method establishes new state-of-the-art performance on the SQA3D test split [32]. It attains the highest overall EM and consistently improves over previous methods across most question types, indicating strong generalization to diverse question categories.

Scan2Cap. On the Scan2Cap validation benchmark [13], we adopt the training and inference protocol of [54]. Under this setting, our method substantially improves over our baseline and attains CIDEr and BLEU-4 scores close to the best results, while remaining competitive on METEOR and ROUGE-L, as summarized in Tab. 9.

9. More Qualitative Results

Figs. 5 and 7 qualitatively summarize the behavior of our model on three challenging 3D scene understanding tasks: language-guided object localization, region-level captioning, and spatial question answering. In the visual grounding examples, the model must retrieve the correct object in a cluttered 3D environment given a natural-language description. For each case we visualize three bounding boxes: green denotes the ground-truth target, red the prediction of the RGB-only baseline, and blue the prediction of our model. Our predictions align much more closely with the intended targets, indicating that the model can reliably interpret both spatial and semantic cues from language. Across all three tasks, these qualitative results demonstrate that, even without any 3D input during inference, our method leverages its learned 3D-aware representation to produce more accurate and coherent outputs than the baseline.

Method	@0.5			
	C	B-4	M	R
Scan2Cap [13]	39.08	23.32	21.97	44.48
3D-VisTA [57]	66.90	34.00	27.10	54.30
ChatScene [21]	77.19	36.34	28.01	58.12
LLaVA-3D [56]	79.21	41.12	30.21	63.41
baseline [51]	31.53	29.98	24.18	57.66
VG-LLM [53]	80.00	41.50	28.90	62.60
Ours	79.02	40.76	28.79	62.13

Table 9. Performance comparison on the **Scan2Cap** validation set.

Method	Test set						**Avg.**
	What	Is	How	Can	Which	Others	
3D-VisTA [57]	34.8	63.3	45.4	69.8	47.2	48.1	48.5
Scene-LLM [18]	40.9	69.1	45.0	70.8	47.2	52.3	54.2
ChatScene [21]	45.4	67.0	52.0	69.5	49.9	55.0	54.6
LLaVA-3D [56]	–	–	–	–	–	–	55.6
Video-3D [51]	51.1	72.4	55.5	69.8	51.3	56.0	58.6
baseline [51]	51.8	73.1	56.5	70.1	51.0	54.7	58.5
Ours	**51.8**	72.7	**60.4**	68.3	49.0	**58.0**	**59.2**

Table 10. Performance comparison on the test set of **SQA3D**.

Table 11. **Correspondence Estimation Results for NAVI.** We present the NAVI correspondence estimation results for all models. The results are presented for features extracted at different layers with performance binned for different relative viewpoint changes between image pairs. The highest performing entry in each column is bolded.

		Block_0				Block_1				Block_2				Block_3			
Model	**Venue**	θ_0^{30}	θ_{30}^{60}	θ_{60}^{90}	θ_{90}^{120}	θ_0^{30}	θ_{30}^{60}	θ_{60}^{90}	θ_{90}^{120}	θ_0^{30}	θ_{30}^{60}	θ_{60}^{90}	θ_{90}^{120}	θ_0^{30}	θ_{30}^{60}	θ_{60}^{90}	θ_{90}^{120}
Video-3D LLM [54]	CVPR'25	**75.99**	**38.83**	**20.27**	10.68	80.48	52.97	32.15	17.70	75.36	49.26	34.94	21.38	71.97	46.28	34.50	22.19
3DRS [24]	NIPS'25	74.05	37.67	19.81	10.60	79.59	52.13	**33.07**	**17.87**	73.88	48.77	35.76	21.89	69.64	45.57	34.86	22.69
3D-IDE (pretrain)	-	74.63	38.07	19.99	10.69	81.60	53.64	33.03	17.33	77.04	**52.25**	37.13	22.34	72.33	47.68	35.15	22.11
3D-IDE (scratch)	–	74.93	38.05	19.89	**10.77**	**81.63**	**53.79**	33.06	17.38	**77.05**	52.02	**37.58**	**23.06**	**72.46**	**47.97**	**36.16**	**23.21**

Table 12. **Surface Normal Estimation Results.** We present the surface normal estimation results for all models. Higher is better for accuracy, lower is better for RMSE. The highest performing entry in each column is bolded.

		NAVI (Test)				
Model	**Venue**	Acc@11.25° (%)	Acc@22.5° (%)	Acc@30° (%)	RMSE (°) ↓	mAcc (%) ↑
Video-3D LLM [54]	CVPR'25	28.69	57.65	70.06	32.26	52.13
3DRS [24]	NIPS'25	28.61	57.90	70.48	31.86	52.33
3D-IDE (pretrain)	–	29.85	58.73	71.05	31.46	53.21
3D-IDE (scratch)	–	**30.15**	**59.01**	**71.32**	**31.46**	**53.49**

Table 13. **Performance comparison on the validation set of ScanRefer.** "Unique" and "Multiple" depends on whether there are other objects of the same class as the target object.

Method	Unique		Multiple		Overall	
	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5	Acc@0.25	Acc@0.5
ScanRefer [5]	76.3	53.5	32.7	21.1	41.2	27.4
MVT [23]	77.7	66.4	31.9	25.3	40.8	33.3
3DVG-Transformer [50]	81.9	60.6	39.3	28.4	47.6	34.7
ViL3DRel [8]	81.6	68.6	40.3	30.7	47.9	37.7
3DJCG [3]	83.5	64.3	41.4	30.8	49.6	37.3
D3Net [6]	–	72.0	–	30.1	–	37.9
M3DRef-CLIP [47]	85.3	77.2	43.8	36.8	51.9	44.7
3D-VisTA [57]	81.6	75.1	43.7	39.1	50.6	45.8
3D-LLM (Flamingo) [20]	–	–	–	–	21.2	–
3D-LLM (BLIP2-flant5) [20]	–	–	–	–	30.3	–
Grounded 3D-LLM [11]	–	–	–	–	47.9	44.1
PQ3D [58]	86.7	78.3	51.5	46.2	57.0	51.2
ChatScene [21]	89.6	82.5	47.8	42.9	55.5	50.2
LLaVA-3D [56]	–	–	–	–	54.1	42.2
Video-3D LLM [51]	88.0	78.3	50.9	45.3	58.1	51.7
baseline [51]	82.17	73.71	45.10	40.14	52.29	46.66
3DRS* [24]	82.76	73.50	50.74	45.37	56.95	50.83
3D-IDE (Ours)	**86.72**	**77.94**	**54.73**	**48.90**	**60.94**	**54.53**

Table 14. **Performance comparison on Multi3DRefer validation set.** ZT: zero-target, ST: single-target, MT: multi-target, D: distractor.

Method	ZT w/o D	ZT w/ D	ST w/o D		ST w/ D		MT		ALL	
	F1	F1	F1@0.25	F1@0.5	F1@0.25	F1@0.5	F1@0.25	F1@0.5	F1@0.25	F1@0.5
M3DRef-CLIP [47]	81.8	39.4	53.5	47.8	34.6	30.6	43.6	37.9	42.8	38.4
D3Net [6]	81.6	32.5	–	38.6	–	23.3	–	35.0	–	32.2
3DJCG [3]	94.1	66.9	–	26.0	–	16.7	–	26.2	–	26.6
Grounded 3D-LLM [11]	–	–	–	–	–	–	–	–	45.2	40.6
PQ3D [58]	85.4	57.7	–	68.5	–	43.6	–	40.9	–	50.1
ChatScene [21]	90.3	62.6	82.9	75.9	49.1	44.5	45.7	41.1	57.1	52.4
Video-3D LLM [51]	94.7	78.5	82.6	73.4	52.1	47.2	40.8	35.7	58.0	52.7
3DRS [24]	95.6	79.4	79.6	71.4	57.0	51.3	43.0	37.8	60.4	54.9
baseline [51]	98.7	91.5	60.5	54.9	36.9	33.8	35.9	31.6	45.9	42.3
3DRS* [24]	96.6	85.2	75.1	67.4	49.0	44.8	42.6	37.6	55.8	51.1
3D-IDE (Ours)	95.6	79.7	**79.9**	**72.6**	**54.7**	**49.7**	**45.3**	**40.5**	**59.8**	**54.9**

Table 15. **Performance comparison on the validation set of ScanQA.** EM indicates exact match accuracy, and B-1, B-2, B-3, B-4 denote BLEU-1, -2, -3, -4, respectively.

Method	EM	B-1	B-2	B-3	B-4	ROUGE-L	METEOR	CIDEr
ScanQA [2]	21.05	30.24	20.40	15.11	10.08	33.33	13.14	64.86
3D-VisTA [57]	22.40	–	–	–	10.40	35.70	13.90	69.60
Oryx-34B [31]	–	38.00	24.60	–	–	37.30	15.00	72.30
LLaVA-Video-7B [49]	–	39.71	26.57	9.33	3.09	44.62	17.72	88.70
3D-LLM (Flamingo) [20]	20.40	30.30	17.80	12.00	7.20	32.30	12.20	59.20
3D-LLM (BLIP2-flant5) [20]	20.50	39.30	25.20	18.40	12.00	35.70	14.50	69.40
Chat-3D [41]	–	29.10	–	–	6.40	28.50	11.90	53.20
NaviLLM [52]	23.00	–	–	–	12.50	38.40	15.40	75.90
LL3DA [10]	–	–	–	–	13.53	37.31	15.88	76.79
Scene-LLM [18]	27.20	43.60	26.80	19.10	12.00	40.00	16.60	80.00
LEO [22]	–	–	–	–	11.50	39.30	16.20	80.00
Grounded 3D-LLM [11]	–	–	–	–	13.40	–	–	72.70
ChatScene [21]	21.62	43.20	29.06	20.57	14.31	41.56	18.00	87.70
LLaVA-3D [56]	27.00	–	–	–	14.50	50.10	20.70	91.70
Video 3D-LLM [51]	30.10	47.05	31.70	22.83	16.17	49.02	19.84	102.0
baseline [51]	29.5	46.9	31.3	22.7	16.2	48.8	19.6	100.5
3DRS* [24]	29.7	**47.9**	32.5	**23.8**	16.9	48.3	20.2	101.3
3D-IDE (Ours)	**29.8**	47.5	**32.9**	23.7	**17.4**	**48.8**	**20.8**	**102.1**