P^3-VLM: A Point-based Alternative for Grounded 3D Vision-Language Models

Anna-Maria Halacheva, Jan-Nico Zaech, Sombit Dey, Luc Van Gool, Danda Pani Paudel
INSAIT, Sofia University "St. Kliment Ohridski"

anna-maria.halacheva@insait.ai https://p3vlm.insait.ai

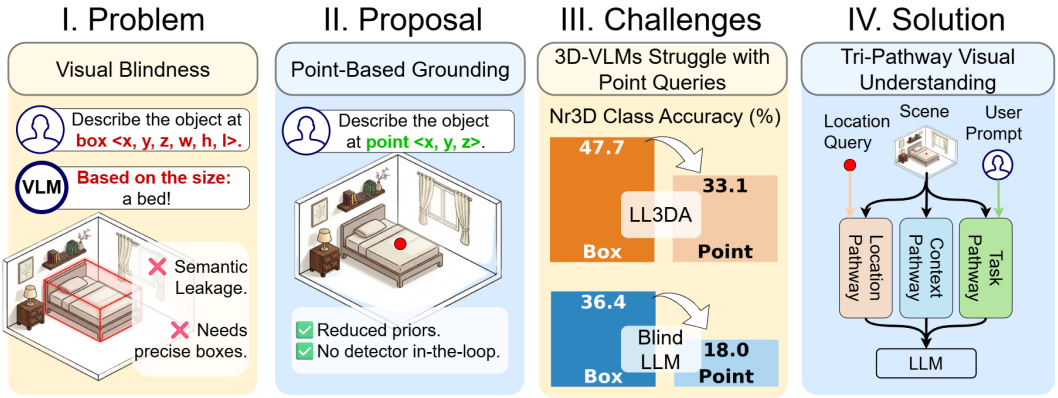

Figure 1: We address bounding box induced semantic leakage and thus the visual blindness in 3D VLMs using a point-based grounding alternative. This is challenging because both 3D VLMs and blind LLMs struggle with point queries. To overcome this, we develop a novel tri-pathway 3D VLM.

Abstract

Examining existing benchmarks for grounding in 3D VLMs reveals a pervasive "semantic leakage" in the evaluation and prompting protocols: a heavy reliance on bounding boxes as spatial prompts or representation primitives. Because bounding boxes are highly informative of object categories, they introduce a shortcut for the models to neglect visual information during 3D VLM training. We demonstrate this with a "blind" LLM that, prompted only with the box coordinates, can perform surprisingly well. To further analyze this shortcut, we propose *point-based grounding prompting* – querying models with a single 3D point instead of a box. Under this protocol, the accuracy of the blind LLM drops from **36.4%** to **18.0%** on ScanNet's Nr3D, and similarly degrades for 3D VLMs (e.g. LL3DA).

Building on this insight, we introduce **P^3-VLM**, a novel tri-pathway 3D VLM that supports point-based prompts and achieves **52.1%** state-of-the-art accuracy on location-grounded captioning under this setting. To achieve this, we redesign the architecture while keeping it detector-free, scene-centric, and suitable for 3D GS inputs. P^3-VLM utilizes three complementary pathways (location-, task-, and context-aware), enabling strong reasoning about scenes without any bounding box dependencies. The key benefit of point-based learning is a consistent improvement across settings: P^3-VLM achieves state-of-the-art performance under point-based grounding while also improving robustness, OOD generalization, and performance on ungrounded scene-centric 3D VQA benchmarks, reflecting its reduced reliance on bounding-box shortcuts and resulting stronger visual-spatial reasoning.

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

1 Introduction

3D Vision-Language Models (VLMs) hold the promise of generalist scene understanding: given a 3D representation of an environment, such a model should be able to answer both grounded queries – those anchored to a specific location or object in the scene – and ungrounded queries about the scene at large [8, 24, 8, 20, 23]. Significant progress has been made in this direction, yet a fundamental question remains open: to what extent do current 3D VLMs actually observe the scene, rather than exploit contextual shortcuts?

We investigate this question through grounded object understanding: identifying the correct object given a 3D spatial prompt. This requires jointly reasoning about geometry and semantics, including localizing the queried region, interpreting its visual content, and producing a semantically accurate response [11, 6, 19, 18]. We find that existing object- and scene-centric paradigms share a common issue of "semantic leakage" in evaluation protocols, arising from the widespread use of bounding boxes as either prompts or representation primitives. Because bounding boxes carry strong categorical cues through their size, shape, and aspect ratio, they enable shortcut solutions (e.g., a tall narrow box often corresponds to a door; beds have standardized measurements) that reduce the need for true visual-spatial reasoning. They also implicitly assume accurate object localization, which is usually unrealistic in real-world 3D applications.

The box dependency appears across both object-centric VLMs [24, 23, 58], which inherit box-based object representations from detectors, and scene-centric models [19, 8], which still rely on bounding boxes as grounding prompts at inference time. As a result, models can exploit categorical priors encoded in box geometry rather than visual evidence. We demonstrate this effect by showing that a blind LLM trained only on bounding box coordinates, without any visual input, achieves 36.4% accuracy on Nr3D as shown in Fig. 1. We further show that the overreliance on bounding boxes makes the existing 3D VLMs fragile, leading to significant degradation in scenarios with atypical object sizes, noisy reconstructions, or poorly aligned bounding boxes in Fig. 2 (robustness analysis).

These findings motivate both a new evaluation protocol and a new model design. We propose *point-based grounding prompting*, where queries are specified using only a single 3D point on the target object. Unlike bounding boxes, point prompts do not reveal object shape or extent, substantially reducing category-specific geometric cues and incentivizing models to rely on genuine visual-spatial understanding. Under this stricter protocol, the shortcut enabled by bounding boxes largely disappears: the accuracy of the blind LLM drops sharply from 36.4% to 18.0% on Nr3D. Existing 3D VLMs, including scene-centric models such as [19, 8], also degrade substantially, further exposing their overreliance on bounding-box cues (e.g. LL3DA drops from 47.7% to 33.1%, see Fig. 1).

To address this new challenging setting, where existing scene-centric 3D VLMs struggle, we introduce **P^3-VLM**, a detector-free, scene-centric 3D VLM built on 3D Gaussian scene representations (3D GS) and designed specifically for point-based grounding. P^3-VLM utilizes a tri-pathway architecture consisting of: (1) a *location-aware* pathway that encodes compact positional cues from the query point, (2) a *task-aware* pathway that sparsifies and refines query-relevant volumetric features, and (3) a *context-aware* pathway that summarizes the global scene for grounded reasoning. By anchoring visual features directly to scene voxels rather than detector-derived geometry, P^3-VLM enables spatially grounded reasoning under the proposed point-based alternative.

Under point-based grounding, P^3-VLM achieves 52.4% class accuracy on Nr3D, substantially outperforming both the blind LLM baseline (18.0%) and previous 3D VLMs adapted to this setting (33.1% and 29.3% for LL3DA and GaussianVLM, respectively; see Tab. 2). This large margin demonstrates that its visual encoder is genuinely load-bearing. Moreover, by avoiding bounding-box shortcuts, P^3-VLM is not only more robust but also exhibits stronger scene-level OOD generalization (see Fig. 2), while maintaining consistent gains on ungrounded scene understanding benchmarks such as ScanQA [3] and SQA3D [33]. Overall, this work makes the following contributions:

- We identify bounding box leakage as a pervasive shortcut in current 3D VLMs, hurting visual-spatial understanding, and introduce point-based grounding as a stricter protocol.
- We present P^3-VLM, a detector-free, scene-centric 3D VLM operating on 3D Gaussian representations via three complementary pathways (location, task, and context aware).
- We show that P^3-VLM achieves SOTA performance under point-based grounding while improving robustness, OOD generalization, and performance on VQA benchmarks.

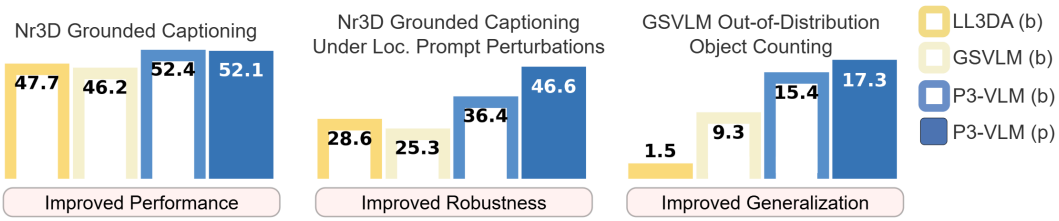

Figure 2: **Performance, robustness, and generalization.** Our P^3-VLM consistently outperforms VLM baselines LL3DA [8] and GaussianVLM [19] (GSVLM). Notably, P^3-VLM on points (p) offers better robustness and generalization than box-based methods (b), despite less spatial information.

2 Related Work

2.1 3D-Grounded Scene Understanding

3D-grounded scene understanding refers to tasks involving visual grounding in 3D space, which can manifest on either the input or the output side of the model [6, 1, 19, 34, 30]. On the output side, the model receives a reference to an object or category and must predict a spatial specification for the matching object(s) – a 3D bounding box or segmentation mask [34, 28, 30, 32, 2, 52, 27, 42]. Specialist models for this setting [28, 30], including those built on language backbones [34, 52, 2], achieve strong performance by combining task-specific pretraining, structured output modeling, and dedicated detection heads. However, because of this specialization, the state-of-the-art methods sacrifice their free-form answer and thus generalist capabilities. Generalist models with multi-task open-language capabilities [8, 23, 24] have been unable to match specialist performance on output-side grounding, lacking, for instance, the structured output modeling and full-scale task-specific supervision. Furthermore, existing free-form benchmarks for diverse scene understanding are largely restricted to ScanNet-scale datasets [3, 33, 20, 24, 25, 6, 1, 19], hindering the training of generalist models for output-side grounding at the scale necessary to achieve state-of-the-art performance [34].

Input-side grounded understanding entails providing the grounding on the input side through a 3D location query, with model output – a caption of the referenced object [19, 8, 18, 58, 24, 56]. This setting is more compatible with open-form answer models. Some free-form VLMs also report bounding box outputs in this setting, yet those are obtained via independent, frozen detectors [9, 41] entirely separate from their core architecture [8, 23, 24], making the box prediction not a product of the model's learned visual understanding. Exploring free-form 3D generalist VLMs, we operate in the input-side captioning setting [19, 18, 56]. The captioning quality directly measures the VLM's visual understanding, not focusing on box predictions, obtainable through an external module.

2.2 3D VLMs for Multi-Task Scene Reasoning

VLMs for 3D scenes aim to support a broad range of tasks, including object captioning [6, 1], question answering [3, 20], spatial reasoning [33, 54] and beyond [19, 20], through a single unified model [19, 8, 55, 39]. The dominant paradigm in this setting has been object-centric: detect a set of object proposals, encode their point cloud features to separate tokens, and align them with language [54, 24, 23, 29, 57, 47, 53, 10, 46, 31, 10]. Prior research suggests that this approach is fundamentally limited as scene-level understanding is bounded by detector recall [19], and is emergent rather than by design [19, 56, 36, 16]. Thus, a parallel line of work moves away from detector dependencies, spanning video VLMs for 3D understanding [17, 55, 39, 22, 7, 49, 45], and native 3D models. The latter processes 3D scenes either (1) in a region-based manner requiring incorporation of advanced methods for instilling cross-regional awareness (e.g., GCNs [18]) or (2) via scene-centric strategies that encode the environment at once as a whole [19, 8, 56]. GaussianVLM [19] pioneered the use of 3D GS for scene-centric understanding and acts as a baseline for this work. However, it suffers from reduced robustness and generalization due to reliance on semantic shortcuts (Sec. 3.2,5.2, D.3).

3 From Bounding Box Semantic Leakage to Point Grounding

A foundational assumption underlying 3D VLM evaluation is that grounded query performance reflects genuine visual-spatial reasoning – an assumption which we investigate in this section. When

models are queried about a specific object via a bounding box prompt – the universal convention across 3D VLMs – the box itself encodes strong categorical information: its dimensions, aspect ratio, and volume are highly predictive of object class. A tall, narrow box standing up from floor level implies a door; a box with measurements of a queen-sized bed – a bed, etc. We term this phenomenon *bounding box semantic leakage* and demonstrate that it acts as a dominant signal that can bypass visual perception in 3D VLMs when it comes to grounding tasks.

3D-Grounded Object Captioning. To investigate this phenomenon, we focus on the task of 3D-Grounded Object Captioning, which requires a model to generate a natural language description for a specific object given a spatial query. We evaluate various 3D VLMs on two ScanNet-based benchmarks: Nr3D [1] and ScanRefer [6]. Firstly, we evaluate models under the standard prompting protocol, *bounding-box prompting*, specifying the query location as a box center and extent.

3.1 Bounding Box Semantic Leakage

We aim to isolate the contribution of geometric priors from other key aspects of 3D grounding such as semantics and context awareness. For this, we conduct two controlled experiments on Nr3D. Specifically, we examine the effect of the strong reliance on bounding boxes within the textual and/or the visual component of the prompt, highlighting that the leakage can occur via both information channels and cause over-reliance on this geometric prior rather than utilization of rich visual features.

Table 1: Object-centric VLMs do not utilize their 3D encoder. A blind LLM operating on bounding box coordinates alone, and an LLM with only box coordinate embeddings, perform close to the full OD-VLM on Nr3D.

Model	Class Acc. (%)
Blind LLM (box coords only)	36.4
Box encoder + LLM	44.2
Full OD-VLM (LEO)	45.1

Leakage via Textual Prompting. First, we examine the effect of bounding boxes as grounding prompts within the textual LLM task prompt. We fine-tune a "blind" LLM (Qwen3-VL 4B Instruct LLM [4]) via LoRA to predict object classes from box center, yaw and extent alone, with no visual encoder of any kind. The fine-tuned blind LLM achieves 36% class accuracy – a significant result that establishes box geometry as a strong categorical signal independent of any visual observation.

Leakage via Architectural Priors. Leakage can also occur through the visual channel, particularly in Object-Detector VLMs (OD-VLMs). These models typically use a decoupled architecture: a detector segments the scene, and a 3D encoder transforms each instance's point cloud into tokens for the LLM. We investigate whether the 3D encoder contributes meaningful semantic understanding beyond what the detector's box metadata already provides.

We ask whether the visual encoder in a full object-centric VLM is strongly utilized on top of this prior. For this, we evaluate an OD-VLM (LEO [24]) against a modified version where the visual features are simply encodings of the bounding boxes (center and extent) of the objects within the scene, derived with a lightweight 2-layer MLP. We observe very close performance of the two models (Tab. 1). These results suggest that, under bounding-box-based visual prompting, object class predictions are once again driven primarily by geometric priors rather than visual features.

3.2 Point-Based Grounding

Our findings indicate that the prevalent use of bounding boxes as spatial prompts may provide geometric priors that simplify the grounding task. To isolate visual perception from these structural cues, we propose *point-based grounding prompting*: a protocol where location queries are specified as a single 3D coordinate (x, y, z). By removing information about the target's extent or aspect ratio, it serves as a more faithful benchmark for a model's ability to extract semantics from visual features.

The effect of removing box geometry is immediate. We fine-tune two variants of the blind LLM: one trained on bounding box prompts and evaluated accordingly, and one trained and evaluated on point-based prompts (Supp. Sec. A.1). Class accuracy drops from 36.4% to 18.0% (Tab. 2) – a 50% relative reduction achieved by changing only the prompt format, with no modification to the model architecture. This confirms the prior inherent in the bounding-box grounding prompts, and its reduced effect in point prompts.

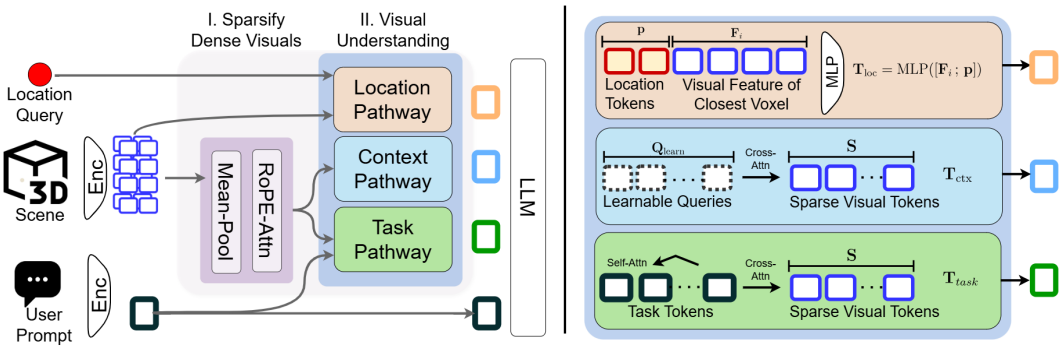

Figure 3: **P^3-VLM architecture overview.** Three complementary pathways process the voxelized 3D scene: the location-aware pathway extracts a single token at the queried point; the task-aware pathway produces prompt-guided scene tokens by first refining the task prompt tokens and subsequently utilizing them for visual refinement of the sparsified features; the context-aware pathway distills 8 global scene summary tokens via unconditional cross-attention. Notably, the context and task pathways operate on the sparsified features for efficiency gains.

We then ask whether existing scene-centric VLMs can close this gap when adapted to the new prompt. For each model, we fine-tune on point-based grounding and re-evaluate under the stricter protocol (training details in the Supplementary A.1). The results in Tab. 2 show that fine-tuning does recover some performance, yet a substantial gap relative to the box-based baseline persists across all models: on Nr3D LL3DA drops from 47.7% to 33.1%, and GaussianVLM from 46.2% to 29.3%. The fact that explicit training on point prompts does not close this gap suggests that the limitation is not one of prompt formatting, but of visual capacity – these models have not developed the dense visual representations that point-based grounding demands.

4 P^3-VLM

P^3-VLM is a scene-centric 3D VLM, that takes as input 3D Gaussian Splats (GS) and a natural language query and produces a free-form language response relying on point-based spatial prompting. An overview of the architecture is shown in Fig. 3. The scene is first voxelized: a sparse 3D GS encoder processes the raw Gaussian splats into a set of voxel features $F \in \mathbb{R}^{N\times D}$ with associated 3D coordinates $C \in \mathbb{R}^{N\times 3}$, where each feature is anchored to a specific spatial location. These voxel features then feed three complementary pathways: location-aware, task-aware, and context-aware, which together produce the visual token sequence as input to the LLM. To the visual tokens, we append the user language task tokens L.

Table 2: **Object captioning performance on Nr3D and ScanRefer with different prompting strategies.** The blind LLM's sharp drop confirms that bounding box geometry was the dominant signal. Also the 3D VLM baselines (LL3DA, GaussianVLM) trained on boxes drop considerably even after fine-tuning on point prompts.

Dataset	Model	Box		Point	
		Acc	Sim	Acc	Sim
Nr3D	Blind LLM	36.4	-	18.0	-
	LL3DA	47.7	48.1	33.1	39.5
	GaussianVLM	46.2	48.2	29.3	35.5
Scan Refer	Blind LLM	38.1	-	18.4	-
	LL3DA	48.02	55.9	29.2	34.9
	GaussianVLM	50.4	59.1	33.2	41.4

4.1 Tri-Pathway Visual Processing

The 3D visual encoder produces rich, yet dense, visual features. Even for our downsampled scans, this can produce more than 10k tokens, which are computationally heavy to be processed as a sequence in the attention mechanism. To alleviate this, we sparsify the dense visual features by aggregation and refinement, which additionally increases their spatial awareness, to make them consumable by the context and task pathways. Notably, the location pathway operates on raw encoder features as it focuses exclusively on locality rather than global awareness. Detailed implementation specifications and hyperparameter choices are provided in Supp. Sec. D.

Sparsification. The raw voxel features are spatially downsampled to at most 2048 tokens per scene via average pooling on a uniform 3D grid, making subsequent self-attention tractable. The downsampled features are then spatially enriched and lifted to LLM space by a stack of spatially-aware self-attention layers where both queries and keys are rotated by 3D Rotary Position Embeddings (3D RoPE) [43]. Applying 3D RoPE to both Q and K encodes pairwise relative 3D distances between voxels directly in the attention scores, giving the projector spatial awareness without positional embedding lookups. The output is $\mathbf{S} \in \mathbb{R}^{N \times D_{\text{lm}}}$ with LLM hidden dimension D_{lm}.

Location-Aware Pathway. The location-aware pathway produces a single token that encodes the visual content at the queried 3D point. Given the query location $\mathbf{p} \in \mathbb{R}^3$, we identify the closest voxel center (with voxel id i). The voxel feature $\mathbf{F}_i$ is concatenated with the normalized query location and passed through a four-layer MLP projecting to LLM hidden dimension:

$$\mathbf{T}_{\text{loc}} = \text{MLP}([\mathbf{F}_i\,;\,\mathbf{p}]) \;\in \mathbb{R}^{D_{\text{lm}}}. \tag{1}$$

The concatenation of raw spatial coordinates with the voxel feature allows the MLP to jointly reason about *what* is at a location and *where* that location is, without a separate conditioning mechanism. When no query location is provided the pathway is suppressed and no location token is injected.

Context-Aware Pathway. The context-aware pathway produces a compact, prompt-independent scene summary. A set of $K{=}8$ learnable query tokens (Q_{learn}) cross-attends over the sparsified and spatially enriched features $\mathbf{S}$ through two cross-attention layers

$$\mathbf{T}_{\text{ctx}} = \text{XAttn}(\mathbf{Q}_{\text{learn}},\, \mathbf{S}) \;\in \mathbb{R}^{K \times D_{\text{lm}}}. \tag{2}$$

Because the queries are unconditional, they learn to summarize scene-level properties (object layout, room type, spatial extent) useful regardless of the query. We observe that this global context allows the task-aware pathway to specialize more sharply: without context tokens, the compressor attends broadly across all voxels; with them, attention concentrates on query-relevant spatial regions.

Task-Aware Pathway. The sparsified visual representation is consumed by the task-aware pathway for further sparsification of the tokens with respect to the user task query. The language query L is processed by two self-attention layers to $\mathbf{L}_{\text{refined}}$. Subsequently, we add a 3D RoPE embedding [43] to each voxel token, which then cross-attends to the text tokens. This asymmetry encodes the relevance of each prompt token for each spatial location in the attention score, allowing selective refinement of spatially relevant voxels. The update is applied as a residual with zero-initialized output projection for stable training:

$$\mathbf{S}_{\text{task}} = \text{LayerNorm}(\mathbf{S} + \mathbf{W}_{\text{out}} \cdot \text{XAttn}_{\text{3DRoPE}}(\mathbf{S},\, \mathbf{L}_{\text{refined}}))\,. \tag{3}$$

The task-refined scene $\mathbf{S}_{\text{task}}$ is compressed via 128 learnable tokens to $\mathbf{T}_{\text{task}}$ LLM-ready tokens.

5 Experiments

5.1 Experimental Setup

Datasets. We follow standard training protocols within the scene-centric 3D VLM field [8, 19] and train our models within the ScanNet [15] settings. Notably, we use 3D GS representation [30] as visual input. We perform joint training on multiple of tasks without task-specific fine-tuning: general question-answering (ScanQA [3]), situated question-answering (SQA3D [33]) and location-grounded captioning (on Nr3D [1], ScanRefer [6]).

Evaluation. Following baseline protocols [19], we evaluate question-answering on a multitude of metrics covering linguistic similarity (METEOR (M) [5], ROUGE (R) [12], BLEU-4 (B4) [35], CIDEr (C) [44]), and exact-match accuracy. For the latter, we adopt the refined EM protocol (EM-R) [24]. For captioning, we evaluate target object identification via accuracy (Acc) and caption similarity via Sentence-BERT Similarity (Sim) [40], with metrics discussion provided in Supp. Sec. A.2. Results are reported on the validation sets of ScanRefer, Nr3D, ScanQA and test set of SQA3D.

Training. Training is carried out in two stages. In the first stage, only the location-aware pathway is active, with the model being trained on point-based grounded captioning (Nr3D and ScanRefer), using the single location token as the sole visual input. This allows the location extractor to learn a stable voxel-to-language mapping before the full scene representation is introduced. In the second stage, all three pathways are active and the model is trained jointly on the full task mixture. Both stages are trained under the standard autoregressive language modeling objective (Supp. Sec. A.5).

Implementation Details. P^3-VLM uses Qwen3-VL-4B [4] as LLM backbone, fine-tuned with LoRA [21] with rank 16 [19, 24]. The 3D encoder is initialized from SceneSplat [30] and kept unfrozen. The three visual pathway components are trained from scratch. Training takes 10 hours on 2 NVIDIA-H200 GPUs with 5 epochs of pre-training the location pathway, followed by 12 epochs of full model training, using batch size 16 across both stages. We use the AdamW optimizer with a cosine annealing learning rate schedule, learning rate 3e-5 and weight decay of 0.05. We perform evaluation every epoch. For further training and hyperparameter details, please refer to Supp. Sec. A.5. We train and evaluate two variants of P^3-VLM differing solely in the adopted location prompting strategy - bounding box (*P^3-VLM (b)* with box format *[center, yaw, extent]*) or point (*P^3-VLM (p)*). For the latter we use the centers of the objects, derived from the ground truth segmentation [15].

To isolate language model performance from architectural contributions, we also retrain one of the baselines, GaussianVLM, using its original training protocol while replacing its language decoder with ours: Qwen3-VL-4B (denoted GaussianVLM^{Q}). This ensures that both P^3-VLM and GaussianVLM^{Q} share identical 3D (SceneSplat) and language backbones.

5.2 Grounded Object Understanding

We evaluate P^3-VLM on the ability to perform grounded visual understanding via location-prompted object captioning on Nr3D and ScanRefer. As other methods do not report class accuracy on the captioned object, we obtain the predictions from all baseline models and extract the predicted object class with an LLM-based evaluator (Gemini 2.5 Flash [13]). More details on our evaluation protocol are provided in Sec. A.3.

Results. As shown in Table 3, both P^3-VLM (b) and P^3-VLM (p) obtain significantly better class-accuracy on both Nr3D and ScanRefer captioning. As expected, P^3-VLM (b), our box-based method, slightly outperforms our point-based method on in-domain grounded tasks, courtesy of having access to the object's geometric properties. The stable sentence similarity results (Sim) demonstrate that P^3-VLM not only identifies the correct object but also maintains high linguistic similarity to the entire caption.

Table 3: **Object captioning performance on Nr3D and ScanRefer.** We report classification accuracy (Acc) and caption similarity (Sim). P^3-VLM consistently outperforms baselines in Acc, showing superior visual grounding across point (p) and box (b) prompts. GaussianVLM^{Q}: [19] with Qwen3-VL-4B LLM, retrained.

	Nr3D		ScanRefer	
Model	Acc	Sim	Acc	Sim
LL3DA (b)	47.7	48.1	48.0	55.9
GaussianVLM (b)	46.2	48.2	50.4	59.1
GaussianVLM^{Q} (b)	47.1	47.8	51.7	59.3
P^3-VLM (p)	52.1	46.0	56.1	54.2
P^3-VLM (b)	**52.4**	46.3	**58.4**	52.5

Priors analysis in points and boxes. Even point prompts are not entirely free of priors: absolute coordinates can still correlate with object category through scene-layout regularities. We examine and quantify these residual priors with a geometry-only kNN classifier, trained on the Nr3D train split and evaluated on the validation split (38 classes with at least 8 instances). Using the object center as the point prompt, the horizontal position (xy) alone yields 8.1% accuracy, the height (z) alone 17.8%, and the full 3D center (xyz) 16.5%, compared to 33.0% for the full bounding box. Height thus accounts for nearly all of the residual signal, while absolute room position is a weak cue. Per class, the center-based kNN reaches $\geq 30\%$ recall on only 3/38 classes (chair, picture, monitor), compared to 15/38 for the box-based one: the residual prior is confined to objects with a consistent mounting or placement height, whereas shape-distinctive objects (e.g., bed, keyboard, door) leak almost entirely through box geometry.

Table 4: **Robustness of 3D VLMs using box prompting to geometric shortcut disruption.** Class accuracy (%) on Nr3D under two failure regimes: non-tight bounding boxes, and noisy reconstructions, provided together with In-distribution performance for reference. Models relying on bounding box semantic leakage degrade sharply across both conditions.

	In-Dist.	Non-Tight Boxes	Noisy Recon.
LL3DA	47.7	28.6	33.4
GaussianVLM	46.2	25.3	32.2
P^3-VLM (b)	**52.4**	**36.4**	**39.9**

Table 5: **P^3-VLM (p) robustness to query point perturbation.** We train on object centers (query offset x=0) and evaluate with query points drawn uniformly at random from a ball of radius $x\times$ the box extent around the center, or uniformly from the object surface. We observe graceful degradation even at extreme offsets.

Query	Nr3D		ScanRefer	
Offset	Acc	Sim	Acc	Sim
0	52.1	46.0	56.1	54.2
0.1	50.6 (-1.5)	45.6 (-0.4)	56.5 (+0.4)	54.0 (-0.2)
0.3	49.9 (-2.2)	45.7 (-0.3)	56.1 (0.0)	54.0 (-0.2)
0.6	47.4 (-4.7)	45.5 (-0.5)	54.1 (-2.0)	53.6 (-0.6)
0.9	46.6 (-5.5)	45.1 (-0.9)	52.8 (-3.3)	53.1 (-1.1)
Surf.	50.7 (-1.4)	45.4 (-0.6)	56.9 (+0.8)	54.1 (-0.1)

5.3 Out-of-Distribution Generalization

Robustness to Bounding Box Perturbation. The bounding boxes provide a categorical prior that often does not hold in noisy or out-of-distribution data, e.g., spaces with larger cabinets or larger windows. We evaluate two failure regimes. (1) Non-tight bounding boxes, produced by imperfect detectors or coarse annotations, dilute the geometric signal that the model has come to depend on. (2) Noisy or partial reconstructions introduce geometric artifacts that corrupt box estimates further.

To simulate regime (1), we shrink each GT box by a per-class scale factor $s_c = \bar{\iota}_c^{-1/3}$, where $\bar{\iota}_c$ is the mean IoU of Vote2Cap-DETR [9] detections on class c, so that the resulting box volume matches the average detection quality of a state-of-the-art detector. Vote2Cap-DETR is chosen as it is adopted within the LL3DA baseline, keeping the simulation as faithful as possible to the conditions these models were designed for. Regime (2) is simulated symmetrically: boxes are expanded by the same per-class factor $1/s_c$, reflecting the tendency of detectors to produce loose predictions under reconstruction noise. Full per-class IoU values and implementation details are provided in Supp. Sec. A.4. As shown in Tab. 4, each of these conditions induces a severe performance drop – not because the visual scene has changed dramatically, but because the leakage channel the model relied on has been disrupted. Notably, P^3-VLM (b) remains the most stable.

Robustness to Query Point Perturbation. To test the spatial stability of P^3-VLM (p), trained on exact center coordinates, we evaluate it on random query points: for offset ratio $x \in \{0.1, 0.3, 0.6, 0.9\}$, the query is drawn uniformly at random from a ball of radius x times the box extent, centered at the object center. At x=0.9 the sampling region covers nearly the full object volume. As shown in Tab. 5, performance degrades gracefully with perturbation distance, dropping by only 5.5 accuracy points on Nr3D and 3.3 on ScanRefer at the most extreme setting ($0.9\times$ extent). We further query points sampled uniformly from the object *surface*, which changes performance only modestly, confirming that P^3-VLM (p) does not depend on the privileged center point.

Table 6: ScanQA Results. GaussianVLM^Q: [19] with Qwen3-VL-4B LLM, retrained.

	ScanQA		
Model	EM-R	M	R
ScanQA [3]	/	13.14	33.33
3D-VLP[26]	/	13.53	34.51
VoteNet+MCAN [51]	/	11.4	29.8
LSceneLLM [56]	/	17.95	/
LL3DA [8]	30.3	22.8	34.7
GaussianVLM [19]	30.7	**22.9**	34.8
GaussianVLM^Q	30.9	21.1	**35.3**
P^3-VLM (p)	**31.9**	22.6	**35.3**
P^3-VLM (b)	30.5	21.2	33.9

5.4 Ungrounded Scene Understanding

We further examine the generalist capabilities of P^3-VLM on three ungrounded scene understanding benchmarks: ScanQA, SQA3D, and GaussianVLM out-of-distribution benchmark (GaussianVLM-OOD) [19] (Tables 6, 7, 8). The latter probes VLM generalization on an object counting task on



Figure 4: **Qualitative results on ScanQA and Nr3D.** (a) P^3-VLM can correctly answer scan questions, sometimes even with answers not captured by the benchmark ground truth. (b) Through the combination of task, context, and location pathway, on Nr3D, P^3-VLM identifies the correct object class, surrounding context, and spatial relationships. The performance remains robust even under perturbations (offset factor $x = 0.9$).

Table 7: SQA3D Results.
GaussianVLM^Q: [19] with Qwen3-VL-4B LLM, retrained.

	SQA3D				
Model	EM-R	B4	C	M	R
LL3DA [8]	43.3	1.0	107.2	21.4	42.14
LEO [24]	50.9	9.4	124.7	25.5	48.4
GaussianVLM [19]	51.4	17.1	129.6	26.4	50.2
GaussianVLM^Q	**51.9**	18.9	125.4	**26.7**	50.0
P^3-VLM (p)	51.8	**30.6**	**130.4**	26.6	50.3
P^3-VLM (b)	**51.9**	19.5	130.3	26.4	**50.5**

Table 8: Out-of-distribution results. Zero-shot evaluation on scenes from unseen ScanNet++ [50].

	GaussianVLM-OOD
Model	EM
LL3DA [8]	1.5
LEO [24]	7.9
GaussianVLM [19]	9.3
P^3-VLM (p)	**17.31**
P^3-VLM (b)	15.42

out-of-distribution ScanNet++ scenes [50], measured via exact match accuracy (EM). P^3-VLM (p) achieves state-of-the-art performance across all three benchmarks, surpassing the state-of-the-art GaussianVLM on most metrics despite being trained with the strictly harder point-based prompting protocol. On the ungrounded scene-level OOD benchmark, P^3-VLM (p) reaches 17.31 EM versus 9.3 for GaussianVLM, with P^3-VLM (b) scoring 15.42. The +1.89 EM advantage of (p) over (b) on this benchmark indicates that removing bounding box supervision improves out-of-distribution robustness, consistent with our hypothesis that point-based training discourages reliance on box-derived shortcuts and encourages stronger visual representations.

5.5 Ablation Studies

We evaluate the contribution of different components within P^3-VLM through a series of ablation studies considering our tri-pathway design (Tab. 9) with ablations on the design of separate components provided in Supp. Sec. C. Our ablations cover location-grounded and ungrounded tasks with both object-level and scene-level understanding focus.

We show the necessity of understanding 3D scenes at three levels of granularity by ablating the three pathways of P^3-VLM. Removing any pathway causes consistent drops in task performance. Notably, the location pathway proves an important component enabling scene-level tasks (SQA3D), although it is used only with location-based queries. As we observe, it allows the other two pathways to focus on scene-level understanding rather than generalist capabilities.

Matched baseline. To fully isolate the contribution of our visual processing, we construct a *matched baseline* that shares the 3D encoder and LLM with P^3-VLM, but uses more simplistic visual features: the query point is represented by the raw feature of its nearest voxel, and scene information is provided only via the context tokens, i.e., without our location MLP and without the task pathway. The matched

Table 9: **Design ablation studies on P^3-VLM (p).** Top: Each of the pathways is removed independently. Bottom: A matched baseline that isolates visual processing innovations by employing a multi-pathway variant with minimal structural complexity – utilizing only nearest-voxel features at the query point and global context tokens.

	Nr3D		ScanQA			SQA3D				
	Acc	Sim	EM-R	M	R	EM-R	B4	C	M	R
w/o location pathway	49.3	43.9	30.4	24.2	34.1	45.6	0	114.6	23.4	44.3
w/o task pathway	50.2	46.0	30.1	24.2	19.2	47.9	22.5	119.0	24.1	46.2
w/o context pathway	50.2	45.2	22.7	14.7	23.3	49.4	34.8	123.2	25.2	47.8
Matched baseline	39.8	41.9	31.3	21.7	33.9	49.6	20.2	124.2	25.3	48.1
Full model	52.1	46.0	31.9	22.9	34.8	51.8	30.58	130.4	26.59	50.32

baseline falls far behind P^3-VLM on point-based grounding (Nr3D: 39.8% vs. 52.1%, ScanRefer: 44.2% vs. 56.1% Acc), showing that the pathway choices and their design are well-motivated.

6 Conclusions

In this work, we highlight the problem of semantic leakage in current 3D VLMs, causing these models to neglect visual information. We propose a new point-based grounding prompting protocol and show that other methods struggle under this new setting. To address this, we present P^3-VLM, a novel 3D VLM based on a tri-pathway design. We demonstrate the proposed point-based grounding approach with our architecture leads to improved visual understanding, demonstrated through improved performance, generalization, and robustness on multiple 3D tasks and benchmarks.

Limitations First, P^3-VLM requires 3D GS representations as input, introducing an extra reconstruction step that adds complexity; however, this step has been performed at scale in prior works [30, 32] and industrial applications [38, 14], and is shown to yield richer visual features [19]. Second, our evaluation is primarily focused on perception-oriented benchmarks, with limited coverage of language-intensive or dialogue-driven tasks. Extending P^3-VLM to more complex multimodal reasoning settings remains an important direction for future work. Third, our approach relies on voxelized visual features, currently obtained via the SceneSplat encoder. Evaluating how well the method generalizes to alternative 3D visual encoders is an interesting avenue for further investigation.

Impact We envision this work as a step toward enabling real-world applications such as high-level planning for navigation and visual assistance systems. Additionally, we hope it draws attention to the issue of information leakage in 3D grounding, encouraging the development of more robust and general-purpose 3D vision-language models.

Acknowledgments

This research was partially funded by the Ministry of Education and Science of Bulgaria (support for INSAIT, part of the Bulgarian National Roadmap for Research Infrastructure).

References

[1] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas J. Guibas. ReferIt3D: Neural listeners for fine-grained 3d object identification in real-world scenes. *European Conference on Computer Vision (ECCV)*, 2020.

[2] Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, Jakob Engel, Edward Miller, Richard Newcombe, and Vasileios Balntas. Scenescript: Reconstructing scenes with an autoregressive structured language model. In *European Conference on Computer Vision (ECCV)*, 2024.

[3] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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

[5] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. *ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005.

[6] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. *European Conference on Computer Vision (ECCV)*, 2020.

[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.

[8] Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, et al. Ll3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[9] Sijin Chen, Hongyuan Zhu, Xin Chen, Yinjie Lei, et al. End-to-end 3d dense captioning with vote2cap-detr. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[10] Yilun Chen, Shuai Yang, Haifeng Huang, Tai Wang, et al. Grounded 3d-llm with referent tokens. *arXiv:2405.10370*, 2024.

[11] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[12] Lin Chin-Yew. Rouge: A package for automatic evaluation of summaries. *Proceedings of the Workshop on Text Summarization Branches Out, 2004*, 2004.

[13] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[14] Eric Cornwell, Dario Macangano, Stanford Lee, and Daniel Zilberman. Guidance for open source 3d reconstruction toolbox for gaussian splats on aws. *See https://aws-solutions-library-samples.github.io/compute/open-source-3d-reconstruction-toolbox-for-gaussian-splats-on-aws.html (accessed 2026-05-05)*, 2024.

[15] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[16] Alexandros Delitzas, Ayca Takmaz, Federico Tombari, Robert Sumner, Marc Pollefeys, et al. SceneFun3D: Fine-Grained Functionality and Affordance Understanding in 3D Scenes. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[17] Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Shijie Zhou, Dilin Wang, Zhicheng Yan, Hongyu Xu, Justin Theiss, Tianlong Chen, Jiachen Li, Zhengzhong Tu, Zhangyang Wang, and Rakesh Ranjan. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction, 2025.

[18] Mei Guofeng, Lin Wei, Riz Luigi, Wu Yujiao, Poiesi Fabio, and Wang Yiming. Perla: Perceptive 3d language assistant. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[19] Anna-Maria Halacheva, Jan-Nico Zaech, Xi Wang, Danda Pani Paudel, and Luc Van Gool. Gaussianvlm: Scene-centric 3d vision-language models using language-aligned gaussian splats for embodied reasoning and beyond. *IEEE Robotics Autom. Lett.*, 10(12):12588–12595, 2025.

[20] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, et al. 3d-llm: Injecting the 3d world into large language models. *NeurIPS*, 2023.

[21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 2022.

[22] Wenbo Hu, Jingli Lin, Yilin Long, Yunlong Ran, Lihan Jiang, Yifan Wang, Chenming Zhu, Runsen Xu, Tai Wang, and Jiangmiao Pang. G^2vlm: Geometry grounded vision language model with unified 3d reconstruction and spatial reasoning. *arXiv preprint arXiv:2511.21688*, 2025.

[23] Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, et al. Chat-scene: Bridging 3d scene and large language models with object identifiers. *NeurIPS*, 2024.

[24] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, et al. An embodied generalist agent in 3d world. *ICML*, 2024.

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

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

[27] Maksim Kolodiazhnyi, Anna Vorontsova, Matvey Skripkin, Danila Rukhovich, and Anton Konushin. Unidet3d: Multi-dataset indoor 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 4365–4373, 2025.

[28] Justin Lazarow, David Griffiths, Gefen Kohavi, Francisco Crespo, and Afshin Dehghan. Cubify anything: Scaling indoor 3d object detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22225–22233, 2025.

[29] Mingsheng Li, Xin Chen, Chi Zhang, Sijin Chen, Hongyuan Zhu, et al. M3dbench: Towards omni 3d assistant with interleaved multi-modal instructions. *European Conference on Computer Vision (ECCV)*, 2025.

[30] Yue Li, Qi Ma, Runyi Yang, Huapeng Li, Mengjiao Ma, Bin Ren, Nikola Popovic, Nicu Sebe, Ender Konukoglu, Theo Gevers, et al. Scenesplat: Gaussian splatting-based scene understanding with vision-language pretraining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[31] Xiongkun Linghu, Jiangyong Huang, Xuesong Niu, Xiaojian Ma, et al. Multi-modal situated reasoning in 3d scenes. *NeurIPS*, 2024.

[32] Mengjiao Ma, Qi Ma, Yue Li, Jiahuan Cheng, Runyi Yang, Bin Ren, Nikola Popovic, Mingqiang Wei, Nicu Sebe, Luc Van Gool, Theo Gevers, Martin R. Oswald, and Danda Pani Paudel. Scenesplat++: A large dataset and comprehensive benchmark for language gaussian splatting. In *Proceedings of the 39th Conference on Neural Information Processing Systems (NeurIPS 2025)*, 2025.

[33] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, et al. Sqa3d: Situated question answering in 3d scenes. *ICLR*, 2023.

[34] 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.

[35] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. *ACL*, 2002.

[36] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Openscene: 3d scene understanding with open vocabularies. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[37] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[38] Pix4D. Transform your work with precise 3D scans and AR with PIX4Dcatch. *https://www.pix4d.com/product/pix4dcatch/ (accessed 2026-05-05)*, 2025.

[39] Zhangyang Qi, Zhixiong Zhang, Ye Fang, Jiaqi Wang, and Hengshuang Zhao. GPT4scene: Understand 3d scenes from videos with vision-language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

[40] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *EMNLP*, 2019.

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

[42] Yichao Shen, Zigang Geng, Yuhui Yuan, Yutong Lin, Ze Liu, Chunyu Wang, Han Hu, Nanning Zheng, and Baining Guo. V-DETR: DETR with vertex relative position encoding for 3d object detection. In *The Twelfth International Conference on Learning Representations*, 2024.

[43] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[44] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[45] Haochen Wang, Yucheng Zhao, Tiancai Wang, Haoqiang Fan, Xiangyu Zhang, and Zhaoxiang Zhang. Ross3d: Reconstructive visual instruction tuning with 3d-awareness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9275–9286, October 2025.

[46] Tai Wang, Xiaohan Mao, Chenming Zhu, Runsen Xu, Ruiyuan Lyu, Peisen Li, Xiao Chen, Wenwei Zhang, Kai Chen, Tianfan Xue, Xihui Liu, Cewu Lu, Dahua Lin, and Jiangmiao Pang. Embodiedscan: A holistic multi-modal 3d perception suite towards embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19757–19767, June 2024.

[47] 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.

[48] 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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22193–22204. IEEE, 2025.

[49] Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10632–10643, June 2025.

[50] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

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

[52] Yuanwen Yue, Theodora Kontogianni, Konrad Schindler, and Francis Engelmann. Connecting the Dots: Floorplan Reconstruction Using Two-Level Queries. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[53] 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)*, pages 8885–8895, October 2025.

[54] Yue Zhang, Zhiyang Xu, Ying Shen, Parisa Kordjamshidi, and Lifu Huang. SPARTUN3d: Situated spatial understanding of 3d world in large language model. In *The Thirteenth International Conference on Learning Representations*, 2025.

[55] 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)*, pages 8995–9006, June 2025.

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

[57] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, et al. 3d-vista: Pre-trained transformer for 3d vision and text alignment. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[58] Ziyu Zhu, Zhuofan Zhang, Xiaojian Ma, Xuesong Niu, et al. Unifying 3d vision-language understanding via promptable queries. *European Conference on Computer Vision (ECCV)*, 2025.

A Experimental Details

A.1 Experimental Details: Point-Based Grounding Adaptation

This section provides full experimental details for the results reported in Sec. 3.2.

Blind LLM Experiments. We fine-tune Qwen3-VL-4B-Instruct [4] on object classification within the Nr3D validation set under two prompt conditions: bounding box and point location. In both cases, the location query is provided as a text string with coordinates rounded to two decimal places. Bounding boxes are encoded via center coordinates, yaw angle and scale, as $\langle c_x, c_y, c_z, yaw, w, h, l \rangle$ without explicitly naming the format in the prompt. Point prompts contain only the 3D coordinate $\langle x, y, z \rangle$.

Fine-tuning is performed with LoRA using rank r=16, scaling factor α=32, dropout 0.05, and a learning rate of 3×10^{-5}. The LoRA adapters are applied to the query, key, value, and output projection layers (`q_proj`, `k_proj`, `v_proj`, `o_proj`). As shown in Fig. 5, validation accuracy saturates between epochs 6–8 and overfitting emerges thereafter.

Scene-Centric VLMs. Based on the convergence behavior observed for the blind LLM, we fine-tune all scene-centric VLMs for 6 epochs. To preserve learned visual representations, only the LLM component of each model is fine-tuned, using the same LoRA configuration described above. Fine-tuning is performed exclusively on Nr3D. We evaluate after every epoch and report results from the checkpoint with the highest validation accuracy. Point prompt format is consistent with the blind LLM and P^3-VLM (p) experiments ($\langle x, y, z \rangle$). For bounding box prompting evaluation, we adhere to the bounding box formats used by the specific baselines.

A.2 N-gram Metrics Conceal Failures in Spatial Understanding

Standard 3D scene understanding benchmarks such as ScanRefer and Nr3D evaluate dense captioning models using n-gram overlap metrics – CIDEr, METEOR, ROUGE – gated by an IoU threshold of 0.25 or 0.50. While these metrics are practical for large-scale evaluation, they measure linguistic surface similarity rather than semantic correctness, and in particular are insensitive to whether the model has correctly identified the class of the referred object. A response that fluently describes the wrong object category can achieve a high n-gram score if it shares common words with the reference caption [19].

To quantify this effect, we compare the correlation of standard n-gram evaluation with class accuracy: the fraction of generated captions from which the correct object category can be extracted. This metric directly measures whether the model has understood what it is looking at, independently of

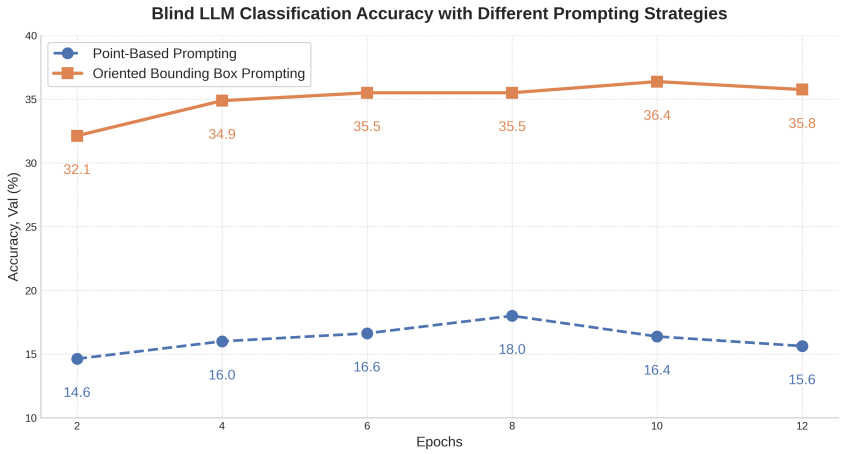

Figure 5: **Validation accuracy of the blind LLM during fine-tuning under bounding-box and point-based prompting.** Performance saturates between epochs 6–8, with overfitting emerging afterwards. This curve motivates the 6-epoch fine-tuning budget used for all VLM experiments.

Table 10: Different evaluation metrics on captioning on the Nr3D dataset. It is evident that high n-gram based metrics do not signal correct object identification.

Model	Sim	M	R	Class Acc
Blind LLM	34.8	17.9	21.3	36.4
LL3DA	48.1	5.8	9.9	47.7
GaussianVLM	48.2	20.8	48.2	46.2
P^3-VLM	46.0	18.6	16.4	52.1

linguistic fluency. As shown in Tab. 10, models that achieve competitive n-gram scores exhibit strikingly low class accuracy – in several cases falling below a blind LLM that receives no visual input whatsoever. These results indicate that n-gram metrics systematically overestimate the spatial understanding of current 3D VLMs, and motivate the use of class accuracy as a more diagnostic evaluation criterion.

A.3 LLM-Based Class Accuracy Evaluator

Class accuracy is computed by extracting the primary object class from each generated caption with Gemini 2.5 Flash [13] and comparing it to the ground-truth class. Captions are processed in numbered batches with the following prompt:

> *You are an object classification assistant. You will receive a numbered list of short scene descriptions. For each description, identify the single primary object being described. Reply ONLY with a JSON array of strings – one object class per entry, in the same order. Use simple, lowercase class names (e.g. "chair", "trash can", "table", "sofa"). If the caption is empty or doesn't describe a recognizable object, reply with "none" for that entry. No explanations, no extra keys, no markdown fences, just the raw JSON array.*

Tab. 11 shows representative extractions. We further carried out a human quality control on 5% of the validation predictions and found no extraction errors, owing to the simple nature of the task: the generated captions name the described object explicitly.

Table 11: **Examples of LLM-based class extraction** on ScanRefer and Nr3D captions generated by LL3DA 3D VLM.

	Generated caption	Extracted class	GT class
ScanRefer	sos this is a white kitchen cabinet . it is to the left of the oven . eos	kitchen cabinet	kitchen cabinet
ScanRefer	sos there is a rectangular soap dispenser . it is on the wall above the sink . eos	soap dispenser	soap dispenser
Nr3D	sos the stool is in the corner of the room . it is to the right of the whiteboard . eos	stool	stool
Nr3D	sos the plant is on the floor . it is to the right of the door . eos	plant	plant

A.4 Out-of-Distribution Generalization

Robustness to Bounding Box Perturbation. To obtain our bounding box perturbation scales, we firstly obtain bounding boxes for every ScanRefer and Nr3D instance via the Vote2Cap-DETR object detector. Here we utilize only the validation split of the datasets as the object detector itself has been trained on the ScanNet dataset train split. The scales are derived based on mean IoU between the ground truth boxes and the predicted ones following our approach introduced in Sec. 5.2. This results to the following scales given in Fig. 6.

A.5 Implementation Details

We train P^3-VLM using AdamW with a learning rate of 3×10^{-5}, $\beta = (0.9, 0.999)$, and weight decay 0.05, with a linear warmup 200 steps followed by cosine decay. Training uses a batch size of

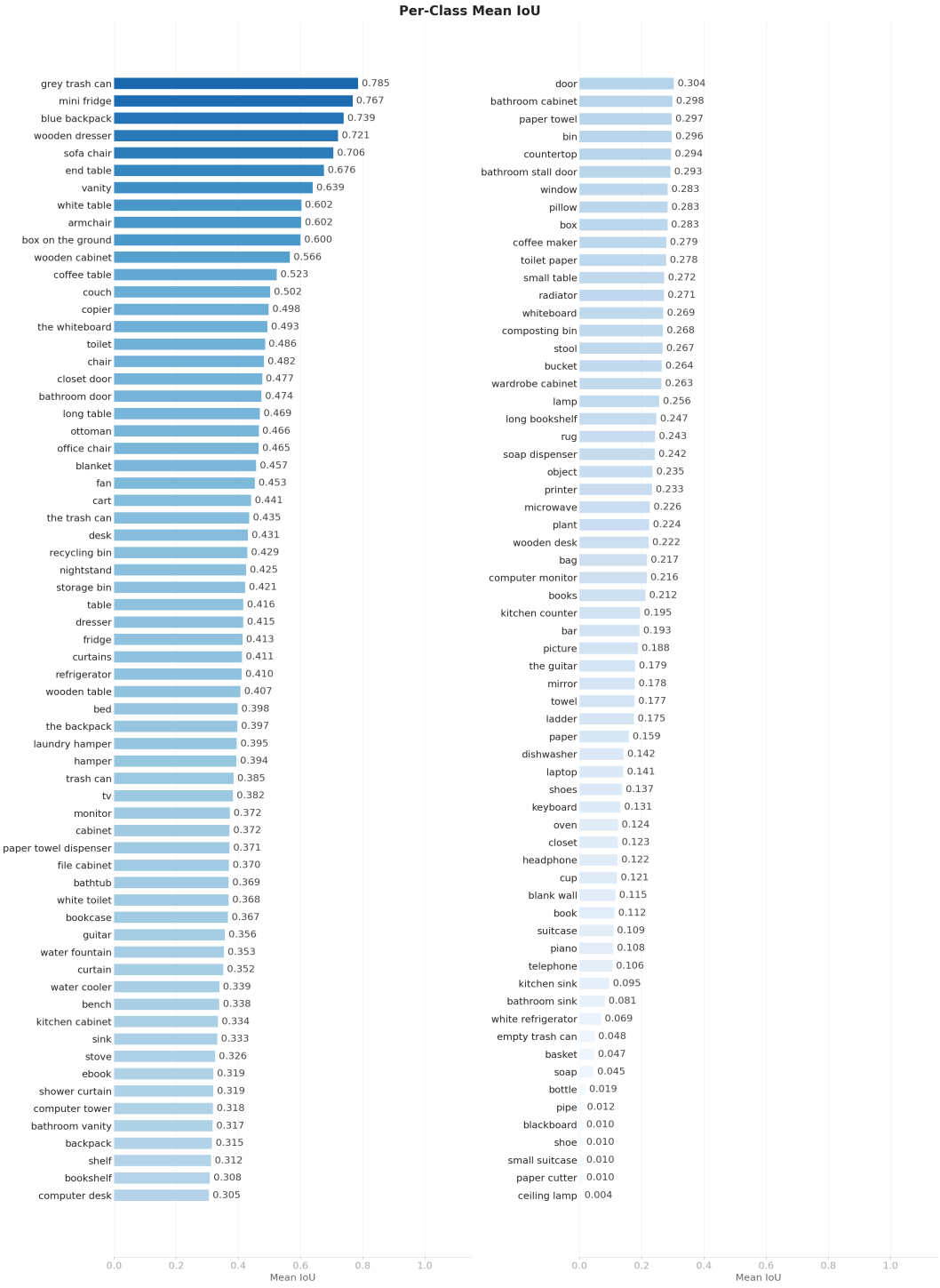

Figure 6: Mean IoU per object class from ScanRefer and Nr3D annotations, validation split.

16 with 4 gradient accumulation steps (effective batch size 64) and gradient clipping at norm 5.0. The language backbone is Qwen3-VL-4B-Instruct, fine-tuned via LoRA (rank 16, $\alpha = 32$, dropout 0.05) applied to all attention projection matrices. The maximum input and output context lengths are set to 2048 and 1024 tokens, respectively. Hyperparameter choices are detailed in Tab. 13 and are selected in close adherence to the GaussianVLM baseline [19]. The training objective is the standard autoregressive objective:

$$\mathcal{L} = -\sum_{t} \log p_\theta \left(y_t \mid y_{<t}, T_{loc}, T_{task}, T_{ctx}, L \right) \tag{4}$$

, where y_t denotes the t-th token of the target response $y = (y_1, \ldots, y_T)$, $y_{<t}$ the tokens generated earlier, and θ the set of trainable parameters. The loss minimizes the negative log-likelihood of the target sequence conditioned on the language prompt L and all three visual pathway outputs: the location token $\mathbf{T}_{\text{loc}}$, the compressed task tokens $\mathbf{T}_{\text{task}}$, and the context summary tokens $\mathbf{T}_{\text{ctx}}$.

A.6 Query point robustness

We carry out robustness experiments of P^3-VLM to the query point offset from the center of the object. We notice that P^3-VLM is very stable to even the extreme condition where the query-point lies near the edge of the object. We plot the mean accuracy across the different samples with error bar indicating 1σ deviation, shown in Fig. 7.

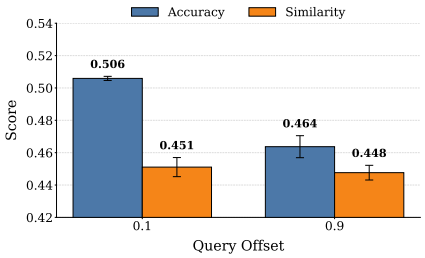

Figure 7: P^3-VLM on Nr3D benchmark

A.7 Training with Surface-Point Grounding

We acknowledge the center point being a canonical point: its height is a stable, category-specific statistic. To escape this prior, we also analyze prompting with *random surface points* instead. A point sampled uniformly on the object surface may lie anywhere within the object's vertical extent, so its height no longer encodes a consistent per-category value. We retrain P^3-VLM with random surface points as the grounding signal during training. As shown in Tab. 12, this yields a further, substantial improvement in grounding accuracy (Nr3D: 58.8% vs. 52.1% for center training; ScanRefer: 64.3% vs. 56.1%), while the caption similarity remains comparable. A surface-trained model evaluated on center points performs worse (50.1% on Nr3D), as the center lies outside of its training distribution.

Table 12: **Center- vs. surface-point grounding** on Nr3D and ScanRefer. c: object center, sp: random surface point. Training on surface points substantially improves class accuracy.

			Nr3D		ScanRefer	
Model	Train	Eval	Acc	Sim	Acc	Sim
LL3DA	box	box	47.7	48.1	48.0	55.9
GaussianVLM	box	box	46.2	48.2	50.4	59.1
P^3-VLM (b)	box	box	52.4	46.3	58.4	52.5
P^3-VLM (p)	c	c	52.1	46.0	56.1	54.2
P^3-VLM (p)	c	sp	50.7	45.4	56.9	54.1
P^3-VLM (p)	sp	c	50.1	42.7	50.4	50.2
P^3-VLM (p)	sp	sp	**58.8**	47.2	**64.3**	54.7

Per-class analysis. To understand where the gain originates, we relate the per-class change in accuracy (center- to surface-point training) to the physical object size, computed from the ground-truth 3D bounding boxes. Medium-to-large furniture (tables, desks, cabinets, shelves, bookshelves, seating) improves consistently, with an average gain of 21.3 accuracy points. Small or thin objects (pillows, boxes, monitors, keyboards, cups) improve much less, by 6.2 points on average; this difference is statistically significant (Mann-Whitney U test, $p = 0.014$). We attribute this behavior to the feature pooling at the query point. Prompting a large object by its center forces the visual features to be pooled from deep inside the object's interior, or even from empty space, e.g., below a table top or inside a cabinet, yielding uninformative features. Surface points, by contrast, anchor the feature

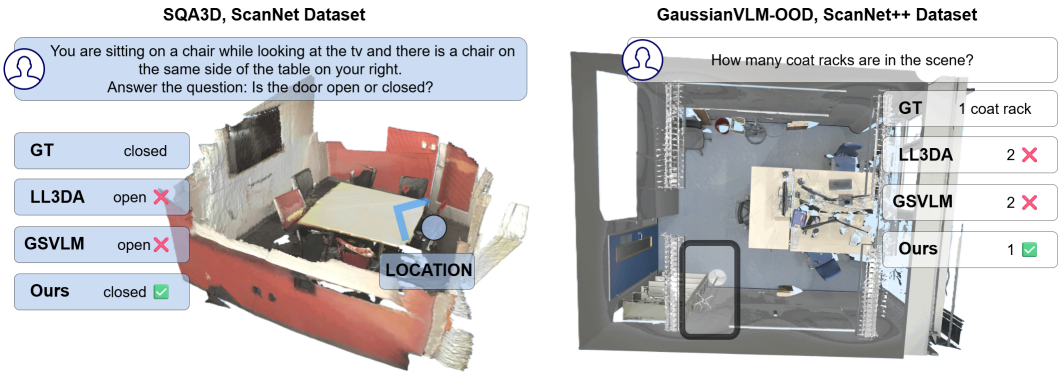

Figure 8: Qualitative results on situated question answering (SQA3D benchmark) and out-of-distribution generalization (GaussianVLM-OOD benchmark). Our model achieves better performance on both benchmarks, showing visual understanding capabilities both on in-domain ScanNet scenes, and on out-of-domain ScanNet++ scenes.

pooling directly to regions rich in visible geometry and texture, which substantially improves visual grounding for larger structures.

B Qualitative Examples

We provide qualitative results showcasing the capabilities of P^3-VLM in Fig. 4, 8 across the tasks of 3D grounded captioning (Nr3D benchmark, validation split), general question-anwering (ScanQA benchmark, validation split), situated question answering (SQA3D benchmark, test split) and out-of-distribution generalization (GaussianVLM-OOD benchmark, test split). We observe visual understanding on in-domain settings within ScanNet input scenes, and out-of-domain on ScanNet++.

Table 13: Training hyperparameters for P^3-VLM.

Hyperparameter	**Value**
Optimization	
Optimizer	AdamW
Learning rate	3×10^{-5}
(β_1, β_2)	$(0.9,\ 0.999)$
Weight decay	0.05
LR schedule	Linear warmup + cosine decay
Warmup steps	200
Gradient clip (norm)	5.0
Batching	
Batch size (per GPU)	16
Gradient accumulation steps	4
Effective batch size	64
Language model	
Backbone	Qwen3-VL-4B-Instruct
Max input length	2048
Max output length	512
LoRA	
Rank (r)	16
Alpha (α)	32
Dropout	0.05
Target modules	`q, k, v, o_proj`

Table 14: **Location pathway and Sparsifier design ablation studies.** We ablate the point-based variant, P^3-VLM (p). *Loc. Design*: raw vs. processed features and FiLM vs. MLP extraction. *Pos. Enc.*: positional encoding strategy in visual sparsification step.

	Nr3D		ScanQA			SQA3D				
	Acc	Sim	EM-R	M	R	EM-R	B4	C	M	R
Location Sparsifier Design										
Processed + MLP	42.3	47.1	-	-	-	-	-	-	-	-
Raw + FiLM	48.5	46.4	-	-	-	-	-	-	-	-
Sparsifier Design: Positional Encoding										
None	47.9	43.7	29.3	18.8	31.1	46.6	28.7	114.5	23.4	44.4
Learned absolute	49.4	46.0	26.9	20.9	32.1	45.3	21.6	113.8	23.3	43.8
Full model	52.1	46.0	31.9	22.9	34.8	51.8	30.58	130.4	26.59	50.32

C Further Ablations

Complementary to the tri-pathway design ablation provided in Sec. 5.5, we also evaluate specific design decisions within the location pathway and sparsifier components of P^3-VLM.

Location Pathway Design. For the location pathway we consider only the location-based captioning task, as the other tasks do not utilize the pathway. First, we ablate whether scene-context information improves local information extraction. The results demonstrate better location understanding on raw voxel features (F) compared to utilization of already sparsified and refined features ($\mathbf{S}$).We attribute this to the sparsifier smoothing voxel features through spatial self-attention, which reduces their locality – precisely the property the location pathway relies on. We also ablate the extraction strategy comparing our MLP-based approach to FiLM location conditioning, with exact details on the FiLM alternative provided in the Supplementary. Results show that processing raw voxel features (F) with an MLP strategy delivers best results.

Sparsification of Visual Tokens. We also ablate the role of positional encodings (PE) for increasing scene-level awareness within the sparsification process. We evaluate the sparsifier module with 3D RoPE PE, learned absolute PE added to voxel features, and no PE. 3D RoPE yields the strongest performance, confirming that explicit spatial structure in the global features is critical.

D P^3-VLM Architecture Details

D.1 Location Pathway Designs.

We investigated two different location pathway designs – FiLM- and MLP-based, Both first identify the nearest voxel to the target location, retrieving its feature vector as a local scene descriptor. The first strategy, **FiLM-based extraction**, conditions this voxel feature on the queried location via Feature-wise Linear Modulation [37]: we use a shallow location encoder MLP (2 linear layers) to map the 3D point coordinates $\mathbf{p}$ to affine parameters $(\gamma, \beta) \in \mathbb{R}^D \times \mathbb{R}^D$. Those are afterwards used to modulate the retrieved voxel feature $\mathbf{F}_i \in \mathbb{R}^D$ as $\mathbf{T}_{\text{loc}} = \sigma(\gamma) \odot \mathbf{F}_i + \beta$, where $\sigma(\cdot)$ denotes the sigmoid function, applied to γ to act as a soft selection mask over feature dimensions. We note that all notation is introduced in Sec. 4.

We note that the MLP-based variant is introduced in Sec. 4. The MLP structure we adopt is four-layer MLP with GELU activations.

D.2 Sparsifier RoPE Details

The spatial refinement is handled by 4 self-attention layers where both queries and keys are rotated by 3D Rotary Position Embeddings. For head dimension d_h (divisible by 6), per-axis frequencies are:

$$\theta_{i,\,\text{axis}} = \frac{\text{pos}_{\text{axis}}}{1000} \cdot \text{base}^{-2i/d_h}, \quad i = 0, \ldots, \tfrac{d_h}{6} - 1, \tag{5}$$

with $\text{base} = 10{,}000$.

D.3 Comparison to the GaussianVLM Architecture

P^3-VLM and GaussianVLM [19] are both scene-centric 3D GS VLMs, but are built around different priorities, from which their architectural differences follow. GaussianVLM is *semantics-oriented*: it fuses language into the 3D backbone itself (per-Gaussian SigLIP2 features), so every downstream token is semantically grounded before sparsification. This favors scene-level tasks and semantic VQA, where broad semantic understanding matters more than precise spatial extent. P^3-VLM is *geometry-oriented*, designed to strengthen grounding, which leads to the following design decisions:

- **Context pathway.** A dedicated, unconditional pathway explicitly recovers global scene semantics; GaussianVLM has no counterpart.
- **Task pathway.** In P^3-VLM, voxel tokens query the text, and the result is applied as a residual correction to the visual features *before* compression, i.e., language reshapes geometry. In GaussianVLM, text tokens query and directly select visual tokens, i.e., language selects semantics.
- **Location pathway.** P^3-VLM derives the local token with an MLP over the concatenated geometric and semantic features at the queried point, explicitly learning location understanding jointly with content. GaussianVLM retrieves the per-Gaussian semantic feature nearest to the query point, without a comparable geometric grounding mechanism.
- **Positional information.** P^3-VLM fuses positional information early via 3D RoPE, reinforcing geometry throughout the visual processing; GaussianVLM adds position as a final Fourier embedding, secondary to semantics.

D.4 Point Cloud Encoder Variant

To assess whether P^3-VLM generalizes beyond 3D GS inputs, we replace the SceneSplat encoder with the self-supervised point cloud encoder Sonata [48] (P^3-VLM-PC), keeping the tri-pathway design and training protocol unchanged. Results are shown in Tab. 15. Owing to Sonata's large-scale pre-training, P^3-VLM-PC achieves strong location-grounded captioning results under point prompts (Nr3D: 58.1%, ScanRefer: 63.3% Acc), outperforming the 3D GS variant. However, Sonata features are not language-aligned, unlike SceneSplat features, which results in lower performance on the VQA benchmarks. This shows that the tri-pathway design transfers to other 3D representations, with the choice of encoder trading off grounding against language-aligned scene understanding.

Table 15: **P^3-VLM (p) with a 3D GS (SceneSplat) vs. point cloud (Sonata) encoder** on grounded captioning (Nr3D, ScanRefer) and VQA (ScanQA, SQA3D).

	Nr3D		ScanRefer		ScanQA			SQA3D				
Encoder	Acc	Sim	Acc	Sim	EM-R	M	R	EM-R	B4	C	M	R
SceneSplat (3D GS)	52.1	46.0	56.1	54.2	**31.9**	22.6	**35.3**	**51.8**	**30.6**	**130.4**	**26.6**	**50.3**
Sonata (PC)	**58.1**	**47.3**	**63.3**	**54.8**	30.5	**22.8**	33.9	48.3	16.8	126.9	23.8	48.8

E Datasets Information

Point Cloud and 3D GS Representations. Our method uses 3D GS scene input, differing from the point cloud input consumed by other methods. To ensure maximal similarity in setup to the point cloud-based methods, we use 3D GS representations with Gaussian centers initialized at the point locations from the original point clouds ("Fixed XYZ" strategy). This guarantees the exact same geometrical properties under the two setups. Our 3D GS data is consistent to the one utilized within the GaussianVLM experiments and is obtained from the GaussianWorld dataset [30].

Dataset Licenses and Terms of Use. We report the licenses and terms of use for all datasets in Table 16 and adhere strictly to each. All datasets are publicly released. We draw attention to ScanNet [15] and ScanNet++ [50] being subject to custom terms of use, requiring separate inspection.

All other datasets are released under standard open-source licenses. The 3D GS representations of ScanNet and ScanNet++ scans are obtained from the respective subsets under GaussianWorld (SceneSplat-7K) and inherit their respective terms.

Table 16: Licenses and terms of use for the used datasets.

Dataset	**License / Terms of Use**
ScanNet [15]	ScanNet Terms of Use[1]
ScanRefer [6]	Creative Commons BY-NC-SA 3.0
ScanQA [3]	Apache 2.0
SQA3D [33]	Apache 2.0
Nr3D [1]	MIT
ScanNet++ [50]	ScanNet++ Terms of Use[2]
GaussianWorld (SceneSplat-7K) [30]	Same as ScanNet, ScanNet++

F Broader Impact

Our work aims to make a step towards VLMs with strong visual understanding capabilities, showing strong robustness and out-of-distribution generalization. With this we invision real-world utilization of 3D VLMs for applications that require free-form, open-world, spatially grounded reasoning, such as robotics, assistive technologies, and AR/VR systems.

We also note the potential environmental implications of training VLM models such as ours. While P^3-VLM is designed with sparsification modules and scene downsampling, its training still requires significant compute resources. With this in mind, we underscore the necessity of research into environmentally-friendly training approaches.

[1] Available at `https://kaldir.vc.in.tum.de/scannet/ScanNet_TOS.pdf` (last accessed: 03/05/2026).

[2] Available at `https://kaldir.vc.in.tum.de/scannetpp/static/scannetpp-terms-of-use.pdf` (last accessed: 03/05/2026).

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The claims of the paper are backed by the results sections.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitation section discusses it.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper provides every detail to reproduce the experiments. Code will be shared after acceptance.

   Guidelines:

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

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

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

Answer: [No]

Justification: We do not provide preliminary access to our codebase, but it will be made openly accessible upon acceptance. The code release will include the codebase for our proposed method and the related experiments, and the weights for the trained models. No data release is necessary, as all data (scans and annotations) we use are publicly available.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We have provided all experimental details across the main manuscript and the supplementary material.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We carry out statistically significant evaluations wherever sampling is involved and report mean and 1σ deviations. Appendix A.5 shows results with error bars.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The experimental details provided in the main manuscript and the supplementary material specify our setup regarding compute resources.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have closely followed the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: A discussion on societal impacts, both positive and negative, is provided in the supplementary material.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The trained 3D VLM model for indoor scene understanding on grounded and question-answering tasks does not fall within the categories for high risk of misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All datasets are clearly specified and credited. Information on license and terms of use is provided in the supplementary.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: All assets (codebase and trained model weights) are documented and the documentation will be released with the assets upon acceptance. For the codebase we have followed standard README practices (environment requirements, .yaml files for setup, setup instructions, dataset links and preprocessing scripts and bash scripts for direct training, evaluation and inference). The models are accompanied by Hugging Face model cards.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not include crowdsourcing experiments and research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not include crowdsourcing experiments and research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: We explicitly declare the use of an LLM for our experiments in deriving object classes. Specifically, the model is employed to extract the primary subject from baseline captions.

    Guidelines:

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