FROM 3D REPRESENTATIONS TO OBJECT TOKENS IN OBJECT-CENTRIC MLLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models (MLLMs) provide a general interface for perception and language-based reasoning, yet it remains unclear how the underlying 3D representation affects the performance. Existing systems often couple object discovery, representation source, and subsequent object-token construction, making their individual contributions difficult to isolate. In this paper, we study these factors under a controlled object-centric formulation that separates object proposals from the representation used for downstream reasoning, and compare multiple representation sources and object-token construction mechanisms. Under our strongest configuration, the resulting model surpasses prior point-cloud-only methods on most comparable metrics across five ScanNet benchmarks. Our experiments further show that object discovery and object representation can be treated as distinct design choices, and that the choice of frozen 3D representation has a substantial effect on downstream performance, whereas increasing the complexity of the object-token construction mechanism provides only limited and task-dependent gains. These results highlight representation source as a central design factor in object-centric 3D MLLMs and show that effective object representations can be constructed with comparatively lightweight aggregation mechanisms.

1 INTRODUCTION

Multimodal large language models (MLLMs) have emerged as a general framework for integrating perceptual representations with language-based reasoning. In the 2D domain, a common paradigm is to connect a strong pretrained visual encoder to a language model through a lightweight trainable interface, allowing the language model to build on representations learned from large-scale visual data rather than learning visual semantics solely from downstream instruction supervision (Liu et al., 2023; 2024; Alayrac et al., 2022; Li et al., 2023). A corresponding question arises in 3D scene understanding: what representation should provide the perceptual evidence presented to the language model? This question is particularly relevant to object-centric 3D MLLMs, which organize a scene around explicit object proposals and expose each proposed object to the language model through an associated representation for downstream 3D understanding tasks (Huang et al., 2024a; Chen et al., 2024c; Kang et al., 2025; Zemskova & Yudin, 2025; Yu et al., 2025; Huang et al., 2026).

Reasoning about a 3D environment through an object-centric interface raises two related but distinct questions: *object discovery* and *object representation*. Object discovery determines which regions of a scene are exposed as candidate objects, whereas object representation determines what information about those regions is made available to the language model for downstream reasoning and generation. Existing systems often couple these choices within the same representation pipeline. Object-level approaches first obtain object proposals and then encode the corresponding points with dedicated object encoders (Huang et al., 2024a; Zhou et al., 2024). Detector-based approaches can instead derive proposal masks and object representations from shared detector or query features (Chen et al., 2024c; Huang et al., 2026). Other approaches further incorporate scene context, object relations, or multimodal features into the resulting representations (Huang et al., 2024b; Kang et al., 2025; Yu et al., 2025). Consequently, object discovery, representation source, and subsequent token construction often vary together, making their individual contributions difficult to isolate.

Recent progress in pretrained point representations provides a natural opportunity to revisit this coupling. Learned from large-scale 3D data, these models expose dense features over an entire scene

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.

and have shown strong transfer across downstream 3D tasks (Wu et al., 2025; 2024; Zhang et al., 2025a; 2026). Importantly, such representations need not be tied to the model responsible for object discovery: proposal masks produced by one model can be used to select features from a separately pretrained 3D representation. This makes it possible to decouple object discovery from the representation subsequently provided to the language model. This formulation enables the underlying 3D representation and the object-token construction mechanism to be varied independently while keeping the proposed objects unchanged, thereby isolating their respective effects on downstream 3D-language performance under a common interface.

In this paper, we systematically compare multiple representation sources and object-token construction mechanisms in object-centric 3D MLLMs. With object discovery separated from object representation, we examine how representation source and object-token construction affect downstream 3D-language performance. Under our strongest configuration, the resulting point-cloud-only MLLM achieves strong performance across five 3D-language benchmarks including grounding, dense captioning, and question answering, which surpasses prior point-cloud-only methods on most comparable metrics. Our experiments further demonstrate that object discovery and object representation can be treated as distinct design choices: the same object proposals can be paired effectively with representations obtained from an independently pretrained 3D representation encoder. Across our experiments, the representation supplied to the language model has a substantial effect on downstream performance, while effective object representations can be constructed with comparatively lightweight aggregation mechanisms. Together, these results provide a systematic study of how representation source and object-token construction shape object-centric 3D MLLMs.

2 Related Work

Multimodal Large Language Models for 3D Scene Understanding. Multimodal large language models have increasingly been extended to 3D scene understanding, with the goal of supporting language-conditioned perception, reasoning, and interaction in complex environments. A central design choice in these systems is how 3D scene information is represented and presented to the language model. Existing approaches broadly differ in whether they organize this information at the scene level or around explicit objects and regions. Holistic methods encode the scene into a set of scene-level visual tokens: 3D-LLM (Hong et al., 2023) lifts multi-view image features into 3D, LL3DA (Chen et al., 2024a) aggregates features from a frozen 3D detector into learned scene queries, while Scene-LLM (Fu et al., 2025), LSceneLLM (Zhi et al., 2025), and 3D-LLaVA (Deng et al., 2025) construct compact scene representations from voxel-, point-, or superpoint-level features. Object-centric approaches instead organize scene information around explicit objects or regions. Chat-3D (Wang et al., 2023) aligns object-centric 3D representations with an LLM, while Chat-Scene (Huang et al., 2024a) associates proposal features with learnable object identifiers, providing a shared interface for downstream tasks. Other work enriches object- or region-level representations through grounded referents (Chen et al., 2024c), embodied supervision (Huang et al., 2024b), relation-aware projection (Kang et al., 2025), semantic scene graphs (Zemskova & Yudin, 2025), cross-modal spatial relations (Yu et al., 2025), and context-rich object features with grounded reasoning (Huang et al., 2026). A complementary branch of methods uses posed images or video as the primary visual interface to the language model (Zhu et al., 2025; Zheng et al., 2025; Qi et al., 2026). Our work adopts the object-identifier formulation of Chat-Scene as a common interface and studies how the underlying point representation and the mechanism that converts masked point features into an object token affect downstream 3D-language performance.

Point Representations for 3D-Language Models. The 3D representations provided to language models reflect substantially different forms of supervision and pretraining. Object-level encoders such as Uni3D (Zhou et al., 2024) learn transferable representations of individual point clouds, whereas scene-level supervised backbones such as Mask3D (Schult et al., 2023) produce dense features optimized for instance segmentation. More recently, self-supervised point foundation models have learned dense scene representations from large-scale 3D data without task-specific annotations. Sonata (Wu et al., 2025) exemplifies this direction, while subsequent models such as Concerto (Zhang et al., 2025a) and Utonia (Zhang et al., 2026) further extend it with joint 2D-3D and cross-domain pretraining, respectively. These pretrained representations are increasingly incorporated into 3D-language models. Recent approaches combine frozen point features with image- or

video-derived visual tokens (Thomas et al., 2025; Zhang et al., 2026), while others use point-based representations as the primary 3D input for scene-level spatial understanding (Mao et al., 2025; Yip et al., 2026). These studies establish pretrained point representations as a useful source of 3D information for language models. However, their role in object-centric 3D MLLMs remains less clear, where object representations are typically constructed together with a particular object encoder, detector, or multimodal processing pipeline. Our work instead examines frozen dense point representations as the primary source of object information.

3D Feature Aggregation and Projection. Beyond the choice of 3D representation, multimodal systems also differ in how perceptual features are aggregated and projected into the language-model token space. In 2D MLLMs, this interface ranges from lightweight linear (Liu et al., 2023) or MLP (Liu et al., 2024) projectors to learned-query architectures such as the Perceiver Resampler (Alayrac et al., 2022) and Q-Former (Li et al., 2023). Such learned-query modules have been adopted in 3D-language models (Hong et al., 2023; Chen et al., 2024a), whereas some systems couple feature aggregation with spatial, contextual, or relational processing (Huang et al., 2024b; Kang et al., 2025; Yu et al., 2025). Other approaches employ comparatively lightweight mappings between 3D features and language-model tokens (Huang et al., 2024a; Zhu et al., 2025). LLaVA-3D (Zhu et al., 2025) further studies both query-based and pooling-based alternatives within a shared 3D feature pipeline, providing a closely related precedent for examining the complexity of this interface. Our work considers this aggregation and projection stage together with the underlying 3D representation, with particular emphasis on how much additional processing is required to form object-level representations for an object-centric MLLM.

3 Object-Centric Representation Formulation

Object-centric 3D MLLMs represent a scene through a set of object proposals and representations associated with those proposals. These correspond to two distinct components of the object-centric interface: *object discovery*, which determines which scene regions are treated as candidate objects, and *object representation*, which determines what information about those objects is made available to the language model. Existing approaches couple these components in different ways. Detector- or query-based pipelines can derive object proposals and their representations jointly from shared features (Chen et al., 2024c; Huang et al., 2026), whereas proposal-first approaches obtain candidate objects before encoding the corresponding point sets with a separate object-level representation (Huang et al., 2024a; Zhou et al., 2024). Consequently, differences in object discovery and object representation can vary together across systems, making it difficult to isolate the role of the representation ultimately provided to the language model. This motivates treating object proposals as an independent input to the representation pipeline, so that the representation can be studied without being tied to a particular object-discovery mechanism.

3.1 Object-Centric Representation Components

The representation question can be isolated by conditioning on a given set of object proposals while obtaining the underlying scene representation from an independently pretrained 3D backbone. Recent progress in large-scale point pretraining has made this setting particularly relevant (Wu et al., 2024; 2025; Zhang et al., 2025a; 2026): the resulting dense scene representations encode transferable 3D information without being tied to the detector or proposal mechanism used downstream. Object proposals can therefore specify the regions to be represented, while the information used to represent those regions is supplied by a separate pretrained representation.

The object-centric representation formulation comprises three components:

$$\underbrace{\mathcal{M}}_{\text{object proposals}}, \qquad \underbrace{\phi}_{\text{3D representation encoder}}, \qquad \underbrace{g_\theta}_{\text{object-token construction}}.$$

The proposal set $\mathcal{M}$ specifies *which* scene regions correspond to candidate objects, the representation encoder ϕ determines *what information* is encoded at the scene points, and g_θ determines *how* the proposal-selected features are summarized into a fixed-dimensional object representation. These components can be varied independently, enabling the effects of representation source and object-token construction to be studied under the same object proposals.

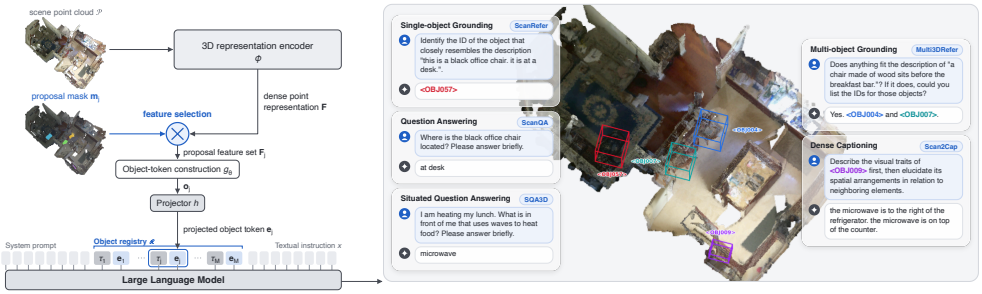

Figure 1: **Overview of the object-centric 3D MLLM. Given a 3D point cloud and a set of object proposals, a 3D representation encoder produces a dense point representation, from which proposal-selected features are converted into object representations and projected into the language-model embedding space. The resulting object tokens are organized in an object registry and provided to the language model together with a textual instruction, supporting grounding, dense captioning, and question answering through an object-centric interface.**

Object proposals $\mathcal{M}$. Let a 3D scene be represented by a point cloud $\mathcal{P} = \{\boldsymbol{p}_i\}_{i=1}^{N}$, where each point has coordinate $\boldsymbol{p}_i \in \mathbb{R}^3$ and associated attributes such as color and surface normal. Candidate objects are specified by a proposal set $\mathcal{M} = \{\boldsymbol{m}_j\}_{j=1}^{M}$, where $\boldsymbol{m}_j \in \{0,1\}^N$ is a binary mask over the scene points. Each proposal therefore specifies a candidate object through the subset of scene points selected by its mask.

3D representation encoder ϕ. A 3D representation encoder ϕ maps the complete scene to a dense point representation

$$\boldsymbol{F} = \phi(\mathcal{P}) = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_N]^\top \in \mathbb{R}^{N \times D}, \tag{1}$$

where $\boldsymbol{f}_i \in \mathbb{R}^D$ denotes the feature associated with point $\boldsymbol{p}_i$. For proposal j, the corresponding proposal feature set is $\boldsymbol{F}_j = \{\boldsymbol{f}_i \mid m_{j,i} = 1\}$. The proposal mask determines which point features belong to the proposal, whereas ϕ determines the information encoded by those features.

Object-token construction g_θ. The object-token construction function g_θ maps each variable-size proposal feature set $\boldsymbol{F}_j$ to a fixed-dimensional object representation,

$$\boldsymbol{o}_j = g_\theta(\boldsymbol{F}_j) \in \mathbb{R}^{D_o}, \qquad j = 1, \ldots, M. \tag{2}$$

The resulting object representation is subsequently projected into the language-model embedding space through the shared interface.

3.2 Language-Model Interface

The object representations constructed in Section 3.1 are projected into the language-model embedding space to form the visual tokens provided to the language model. The projector h maps each object representation as $\boldsymbol{e}_j = h(\boldsymbol{o}_j) \in \mathbb{R}^d$.

To support explicit references to individual objects, each projected representation $\boldsymbol{e}_j$ is paired with an identifier token τ_j, following prior object-identifier formulations (Huang et al., 2024a). The resulting scene representation is organized as the object registry

$$\mathcal{R} = \big[\tau_1\, \boldsymbol{e}_1,\ \tau_2\, \boldsymbol{e}_2,\ \ldots,\ \tau_M\, \boldsymbol{e}_M\big]. \tag{3}$$

Together with a textual instruction x, the registry $\mathcal{R}$ provides a shared input format for tasks requiring either explicit object references or free-form language generation.

3.3 Study Questions

The decomposition introduced above in Section 3.1 separates the source of 3D information from the mechanism used to construct object representations, which leads to two questions.

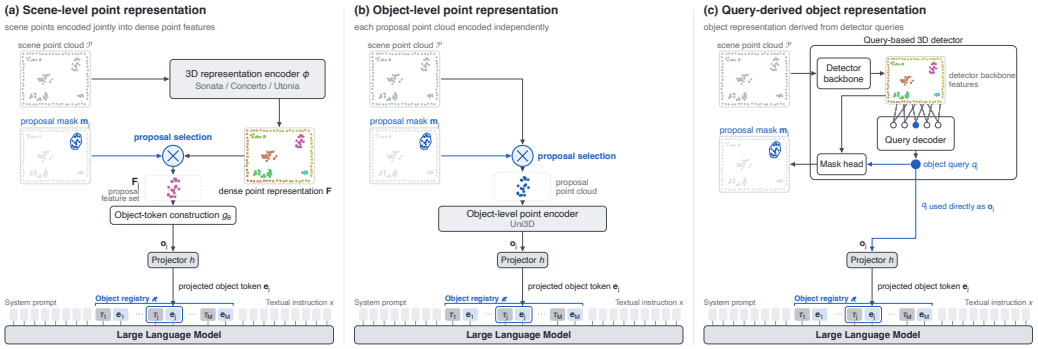

Figure 2: **Representation sources considered in this work. (a) A scene-level 3D representation encoder produces a dense point representation, from which proposal-selected features are summarized by g_θ into an object representation o_j. (b) An object-level point encoder independently encodes the point cloud associated with each proposal to produce o_j. (c) A query-based 3D detector uses the object query associated with a proposal mask directly as o_j.**

Representation source. *Given the same object proposals and object-token construction mechanism, how strongly does downstream performance depend on the underlying 3D representation?*

Object-token construction. *Given the same object proposals and 3D representation, how much additional processing is required to construct an effective object representation?*

We study the first by varying the representation source while holding $\mathcal{M}$ and g_θ fixed, and the second by varying g_θ while holding $\mathcal{M}$ and the representation source fixed. Section 4 introduces the concrete instantiations used to realize these controlled comparisons.

4 Experimental Results

Following the formulation described in Section 3, we examine the effects of object proposal source, 3D representation source, and object-token construction across grounding, dense captioning, and question-answering tasks. We first describe the experimental setup and training protocol, followed by the controlled comparisons and downstream results.

4.1 Experimental Setup

4.1.1 Object proposal sources.

The proposal set $\mathcal{M}$ is held fixed within each comparison, thereby isolating the effects of representation source and object-token construction from differences in object discovery. The primary experiments use the proposal set from Grounded 3D-LLM (Chen et al., 2024c), which is produced by a 3D detector trained with Contrastive Language-Scene Pre-training (CLASP). We use proposals generated by Mask3D (Schult et al., 2023) as an alternative proposal source in the ablation study.

4.1.2 Representation sources.

We consider three classes of representation source, as presented in Figure 2.

Dense scene representations. Within the dense scene setting, we evaluate two sources of point-wise representations: point foundation models and 3D detection/segmentation backbones. For the former, we use Sonata (Wu et al., 2025), Concerto (Zhang et al., 2025a), and Utonia (Zhang et al., 2026). For the latter, we extract dense features from the CLASP-pretrained detector backbone used by Grounded 3D-LLM (Chen et al., 2024c) and from the Mask3D (Schult et al., 2023) instance-segmentation backbone. For each representation source, the resulting features are associated with the scene points to form a dense point representation.

Object-level point representations. In the object-level setting, each proposal is encoded independently rather than as part of the complete scene. We use Uni3D (Zhou et al., 2024), following its

use for proposal-level point encoding in prior object-centric 3D MLLMs (Huang et al., 2024a). The points belonging to each proposal are extracted from the scene and encoded as a proposal point cloud, yielding a single object representation $\boldsymbol{o}_j$.

Query-derived object representations. In the query-derived setting, the object representation is obtained directly from the learned queries of a query-based 3D detector. We use the query representations from the CLASP-pretrained detector released with Grounded 3D-LLM (Chen et al., 2024c). Its query decoder operates on dense scene features and produces object queries associated with the predicted proposal masks, with the corresponding query representation used as $\boldsymbol{o}_j$.

4.1.3 Object-Token Construction

The object-token construction function g_θ maps a variable-size proposal feature set $\boldsymbol{F}_j$ to a fixed-dimensional object representation $\boldsymbol{o}_j$. We consider mean pooling, MLP–mean pooling, attention-based aggregation, and Perceiver IO (Jaegle et al., 2022), which uses a learned latent array to aggregate the proposal features into a single object representation.

4.1.4 Downstream Tasks and Evaluation

We evaluate on five ScanNet-based benchmarks spanning grounding, dense captioning, and question answering. For each task, x denotes the textual instruction and y the expected response. We summarize the task formulation and evaluation protocol for each benchmark below.

- *Single-object grounding* from ScanRefer (Chen et al., 2020): x describes a target object; y is a single object identifier. Performance is measured by the IoU between the predicted proposal box and the ground-truth box at thresholds of 0.25 and 0.5 (Acc@0.25, Acc@0.5).
- *Multi-object grounding* from Multi3DRefer (Zhang et al., 2023): x describes zero, one, or multiple target objects; y is a list of object identifiers or "No." Performance is measured by F1 at IoU thresholds of 0.25 and 0.5 (F1@0.25, F1@0.5).
- *Dense captioning* from Scan2Cap (Chen et al., 2021): x specifies an object identifier and requests a description; y is free-form text. Performance is measured by CIDEr (Vedantam et al., 2015) for proposals with IoU ≥ 0.5 to the ground-truth object.
- *Question answering* from ScanQA (Azuma et al., 2022): x is a question about the scene; y is a short free-form answer. Performance is measured by CIDEr and BLEU-4 (Papineni et al., 2002) against the reference answers.
- *Situated question answering* from SQA3D (Ma et al., 2023): x describes the situation of an agent in the scene and asks a question from the corresponding viewpoint; y is a short answer. Performance is measured by exact match (EM) and its refined variant EM-R.

Following Chat-Scene (Huang et al., 2024a), we use up to 100 object proposals for each scene. For grounding and question-answering tasks, the object registry $\mathcal{R}$ in Eq. 3 contains all $M \leq 100$ proposals, while question-answering prompts do not reveal which objects support the answer. For dense captioning, we restrict the registry to the queried proposal.

4.1.5 Training Setup

We follow a two-stage training procedure consisting of an initial alignment stage followed by downstream instruction tuning. All 3D representation sources remain frozen throughout both stages.

Stage 1: Object-token Alignment. In the first stage, the object-token construction function g_θ, the projector h, and the embeddings of the object-identifier tokens (Huang et al., 2024a) are optimized while the language model remains frozen. For object-level point and query-derived representations, which already provide a single representation per object, only the projector and object-identifier embeddings are optimized. The alignment-stage training data include region descriptions of ScanNet instances (Dai et al., 2017), dense object captions from Scan2Cap (Chen et al., 2021), and object–object distance questions that provide explicit geometric supervision.

Stage 2: Instruction Tuning. In the second stage, the object-token construction function g_θ is frozen, while the projector h and object-identifier embeddings are optimized jointly with LoRA

Table 1: **Comparison of point-cloud-based methods across five benchmarks. The methods listed in this table use only 3D point clouds. We include task-specific specialist models as references alongside generalist 3D MLLMs. The best and second-best results among comparable entries are shown in bold and underlined, respectively. † LEO's ScanQA results use prompts that identify the ground-truth answer objects and are excluded from the ranking. ‡ Chat-Scene++ results correspond to its 3D-only setting and were reported with chain-of-thought (CoT) supervision.**

Method	LLM	ScanRefer		Multi3DRefer		Scan2Cap	ScanQA		SQA3D	
		A@0.25	A@0.5	F1@0.25	F1@0.5	C@0.5	CIDEr	B-4	EM	EM-R
Specialist models										
3D-VisTA (Zhu et al., 2023)	–	50.6	45.8	–	–	66.9	72.9	13.1	48.5	–
Vote2Cap-DETR++ (Chen et al., 2024b)	–	–	–	–	–	67.6	–	–	–	–
Vote2Cap-DETR++ (SCST) (Chen et al., 2024b)	–	–	–	–	–	78.2	–	–	–	–
ConcreteNet (Unal et al., 2024)	–	50.6	46.5	–	–	–	–	–	–	–
Point-cloud-only 3D MLLMs										
LL3DA (Chen et al., 2024a)	OPT-1.3B	–	–	–	–	65.2	76.8	13.5	–	–
LEO (Huang et al., 2024b)	Vicuna-7B	–	–	–	–	72.4	101.4†	13.2†	50.0	52.4
Grounded 3D-LLM (Chen et al., 2024c)	Tiny-Vicuna-1B	48.6	44.0	44.7	40.8	70.2	75.9	13.2	–	–
LSceneLLM (Zhi et al., 2025)	Llama2-7B	–	–	–	–	–	88.2	–	–	–
3D-LLaVA (Deng et al., 2025)	Vicuna-1.5-7B	51.2	40.6	–	–	78.8	92.6	**17.1**	54.5	56.6
Chat-Scene++ ‡ (Huang et al., 2026)	Vicuna-7B	53.0	47.3	56.4	51.6	72.8	82.5	13.1	54.2	57.3
Ours	Vicuna-7B	<u>57.2</u>	<u>51.9</u>	<u>62.0</u>	<u>57.4</u>	<u>79.6</u>	<u>93.3</u>	15.4	<u>57.4</u>	<u>60.2</u>
Ours	Qwen3.5-9B	**62.0**	**56.3**	**66.4**	**61.6**	**84.0**	**97.9**	<u>16.4</u>	**60.4**	**63.5**

Table 2: **Effect of the dense representation source. We compare dense representations from 3D detection/segmentation backbones with those from point foundation models. † denotes the dense features from its detector backbone rather than its query-derived object representation. "2D–3D" denotes pretraining using paired 2D and 3D information.**

Representation source	Pretrain Supervision	Point Dim.	ScanRefer		M3DRef	Scan2Cap	ScanQA	SQA3D
			A@0.25	A@0.5	F1@0.25	C@0.5	CIDEr	EM
3D detection/segmentation backbones								
Mask3D (Schult et al., 2023)	instance masks	832	53.0	48.0	57.5	71.7	84.4	55.3
Grounded 3D-LLM† (Chen et al., 2024c)	instance masks + text	832	52.4	47.6	56.2	69.9	85.2	54.4
Object-level descriptors								
Grounded 3D-LLM (Chen et al., 2024c)	instance masks + text	-	49.7	44.5	54.5	67.0	82.4	53.2
Uni3D (Zhou et al., 2024; Huang et al., 2024a)	image + text	-	47.0	43.3	50.7	66.6	83.6	55.0
Point foundation models								
Sonata (Wu et al., 2025)	self-supervised	1,232	55.1	49.7	59.6	74.4	90.3	56.7
Concerto-Base (Zhang et al., 2025a)	self-supervised + image	1,232	**58.0**	**52.7**	**62.3**	81.5	**95.5**	**58.0**
Concerto-Large (Zhang et al., 2025a)	self-supervised + image	1,728	57.2	51.9	62.0	79.6	93.3	57.4
Utonia (Zhang et al., 2026)	self-supervised + image	1,386	56.9	51.7	61.6	**82.6**	92.5	57.0

adapters (Hu et al., 2022) inserted into the language model. The data for instruction tuning include ScanRefer (Chen et al., 2020), Multi3DRefer (Zhang et al., 2023), Scan2Cap (Chen et al., 2021), ScanQA (Azuma et al., 2022), and SQA3D (Ma et al., 2023).

Further implementation details, including architectural hyperparameters, optimization settings, training schedules are provided in the supplementary material.

4.2 Overall Performance Across 3D-Language Tasks

Table 1 compares the resulting object-centric MLLM with prior point-cloud-based methods across five ScanNet benchmarks spanning grounding, dense captioning, and question answering. The model performs strongly across all evaluated tasks using only 3D point clouds. Replacing Vicuna-7B with Qwen3.5-9B further improves performance across all reported metrics. These results demonstrate that dense point representations provide an effective basis for diverse 3D-language tasks without relying on additional image or multi-view features. We next examine how much of this performance depends on the underlying dense 3D representation under controlled comparisons.

Table 3: **Effect of object-token construction. All configurations use the same object proposals and frozen Concerto-L representation. #Params counts the learnable parameters of g_θ only.**

Object-token construction g_θ	#Params	ScanRefer		M3DRef		Scan2Cap	ScanQA	SQA3D
		A@0.25	A@0.5	F1@0.25	F1@0.5	C@0.5	CIDEr	EM
Training-free								
Mean pooling	0	57.2	51.9	62.0	57.4	79.6	93.3	**57.4**
Max pooling	0	55.8	50.4	60.5	55.8	78.2	88.1	56.7
Learnable aggregation								
Single-query attention	1.7K	**58.1**	**52.8**	**62.1**	**57.5**	81.8	**94.8**	57.3
MLP $\rightarrow$ mean	0.89M	57.6	52.2	61.8	57.3	80.7	94.0	57.1
Perceiver IO pooling	1.94M	57.9	52.4	61.9	57.3	81.0	92.5	56.8

4.3 Effect of the 3D Representation Source

Table 2 compares the three representation sources introduced in Section 4.1.3. This comparison addresses two complementary questions: how effectively object discovery can be decoupled from the representation used for downstream reasoning, and how the choice of dense 3D representation affects downstream performance when object proposals and object-token construction are fixed.

We first examine the separation between object discovery and object representation. As shown in Table 2, we compare two representations extracted from different stages of Grounded 3D-LLM Chen et al. (2024c): dense point features from its backbone, corresponding to Figure 2(a), and the query-derived object representation associated with each proposal, corresponding to Figure 2(c). The backbone representation consistently achieves stronger performance across all reported metrics. More importantly, the same proposal masks can be paired with dense representations from independently pretrained point foundation models, which yield further improvements. These results demonstrate that object discovery and object representation can be treated as distinct design choices within an object-centric MLLM. In particular, proposal masks produced by one model can be paired effectively with representations from a different 3D representation encoder, without requiring the representation used by the language model to be inherited from the proposal model.

We next isolate the effect of the dense 3D representation source. Within the dense scene setting, Table 2 holds the object proposals and object-token construction fixed, with mean pooling used throughout. The point foundation models consistently outperform the dense features from the 3D detection/segmentation backbones on every reported metric. This trend holds across Sonata, Concerto, and Utonia, indicating that the difference is not specific to a single point foundation model. Concerto-Base achieves the strongest performance on most metrics, while Utonia performs best on Scan2Cap. Compared with Sonata, both Concerto and Utonia achieve stronger downstream performance across most metrics. This trend is consistent with their broader pretraining strategies, including joint 2D–3D and cross-domain pretraining (Zhang et al., 2025a; 2026). Concerto-Base and Concerto-Large achieve similar performance across all reported metrics despite their different representation dimensionalities, suggesting that dimensionality alone is not a dominant factor in the observed performance differences. The point foundation representations considered here are obtained through self-supervised pretraining on large-scale 3D corpora that include ScanNet scenes across dataset splits, but are not optimized using supervision from the downstream 3D-language tasks. The observed improvements therefore demonstrate the effectiveness of these pretrained representations when used as fixed inputs to the object-centric MLLM, but should not be interpreted as measuring clean transfer to unseen ScanNet scenes. Under fixed object proposals and object-token construction, these results show that the choice of frozen dense 3D representation has a substantial effect on downstream performance.

4.4 Effect of Object-Token Construction

In this section, we fix the object proposals and dense point representation, and vary only the object-token construction function g_θ. As shown in Table 3, straightforward mean pooling for object-feature aggregation provides a strong baseline across all evaluated tasks. Max pooling performs

consistently worse, suggesting that retaining only the strongest response along each feature dimension discards information that is useful for constructing the object representation. It can also be observed that increasing the complexity of g_θ yields only small and task-dependent gains. Single-query attention performs best on several metrics; however, its improvements over mean pooling are limited. Perceiver IO is particularly well suited for our setting, as it is designed to aggregate a variable-size set of input features through a learned latent representation into a fixed-dimensional output. Nevertheless, it remains close to the mean-pooling baseline despite introducing substantially more learnable parameters. These results indicate that, for the dense representation considered here, increasing the complexity of object-token construction does not consistently improve downstream performance. This may result from the g_θ is frozen during instruction tuning stage while the projector and LoRA adapters of the language model remain trainable, part of this insensitivity may arise from downstream adaptation compensating for differences among the constructed object representations. We therefore interpret these results as showing that more complex object-token construction provides limited additional benefit under our full training procedure.

5 Conclusion

We presented a controlled study of representation source and object-token construction in object-centric 3D MLLMs. By separating object proposals from the representation supplied to the language model, our formulation enables these factors to be examined independently. Across five ScanNet benchmarks, our strongest point-cloud-only configuration surpasses prior point-cloud-only methods on most comparable metrics. Our controlled experiments show that object discovery and object representation can be treated as distinct design choices, with the same object proposals effectively paired with representations from an independently pretrained 3D representation encoder. Within the dense scene setting, the choice of frozen 3D representation has a substantial effect on downstream performance, whereas increasing the complexity of the object-token construction mechanism provides only limited and task-dependent gains. Overall, these results emphasize representation source as an important design choice for object-centric 3D MLLMs and show that strong pretrained 3D representations can support effective object-level reasoning with comparatively lightweight aggregation mechanisms.

AI Use Statement

In this work, we used generative AI tools to assist with literature search, drafting and editing the manuscript, providing feedback on methodology and experimental design, and writing and refactoring parts of the experiment and analysis code. We did not use generative AI tools to generate, fabricate, or alter experimental results. All AI-assisted research suggestions and interpretations were reviewed by the authors against the experimental evidence and relevant literature. AI-assisted code was verified and tested by the authors, and citations were checked against the original publications. The authors take responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the aid of generative AI.

Reproducibility Statement

We provide the information needed to reproduce our experimental setup throughout the main paper and appendix. The main paper describes the object-centric formulation, representation sources, object-token construction mechanisms, training procedure, downstream tasks, and evaluation metrics. The appendix provides the complete prompt templates, dense representation preprocessing procedure, object-proposal generation protocol, training data composition, training hyperparameters and optimization settings, detailed evaluation splits and scoring protocols. All evaluated benchmarks are publicly available. Training and evaluation code, data-generation scripts, and trained checkpoints will be released upon publication.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick,

Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikołaj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23716–23736, 2022.

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 (CVPR)*, pp. 19129–19139, 2022. doi: 10.1109/CVPR52688.2022.01854.

Duc Cd, Khai Le-Duc, Florent Draye, Chris Ngo, Terry Jingchen Zhang, Bernhard Schölkopf, and Zhijing Jin. PruneGround: Plug-and-play spatial pruning for 3D visual grounding. *arXiv preprint arXiv:2606.31148*, 2026.

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 (ECCV)*, volume 12365 of *Lecture Notes in Computer Science*, pp. 202–221, 2020. doi: 10.1007/978-3-030-58565-5_13.

Dave Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X. Chang. Scan2Cap: Context-aware dense captioning in RGB-D scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3193–3203, 2021. doi: 10.1109/CVPR46437.2021.00321.

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 (CVPR)*, pp. 26428–26438, 2024a.

Sijin Chen, Hongyuan Zhu, Mingsheng Li, Xin Chen, Peng Guo, Yinjie Lei, Gang Yu, Taihao Li, and Tao Chen. Vote2Cap-DETR++: Decoupling localization and describing for end-to-end 3D dense captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(11): 7331–7347, 2024b. doi: 10.1109/TPAMI.2024.3387838.

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*, 2024c.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5828–5839, 2017. doi: 10.1109/CVPR.2017.261.

Jiajun Deng, Tianyu He, Li Jiang, Tianyu Wang, Feras Dayoub, and Ian Reid. 3D-LLaVA: Towards generalist 3D LMMs with omni superpoint transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3772–3782, 2025.

Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-LLM: Extending language model for 3D visual reasoning. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2195–2206, 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. In *Advances in Neural Information Processing Systems*, volume 36, pp. 20482–20494, 2023.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, and Zhou Zhao. Chat-Scene: Bridging 3D scene and large language models with object identifiers. In *Advances in Neural Information Processing Systems*, volume 37, pp. 113991–114017, 2024a.

Haifeng Huang, Yilun Chen, Zehan Wang, Jiangmiao Pang, and Zhou Zhao. Chat-Scene++: Exploiting context-rich object identification for 3D LLM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(8):9813–9825, 2026.

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 *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20413–20451, 2024b.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations (ICLR)*, 2022.

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 (ICCV)*, pp. 3905–3915, 2025.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742. PMLR, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916, 2023.

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 (CVPR)*, pp. 26296–26306, 2024.

Xianzheng Ma, Tao Sun, Shuai Chen, Yash Bhalgat, Jindong Gu, Angel X. Chang, Iro Armeni, Iro Laina, Songyou Peng, and Victor Adrian Prisacariu. Do 3D large language models really understand 3D spatial relationships? In *International Conference on Learning Representations (ICLR)*, 2026.

Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. SQA3D: Situated question answering in 3D scenes. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=IDJx97BC38.

Yongsen Mao, Junhao Zhong, Chuan Fang, Jia Zheng, Rui Tang, Hao Zhu, Ping Tan, and Zihan Zhou. SpatialLM: Training large language models for structured indoor modeling. In *Advances in Neural Information Processing Systems*, 2025.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 311–318, 2002. doi: 10.3115/1073083.1073135.

Zhangyang Qi, Zhixiong Zhang, Ye Fang, Jiaqi Wang, and Hengshuang Zhao. GPT4Scene: Understand 3D scenes from videos with vision-language models. In *International Conference on Learning Representations (ICLR)*, 2026.

Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D: Mask transformer for 3D semantic instance segmentation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8216–8223, 2023.

Hugues Thomas, Chen Chen, and Jian Zhang. Pts3D-LLM: Studying the impact of token structure for 3D scene understanding with large language models. *arXiv preprint arXiv:2506.05689*, 2025.

Ozan Unal, Christos Sakaridis, Suman Saha, and Luc Van Gool. Four ways to improve verbo-visual fusion for dense 3D visual grounding. In *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, pp. 196–213, 2024. doi: 10.1007/978-3-031-73116-7_12.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4566–4575, 2015. doi: 10.1109/CVPR.2015.7299087.

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.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer V3: Simpler, faster, stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4840–4851, 2024.

Xiaoyang Wu, Daniel DeTone, Duncan Frost, Tianwei Shen, Chris Xie, Nan Yang, Jakob Engel, Richard Newcombe, Hengshuang Zhao, and Julian Straub. Sonata: Self-supervised learning of reliable point representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22193–22204, 2025.

Koonting Yip, Qiyan Zhao, Wenhao Yu, Liangyu Yuan, Mingkai Li, Xiaofeng Zhang, Jianmin Ji, Yanyong Zhang, Qing Jiang, and Ka-Veng Yuen. SoPE: Spherical coordinate-based positional embedding for enhancing spatial perception of 3D LVLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Hanxun Yu, Wentong Li, Song Wang, Junbo Chen, and Jianke Zhu. Inst3D-LMM: Instance-aware 3D scene understanding with multi-modal instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14147–14157, 2025.

Tatiana Zemskova and Dmitry Yudin. 3DGraphLLM: Combining semantic graphs and large language models for 3D scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8885–8895, 2025.

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 (ICCV)*, pp. 15225–15236, 2023. doi: 10.1109/ICCV51070.2023.01397.

Yujia Zhang, Xiaoyang Wu, Yixing Lao, Chengyao Wang, Zhuotao Tian, Naiyan Wang, and Hengshuang Zhao. Concerto: Joint 2D-3D self-supervised learning emerges spatial representations. In *Advances in Neural Information Processing Systems*, 2025a.

Yujia Zhang, Xiaoyang Wu, Yunhan Yang, Xianzhe Fan, Han Li, Yuechen Zhang, Zehao Huang, Naiyan Wang, and Hengshuang Zhao. Utonia: Toward one encoder for all point clouds. In *International Conference on Machine Learning (ICML)*, 2026.

Zhi Zhang, Srishti Yadav, Fengze Han, and Ekaterina Shutova. Cross-modal information flow in multimodal large language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19781–19791. IEEE, 2025b.

Duo Zheng, Shijia Huang, and Liwei Wang. Video-3D LLM: Learning position-aware video representation for 3D scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3761–3771, 2025.

Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3D: Exploring unified 3D representation at scale. In *International Conference on Learning Representations (ICLR)*, 2024.

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 (ICCV)*, 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 (ICCV)*, pp. 2911–2921, 2023. doi: 10.1109/ICCV51070.2023.00272.