arXiv:2609.08345v1 [cs.CV] 8 Sep 2026

CoVeR: Coverage-Based Token Pruning for Multi-View 3D Reasoning in VLMs

Nhat-Tan Bui[1,*] Varshini Elangovan[1,*] Arun Reddy Anugu[1] Sreyas Mohan[2] Wei Ye[2]
Dilin Wang[2] JQ Huang[2] Rakesh Ranjan[2] Aviral Chharia[1,†] Fernando De la Torre[1]
[1]Carnegie Mellon University [2]Meta Reality Labs
https://humansensinglab.github.io/CoVeR

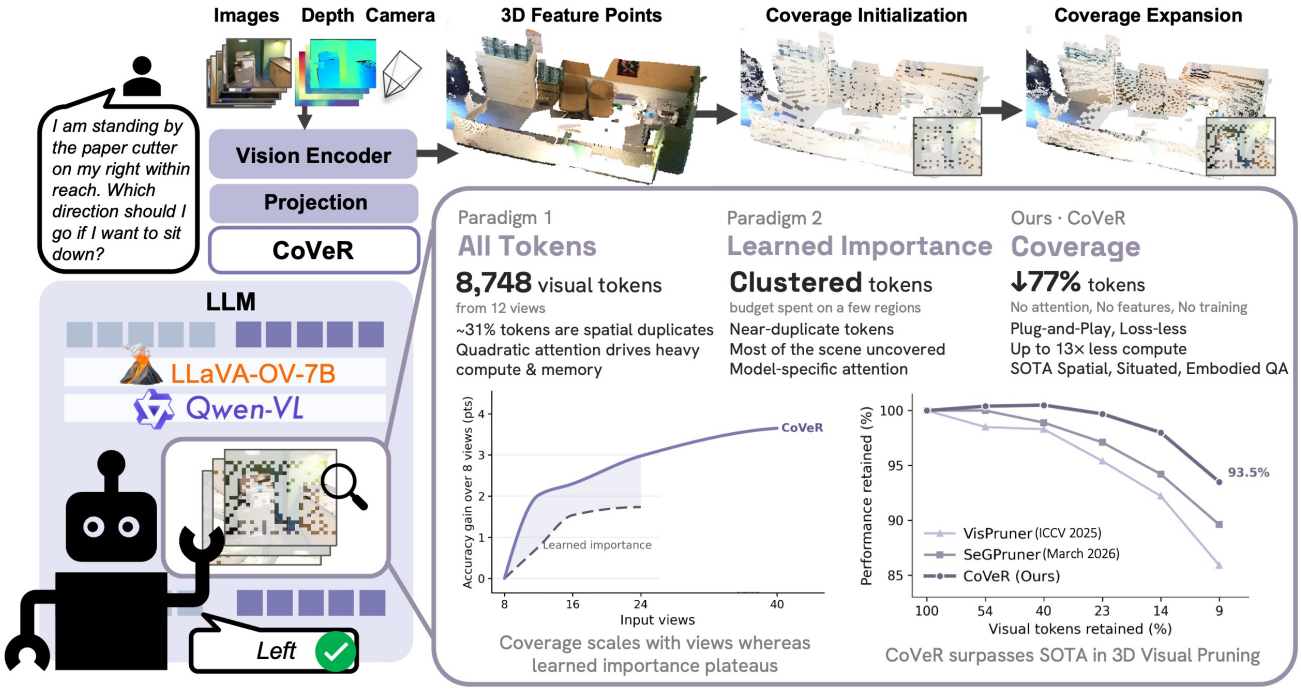


Figure 1. **Overview of CoVeR.** A deterministic, training-free token selector for multi-view 3D reasoning in 2D VLMs that prunes by spatial coverage, using geometry alone. We show that CoVeR surpasses state-of-the-art token pruning methods in the multi-view 3D setting.

Abstract

*Representing a 3D scene as multi-view images allows 2D VLMs to reason in 3D by reusing priors from pre-training, sidestepping the scarcity of annotated 3D data. However, it produces thousands of redundant visual tokens whose cost grows with every view. Existing visual token pruners fall into two families, each limited in the 3D multi-view setting. Learned importance methods rank tokens by attention or encoder features; because redundancy here is fundamentally spatial, they keep near-duplicate tokens from a few prominent regions and leave most of the scene unrepresented. Voxelization methods improve spatial coverage but cannot enforce an exact token budget and saturate as multi-view observations overlap in 3D, capping retention well below the target. We show that spatial coverage is associated with 3D reasoning performance and introduce **CoVeR**, a deterministic, training-free selector that uses only token coordinates, with no learned signals. CoVeR selects tokens that collectively cover every region of the scene, and solves the limitations of both families: it enforces an exact per-scene budget, breaks the voxelization saturation plateau, and avoids the near-duplicate selections of learned importance. Extensive experiments show CoVeR outperforms prior SOTAs on all **three** 3D reasoning benchmarks and generalizes as a plug-and-play module tested across **four** VLMs. Notably, with only ≈8% of visual tokens, it preserves 93.5% of full-token performance, surpassing SOTA by 3.9 percentage points on average across benchmarks.*

*Equal Contribution †Corresponding Author

1. Introduction

Reasoning about 3D space is a prerequisite for systems that perceive and act in the physical world. Directly training models at scale on 3D representation [8, 17, 18, 21, 23, 25, 26, 43, 44, 50, 51, 61–63, 69] is limited by the scarcity of 3D-language data, which is orders of magnitude smaller than the internet-scale image-text datasets behind 2D VLMs [3, 4, 32]. A practical alternative renders the scene as multi-view images and uses a pre-trained 2D VLM to reason over them [11, 16, 19, 20, 27, 45, 53, 57, 82, 83], inheriting their strong visual and language priors. However, this introduces a major bottleneck: the number of visual tokens grows linearly with the number of views. For example, 12-views produce 8,748 visual tokens in `LLaVA-OneVision-7B` [32], substantially increasing the LLM inference cost. Reducing the visual token count is thus necessary for scaling 3D reasoning on 2D VLMs.

Most token pruning methods retain the top-K tokens ranked by *learned importance* from attention or visual features, an intuition inherited from single-view pruning [5, 7, 36, 47, 48, 60, 67, 68, 75, 78]. We argue this is ill-suited to the multi-view 3D setting, where the dominant redundancy is geometric: different cameras observe the same physical regions, and importance scores computed independently of geometry retain near-duplicate tokens while leaving distinct objects or boundaries elsewhere under-represented. Such signals are also model-specific, depending on attention or auxiliary features that vary across architectures.

Voxelization-based pruners back-project tokens into 3D space and pool those falling in the same voxels. This improves coverage but does not control the output token count: a fixed voxel size leaves a different number of tokens in different scenes, and the count plateaus once overlapping views already share voxels. For example, ≈31% of tokens overlap (Fig. 2(a)), so voxel-only pruning keeps about 69% of tokens on average and 46-57% in the most redundant scenes (Fig. 2(b)). Such methods give coverage but not an exact budget.. Exact control matters because a single scene can exceed a fixed memory or latency limit even when the dataset average stays within it. Both approaches are thus insufficient in the 3D multi-view setting.

We address this with **CoVeR**, a deterministic, training-free selector that uses only token coordinates, with no attention, features, or learned signals. *Coverage initialization* finds a scene-specific voxel size and keeps one representative token per occupied voxel, removing overlapping observations while preserving a coarse cover of the whole scene. *Coverage expansion* then iteratively selects the tokens farthest from those retained, adding tokens in the least-covered regions until the budget is met exactly, thereby recovering fine-scale evidence beyond the saturation limit. Both stages reduce the directed Hausdorff distance from the original token set to the retained subset. CoVeR yields consistent gains at aggressive budgets and transfers across VLMs. Our contributions include:

Table 1. **Comparison with token pruning methods.** Avoiding learned signals (attention, visual features, auxiliary encoders) removes model-specific dependence, while deterministic selection and exact per-scene budget give precise control over token count.

	Voxelization		Learned Importance			Ours
Property	VTC [24]	DTC [24]	VisPruner [75]	SeGPruner [34]	Geo3DPruner [33]	CoVeR
Attention-free	✓	✓	✗	✗	✗	✓
Visual feature-free	✓	✗	✗	✗	✗	✓
Auxiliary encoder free	✓	✓	✓	✓	✗	✓
Training-free	✓	✓	✓	✓	✗	✓
Deterministic	✓	✗	✓	✓	✓	✓
Exact per-scene budget	✗	✗	✓	✓	✓	✓

- **Problem analysis.** We identify key limitations in both 3D token pruning families: voxelization-based methods cannot enforce exact per-scene budgets and are capped by voxel saturation, while learned importance-based methods spend their budget on near-duplicates, leaving the scene under-covered.
- **Coverage-based paradigm.** We propose CoVeR, a training-free, deterministic, geometry-only token pruning framework that optimizes for scene coverage under a guaranteed exact per-scene budget.
- **Analytical insights.** Statistical metrics link geometric coverage to 3D reasoning, while directed distances show it preserves regions favored by learned importance; stage-wise analysis shows real tokens beat merged features and pure spatial distance outperforms learned signals.
- **Extensive evaluation.** CoVeR achieves SOTA on spatial scene understanding (ScanQA [2]), situated reasoning (SQA3D [40]), and embodied question answering (OpenEQA [42]), while generalizing across four VLMs. On ScanQA at 14% retention, it cuts LLM TFLOPs by 8.6× and KV cache by 7×, with a 1.4× lower GPU memory and 2.5× inference speedup at a 1.1% relative drop.

2. Related Works

***Voxelization*-based Pruning.** These methods back-project tokens into 3D and reduce them within voxels: VTC [24] averages a voxel's features into a synthetic token, while DTC [24] raises voxel resolution and merges tokens by feature similarity. In all cases, the token count stays tied to geometry, so reported budgets are dataset averages, not exact per-scene counts.

***Learned importance*-based Pruning.** These methods instead anchor on attention or visual features. SeGPruner [34] combines attention-ranked initialization with diversity stage under a joint semantic-spatial metric, making its coverage

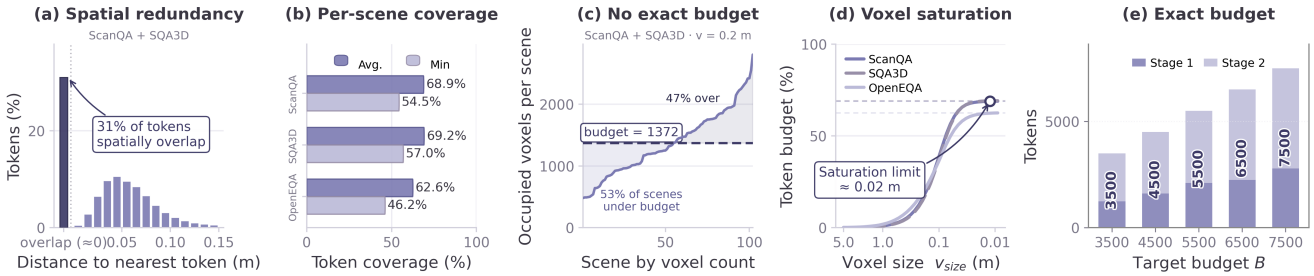


Figure 2. **Motivation (a-d) and Exact-budget Results (e).** Analysis using 12-frame inputs to `LLaVA-OneVision-7B`.

attention-anchored. Geo3DPruner [33] relies on attention and introduces a large `VGGT` [58] encoder, re-training the backbone. VisPruner [75] ranks tokens by text-visual attention and encoder features. CoVeR differs from prior works on several axes (Table 1): selection is purely geometric, using no attention, encoder features, or semantic similarity; it is training-free and deterministic; and it yields exact per-scene budgets that current voxelization-based pruners cannot guarantee.

3. Methodology

3.1. Problem Formulation

We aim to design a selector that, for a budget B, returns B tokens whose 3D locations represent the whole observed scene, using geometry alone and no learned signals. The budget must hold on every scene rather than a dataset average, and coverage must be an explicit objective rather than a by-product of ranking. Given posed RGB-D views, a frozen visual encoder produces M patch tokens, each back-projected to a world coordinate $\mathbf{t}_i \in \mathbb{R}^3$. Let $X = \{\mathbf{t}_i\}_{i=1}^{M}$ denote the token locations and let $\mathcal{C} \subseteq \{1, ..., M\}$, $|\mathcal{C}| = B$ index the selected tokens. For a point $\mathbf{t}$ and an index set $\mathcal{A}$, we define $\delta(\mathbf{t}; \mathcal{A}) = \min_{j \in \mathcal{A}} ||\mathbf{t} - \mathbf{t}_j||_2$. We ask: given exactly B retained tokens, how well do they cover the observed scene? We measure this with the directed Hausdorff distance,

$$d_H(X, \mathcal{C}) = \max_{\mathbf{t} \in X} \delta(\mathbf{t}; \mathcal{C}) \tag{1}$$

i.e., the distance from the worst-covered token to its nearest retained token. Intuitively, d_H asks how far the most under-represented part of the scene sits from anything retained, so a small value certifies that no region is dropped outright, which is precisely the guarantee *learned importance* does not offer. Therefore, we aim

$$\mathcal{C}^* = \underset{\mathcal{C} \subseteq \{1,...,M\}, |\mathcal{C}|=B}{\arg\min} d_H(X, \mathcal{C}) \tag{2}$$

the discrete Euclidean k-center problem.

3.2. Why Voxelization Is Insufficient

A natural route to the k-center objective is voxelization: at voxel size v_s, token i has index $\mathbf{v}_i = \lfloor \mathbf{t}_i / v_s \rfloor$, occupied voxel k holds $\mathcal{V}_k = \{i \mid \mathbf{v}_i = k\}$, and keeping one token per occupied voxel returns $G(v_s; X)$ tokens, where $G(v_s; X)$ is the occupied-voxel count. This maps repeated cross-view observations to the same voxel, but its output is governed indirectly by v_s rather than by a token count, which creates two problems.

No exact per-scene budget. Meeting budget B requires $G(v_s; X) = B$, yet occupancy depends on scene geometry, extent, and view overlap, so one v_s under- or overshoots B across scenes. At a fixed v_s=0.2 m and B=1342, 56% of scenes fall below budget and 44% exceed it (Fig. 2(c)). VTC inherits this variability, and DTC matches retention only on average through tuning; neither guarantees a per-scene memory or latency limit.

Voxel occupancy saturates at practical resolutions. Reducing v_s raises occupancy only up to a point: many tokens are near-duplicate observations of the same surface from different views, so past a certain resolution smaller voxels no longer separate them. About 31% of ScanQA and SQA3D tokens spatially overlap (Fig. 2(a)), so achievable retention plateaus at ≈69% near v_s=0.02 m (Fig. 2(d)); highly redundant scenes saturate at 46.2–57% (Fig. 2(b)). Within the practical range, voxelization alone therefore cannot reach arbitrary budgets, motivating our method.

3.3. Overview

Fig. 1 and Algo. 1 summarize CoVeR. *Coverage initialization* (Sec. 3.4) adaptively voxelizes a scene and keeps an original token per occupied voxel, removing cross-view duplicates into a coarse scene-wide spatial cover. *Coverage expansion* (Sec 3.5), then repeatedly adds the token farthest from the current selection, extending coverage to regions underrepresented by voxelization, until exactly B tokens remain. The two stages are complementary. Voxelization is cheap and spreads tokens over the whole scene at once, but its output count is capped by resolution; farthest point sampling (FPS) [15] can place a variable number of tokens, but builds a spread slowly.

Budget split. A single hyperparameter $\alpha \in (0, 1)$ sets the stage 1 target $B_{\text{init}} = \max(1, \lfloor \alpha B \rfloor)$; stage 2 then supplies the remaining $B_{\text{expan}} = B - |\mathcal{C}_{\text{init}}|$ tokens. For any feasible budget $B \leq M$, three cases ensure exactly B output tokens:

(1) if $|\mathcal{C}_{\text{init}}| < B$, stage 2 adds the remaining tokens (i.e., $B_{\text{expan}} \geq 1$), this is the regime observed for all evaluated scenes and budgets; (2) if $|\mathcal{C}_{\text{init}}| = B$, stage 2 is empty; and (3) if $|\mathcal{C}_{\text{init}}| > B$, a safeguard (Alg. 1) retains representatives from the B most populated voxels (which is not activated at any scene or budget with our ratio α). Thus, regardless of where the voxel search terminates, the selector always returns exactly B tokens (Fig. 2(e)).

Integration with a VLM. CoVeR is a plug-in module that selects token indices $\mathcal{C}$ right after the visual encoder. All visual tokens still pass through the projector, and after projection, only the features indexed by $\mathcal{C}$ reach the LLM, kept in original sequence order. The native token layout is left intact with pruned tokens simply removed, making CoVeR compatible across diverse VLMs whose projectors preserve token-wise correspondence.

Algorithm 1 Coverage-based 3D Token Pruning

Inputs: Token coordinates $X = \{\mathbf{t}_i\}_{i=1}^{M}$, budget B, ratio α
Constants: $\tau = 0.05$, $T = 16$, $[v_{\min}, v_{\max}] = [0.02, 5.0]$
Output: Selected token set $\mathcal{C}$, $|\mathcal{C}| = B$

$B_{\text{init}} \leftarrow \max(1, \lfloor \alpha B \rfloor)$
Stage 1: Coverage initialization
$v_{\text{lo}} \leftarrow v_{\min},\ v_{\text{hi}} \leftarrow v_{\max}$
for $t = 1, \ldots, T$ **do**
  $v_s \leftarrow (v_{\text{lo}} + v_{\text{hi}})/2$
  $G \leftarrow |\{\lfloor \mathbf{t}_i / v_s \rfloor\}_{i=1}^{M}|$
  if $(1-\tau)B_{\text{init}} \leq G \leq (1+\tau)B_{\text{init}}$ **then break**
  $v_{\text{lo}} \leftarrow v_s$ **if** $G > (1+\tau)B_{\text{init}}$ **else** $v_{\text{hi}} \leftarrow v_s$
end for
$\mathcal{C}_{\text{init}} \leftarrow$ representative per occupied voxel ▷ Eq. 3
if $|\mathcal{C}_{\text{init}}| > B$ **then** $\mathcal{C}_{\text{init}} \leftarrow$ representatives of B most populated voxels ▷ budget safeguard
Stage 2: Coverage expansion
$B_{\text{expan}} \leftarrow B - |\mathcal{C}_{\text{init}}|, \quad \mathcal{C}_{\text{expan}} \leftarrow \emptyset$
$d_{\min}(m) \leftarrow \min_{s \in \mathcal{C}_{\text{init}}} \|\mathbf{t}_m - \mathbf{t}_s\|_2^2 \quad \forall m \notin \mathcal{C}_{\text{init}}; \quad d_{\min}(s) \leftarrow -1 \ \forall s \in \mathcal{C}_{\text{init}}$
for $j = 1, \ldots, B_{\text{expan}}$ **do**
  $m^* \leftarrow \arg\max_m d_{\min}(m)$
  $\mathcal{C}_{\text{expan}} \leftarrow \mathcal{C}_{\text{expan}} \cup \{m^*\}, \quad d_{\min}(m^*) \leftarrow -1$
  $d_{\min}(m) \leftarrow \min(d_{\min}(m), \|\mathbf{t}_m - \mathbf{t}_{m^*}\|_2^2) \ \forall m$
end for
return $\mathcal{C} \leftarrow \mathcal{C}_{\text{init}} \cup \mathcal{C}_{\text{expan}}$

3.4. Coverage Initialization

Stage 1 constructs a coarse scene-wide cover. As $G(v_s; X)$ varies with scene geometry, CoVeR estimates v_s per scene rather than transferring one global value across the dataset.

Adaptive voxel size via heuristic interval search. Because voxel grids at different sizes are not nested, $G(v_s; X)$ is not guaranteed to be monotonic in v_s. However, decreasing v_s tends to increase $G(v_s; X)$ (Fig. 2(d)). Motivated by this observation, we use binary search as a heuristic over an interval $[v_{\text{lo}}, v_{\text{hi}}]$, [1], raising v_s when too many voxels are produced and lowering it when too few are, until G lands within a tolerance τ of B_{init} or T iterations are reached. Because the search is heuristic, the terminal occupancy may fall on either side of B_{init}, the budget safeguard (Alg. 1) makes the final selection independent of this outcome.

Representative token per voxel. To ensure one token per occupied voxel, we retain the token nearest the mean of the others in its voxel, which sits near the geometric center of the voxel's occupancy and is the most representative proxy. A boundary token would risk placing neighboring voxels' representatives close together, undermining uniformity:

$$i_k^{\text{rep}} = \arg\min_{i \in \mathcal{V}_k} \left\| \mathbf{t}_i - \frac{1}{|\mathcal{V}_k| - 1} \sum_{j \in \mathcal{V}_k,\, j \neq i} \mathbf{t}_j \right\|_2 \tag{3}$$

The output of this stage is the selected set:

$$|\mathcal{C}_{init}| = \min(G(v_s; X),\ B) \tag{4}$$

3.5. Coverage Expansion

Since $G(v_s; X)$ is capped by the number of distinct token locations, stage 1 alone cannot reach the full budget. Stage 2 lifts the ceiling by expanding $\mathcal{C}_{\text{init}}$ with an expansion set $\mathcal{C}_{\text{expan}}$ of size B_{expan} via farthest point sampling (FPS) [15], an inherently coverage-seeking procedure. FPS iteratively builds the expansion set by selecting candidate tokens that are maximally distant from all currently selected tokens. Starting from an empty set $\mathcal{C}_{\text{expan}}$, it treats running selection as $\mathcal{C} = \mathcal{C}_{init} \cup \mathcal{C}_{\text{expan}}$. At each iteration, new token s^* maximizes its spatial distance to nearest token already in $\mathcal{C}$:

$$m^* = \arg\max_{m \notin \mathcal{C}} \left(\min_{s \in \mathcal{C}} \mathcal{D}(\mathbf{t}_m, \mathbf{t}_s) \right) \tag{5}$$

Intuitively, this adds each new token m^* to $\mathcal{C}_{\text{expan}}$, i.e., wherever the scene is currently least covered, and repeats until the budget is met, ensuring $|\mathcal{C}| = B$.

Voxel-initialized expansion. Because $\mathcal{C}$ is initialized with $\mathcal{C}_{\text{init}}$, the first expansion step measures distance to the already covered regions, allowing FPS to select tokens in uncovered areas rather than re-covering regions explored in stage 1. Concretely, for every unselected token $m \notin \mathcal{C}_{\text{init}}$, the minimum distance to the initial set is computed as $d_{\min}(m) = \min_{s \in \mathcal{C}_{init}} \mathcal{D}(\mathbf{t}_m, \mathbf{t}_s)\ \forall m \notin \mathcal{C}_{init}$. At each step, the token with the largest $d_{\min}$ is added to $\mathcal{C}_{\text{expan}}$ and the distances are updated against the new token, for B_{expan} steps, yielding $|C| = B$.

Distance metric. We use the squared Euclidean distance, i.e., $\mathcal{D}_{\text{spatial}}(\mathbf{t}_i, \mathbf{t}_j) = \|\mathbf{t}_i - \mathbf{t}_j\|_2^2$ between the 3D coordinates as the FPS metric.

[1] Bounds $[v_{\min}, v_{\max}] = [0.02, 5.0]$ are fixed once by the scale of indoor scenes: below 0.02 m occupancy no longer grows (Fig. 2(d)), and 5 m yields below 0.1% retention, so the interval spans the full practical range. Any $T \geq 8$ is sufficient over this range.

3.6. Coverage Objective

Stage 2 always returns B tokens for any feasible budget $B \leq M$; this budget guarantee does not depend on the voxel search. Coverage admits a bound in the common case where the safeguard is inactive, i.e. $|\mathcal{C}_{\text{init}}| = G(v_s; X) \leq B$: every unselected token then shares a voxel of side v_s with a selected one and lies within its space diagonal, yielding $d_H(X, \mathcal{C}_{\text{init}}) \leq \sqrt{3}\, v_s$. Each FPS step then selects the token attaining the inner max–min of Eq. 1, so $d_H(X, \mathcal{C})$ is non-increasing through stage 2 (the squared Euclidean distance leaves the selection order unchanged) and the bound carries to the final selection. When the safeguard is active, the exact budget guarantee still holds, but this particular bound may no longer apply; full proofs are in the Appendix.

4. Experiments and Results

Benchmarks and Metrics. We evaluate all three forms of 3D reasoning: ScanQA [2] for 3D spatial understanding, SQA3D [40] for situated reasoning grounded in an agent's position, and OpenEQA [42] for open-vocabulary embodied QA. Together, they probe object recognition, attributes, counting, localization, and spatial relations. Following prior work [2, 21, 24, 33, 34, 42, 83], we report EM@1, CIDEr [56], and ROUGE-L [35] for ScanQA; EM@1 for SQA3D; and LLM-Match for OpenEQA. Efficiency is measured by inference and pruning time (s), LLM TFLOPs, KV-cache (MB), and peak GPU memory (GB).

Models and protocol. We test CoVeR with four VLMs: `LLaVA-OV-7B` [32], `Video-3D LLM` [82], `Qwen2.5-VL-7B` [4], and `Qwen3-VL-8B` [3]. Following [24, 34], we uniformly sample 12 views, evaluate prior work retention ratios, and compare with VTC [24], DTC [24], VisPruner [75], and SeGPruner [34] under matched inputs. For Geo3DPruner [33], we follow its protocol with `Video-3D LLM` using 16 views. [2] CoVeR uses $\alpha = 0.4$ throughout; all experiments run on one NVIDIA H100 GPU. Additional details are in the Appendix.

4.1. Main Results

CoVeR achieves the best aggregate performance at every token budget (Table 2). Averaging across three datasets, CoVeR retains 93.5% of full-token performance at 8% token retention, versus 89.6% and 85.9% for SeGPruner and VisPruner. On ScanQA, it improves over the full-token baseline at 23% retention, reaching 28.5 EM@1 and 85.5 CIDEr, while at 9% budget it achieves 27.1 EM@1, 81.4 CIDEr, and 41.5 ROUGE-L, substantially outperforming SeGPruner and VisPruner. It also reaches 48.6 EM@1 on SQA3D at 8% retention and outperforms prior pruning methods on OpenEQA at aggressive budgets. We further report category-level OpenEQA results and comparisons in the Appendix. Fig. 3 shows CoVeR spreading tokens across all chairs and answering correctly, while VisPruner and SeGPruner cluster on a few patches and undercount.

[2] Official code for VTC/ DTC and Geo3DPruner are not publicly available; we follow their reported protocol for a direct comparison.

Table 2. **Performance Comparison**. CoVeR compared to *Voxelization* and *Learned Importance* methods on 12-view ScanQA [2], OpenEQA [42], and SQA3D [40]. **Avg.** is over benchmarks, while **Rel.** is avg. % of performance maintained. Higher is better.

Methods	ScanQA			OpenEQA	SQA3D	Avg.	Rel.
	EM@1	CIDEr	ROUGE-L	L-Match	EM@1		
Retain 100% Tokens							
`LLaVA-OV-7B`	28.2	83.6	42.6	59.1	51.7	54.1	100.0
Retain 54% Tokens				Retain 56% Tokens			
DTC [24]	27.8	–	–	–	–	–	–
VisPruner [75]	27.7	80.5	41.3	59.1	50.9	53.3	98.5
SeGPruner [34]	28.5	83.8	42.6	**58.9**	**51.7**	54.1	100.0
CoVeR (Ours)	**28.7**	**85.0**	**43.2**	**58.9**	**51.7**	**54.3**	**100.4**
Retain 40% Tokens				Retain 43% Tokens			
DTC [24]	27.7	–	–	–	–	–	–
VisPruner [75]	28.0	80.3	41.4	58.4	51.0	53.1	98.3
SeGPruner [34]	28.2	81.9	42.0	58.0	51.5	53.4	98.9
CoVeR (Ours)	**28.9**	**85.5**	**43.4**	**58.6**	**51.7**	**54.3**	**100.5**
Retain 23% Tokens				Retain 26% Tokens			
DTC [24]	27.7	–	–	–	–	–	–
VisPruner [75]	26.9	77.1	40.1	57.1	49.5	51.5	95.4
SeGPruner [34]	27.7	78.9	40.7	57.5	50.6	52.4	97.1
CoVeR (Ours)	**28.5**	**85.5**	**43.2**	**57.7**	**51.7**	**53.9**	**99.7**
Retain 14% Tokens				Retain 17% Tokens			
DTC [24]	26.7	–	–	–	–	–	–
VisPruner [75]	24.8	71.7	37.5	55.9	49.0	49.9	92.2
SeGPruner [34]	26.4	75.2	38.9	56.0	49.7	50.8	94.2
CoVeR (Ours)	**27.9**	**82.4**	**42.2**	**56.8**	**51.2**	**52.9**	**98.0**
Retain 9% Tokens				Retain 8% Tokens			
DTC [24]	26.1	–	–	–	48.0	–	–
VisPruner [75]	23.4	66.9	35.6	51.5	45.7	46.4	85.9
SeGPruner [34]	24.5	71.2	37.0	52.5	48.4	48.4	89.6
CoVeR (Ours)	**27.1**	**81.4**	**41.5**	**53.0**	**48.6**	**50.5**	**93.5**

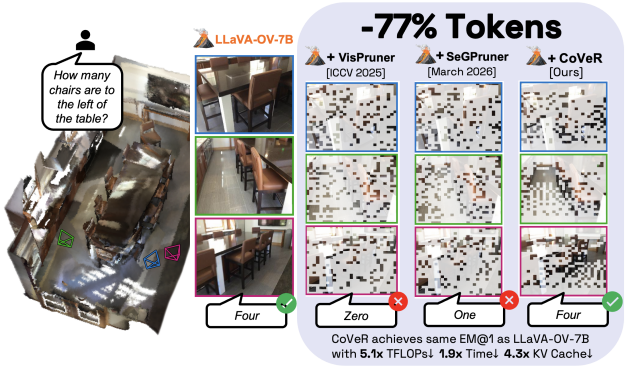


Figure 3. **Qualitative comparisons.** CoVeR provides correct answers while substantially reducing the computational cost.

4.2. Geometric Coverage Analysis

Does CoVeR improve Geometric Coverage? Accuracy alone does not reveal how a pruning method should spend its target budget. We therefore measure coverage at the most aggressive ScanQA budget with two metrics. The *Nearest Neighbor Index* (NNI) [12] measures how evenly the retained tokens are spread: it is the ratio of their mean nearest-neighbor distance to the value expected under a spatially random process, so a low value flags the clustering we want to avoid. A spread-out set can still miss whole regions, so the *Nearest Neighbor Distance* (NND_q) measures coverage of the scene directly: one minus the q-th percentile of every original token's distance to its nearest selected token, normalized by the scene diagonal. We report NND_{95} for robustness and NND_{100}, the worst case, which equals the normalized complement of the directed Hausdorff distance minimized by CoVeR (Eq. 1). Higher indicates better coverage. Details are included in the Appendix.

As shown in Table 3, SeGPruner concentrates tokens on a few regions ($\mathrm{NNI} = 0.458$), while CoVeR is far more uniform ($\mathrm{NNI} = 0.924$). CoVeR also covers the scene better on both NND_{95} (0.980 vs. 0.967) and NND_{100} (0.977 vs. 0.917), with the largest gap on the worst case. All gaps are significant under the paired t- and Wilcoxon signed-rank tests. CoVeR also attains higher accuracy than SeGPruner, supporting our motivation that, at a fixed budget, spatial coverage beats concentrating tokens on a few regions. The widest gap falls on NND_{100}, the normalized complement of the directed Hausdorff distance CoVeR minimizes, confirming it as the right objective for coverage-based selection.

Table 3. **Coverage and Performance**. CoVeR achieves substantially better spatial coverage with higher downstream 3D scene understanding performance compared to SeGPruner [34]. Higher ↑ is better on all columns.

Method	Coverage			Performance		
	NNI	NND_{95}	NND_{100}	EM@1	CIDEr	ROUGE-L
SeGPruner	0.458	0.967	0.917	24.5	71.2	37.0
CoVeR	**0.924**	**0.980**	**0.977**	**27.1**	**81.4**	**41.5**

Does CoVeR preserve informative regions? A geometry-only selector raises a concern: broad coverage might come from visually uninformative regions while discarding answer-critical evidence. We therefore compare CoVeR and SeGPruner selections directly at 9% ScanQA retention with Token Recovery (TR) and Token Expansion (TE). TR measures the normalized distance from each SeGPruner token to its nearest CoVeR token, while TE is the reverse direction; both are defined in the supplementary materials. A small TR means CoVeR keeps a token near every region SeGPruner selects, while a large TE means CoVeR also covers regions SeGPruner ignores.

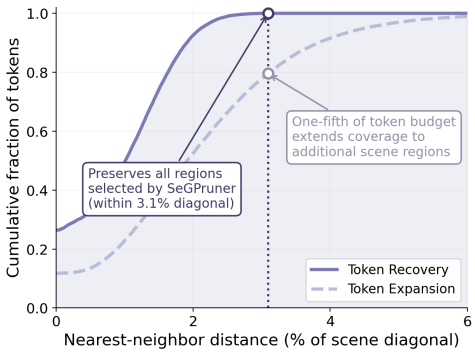


Figure 4. **Cumulative distributions of directed nearest-neighbors**. CoVeR preserves SeGPruner's selected regions within 3.1% of scene diagonal while covering ≈20% additional regions.

Table 4. **Directed distance**. CoVeR remains close to regions selected by SeGPruner while expanding into additional regions.

Comparison	TR ↓	TE ↑
CoVeR vs. SeGPruner	**0.009**	**0.020**

Table 4 shows TR is small (0.009 of the scene diagonal) and TE about twice as large (0.020), so CoVeR's tokens stay close to SeGPruner's while the reverse does not hold. The cumulative distributions (Fig. 4) make this sharper: every SeGPruner token lies within 3.1% of the scene diagonal of a CoVeR token, while roughly 20% of CoVeR tokens remain farther than that from any SeGPruner token. Thus, coverage preserves salient regions while additionally covering the rest of the scene.

Insight 1: Broader spatial coverage accompanies stronger 3D reasoning. Prioritizing coverage does not exclude informative regions but instead preserves them, extending coverage to additional parts of the scene.

4.3. Efficiency and Generalization

Efficiency. Relative to `LLaVA-OV-7B` (Table 5), CoVeR delivers substantial efficiency gains while largely preserving performance. At the most aggressive 9% budget, it achieves 13.3× fewer TFLOPs, 10.7× smaller KV cache, and a 2.9× speedup for a 1.1-point drop in performance. Against the *learned importance* pruners, CoVeR also pairs the highest performance with the lowest peak GPU memory (Table 6): they must run the encoder with attention outputs enabled and hold the attention maps to rank tokens, whereas CoVeR selects from 3D coordinates alone, with only the negligible overhead of a binary search plus FPS.

Additional Backbones. Under Geo3DPruner's [33] 16-view, 10% retention protocol on `Video-3D LLM` [82], CoVeR obtains 26.5 ScanQA EM@1 versus 26.0 for Geo3DPruner,

and retains 93.5% of full performance versus 90.7% (Table 7), even though Geo3DPruner adds a `VGGT-1B` encoder [58] and fully retrains the backbone. Without changing the selection rule or α, CoVeR also transfers to `Qwen2.5-VL-7B` and `Qwen3-VL-8B`, which differ substantially in visual encoders, tokenization, and resolution handling. Both retain over 96% of ScanQA performance at retention levels above 20% (Fig. 5(a)) and over 95% on SQA3D until retention falls below 20% (Fig. 5(b)), matching trends across architectures.

Table 5. **Efficiency at varying token retention.** 'Pruning' denotes the average time which CoVeR needs to select tokens, while 'Time' reports end-to-end inference latency (in sec). CoVeR substantially reduces TFLOPs, KV cache (MB), memory (GB) while remaining competitive across token budgets. Results are relative to `LLaVA-OV-7B` on ScanQA.

Tokens Retained	Efficiency					Accuracy	
	Pruning ↓	Time ↓	TFLOPs ↓	KV ↓	Mem ↓	EM@1 ↑	Δ
100%	–	0.497	145.5	480.0	24.1	28.2	–
54%	0.189	0.493 1.0×	71.1 2.0×	259.9 1.8×	20.3 1.2×	28.7	+0.5
40%	0.141	0.388 1.3×	51.1 2.8×	192.9 2.5×	19.2 1.3×	28.9	+0.7
23%	0.082	0.268 1.9×	28.3 5.1×	111.7 4.3×	17.8 1.4×	28.5	+0.3
14%	0.049	0.202 2.5×	17.0 8.6×	68.6 7.0×	**17.2** 1.4×	27.9	−0.3
9%	**0.034**	**0.174** 2.9×	**10.9** 13.3×	**44.7** 10.7×	**17.2** 1.4×	27.1	−1.1

Table 6. **Efficiency comparison.** CoVeR attains the highest accuracy at the lowest peak memory. Its selection cost remains a fraction of end-to-end inference. Scores report pruning on `LLaVA-OV-7B` at 9% token retention for ScanQA.

Methods	EM@1 ↑	CIDEr ↑	ROUGE-L ↑	Pruning (s) ↓	Mem (GB) ↓
VisPruner [75]	23.4	66.9	35.6	0.010	22.1
SeGPruner [34]	24.5	71.2	37.0	**0.008**	22.1
CoVeR	**27.1**	**81.4**	**41.5**	0.034	**17.2**

Table 7. **Generalization with `Video-3D LLM` [82] backbone.** Geo3DPruner [33] introduces a `VGGT-1B` [58] encoder and requires full backbone retraining, while CoVeR is training-free. Scores are EM@1 ↑ on 16-view at 10% budget.

Methods	Encoder	Retrain	ScanQA		SQA3D		Rel.(%)
			100%	10%	100%	10%	
Geo3DPruner [33]	`VGGT1B`	Full	29.7	26.0	59.3	55.7	90.7
CoVeR	None	None	28.9	26.5	57.9	55.1	**93.5**

4.4. Design Ablations

We ablate CoVeR one component at a time. Full details have been provided in the Appendix.

Stage 1: Preserving encoder-native tokens. Fig. 6 shows that the advantage of keeping a real token (pruning) over averaging features within a voxel (merging) widens with stronger compression: at the tightest budget, pruning improves EM@1 by 7.7/18.1 on ScanQA/SQA3D, with the largest SQA3D gains on `Can` (+37) and `Which` (+29), which need spatial grounding. Coarse voxels mix objects, surfaces, and viewpoints, so averaging yields a synthetic feature, whereas CoVeR keeps an encoder-native token with a valid spatial identity. CoVeR also beats DTC [24] at every budget, so the gain is not merely voxel-size selection.

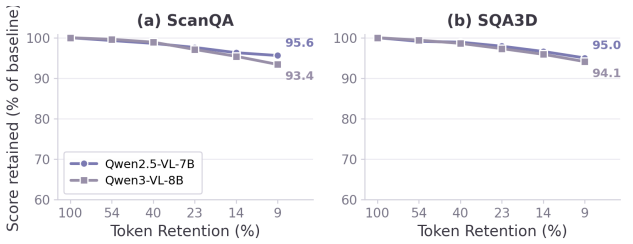


Figure 5. **CoVeR transfers across VLMs.** CoVeR exhibits similar performance on `Qwen2.5-VL-7B` and `Qwen3-VL-8B`. Performance remains above 93% of the baseline even at 9% token retention for both (a) ScanQA and (b) SQA3D.

Insight 2. Original encoder tokens are easier for LLM to interpret than synthetic avg. of heterogeneous regions.

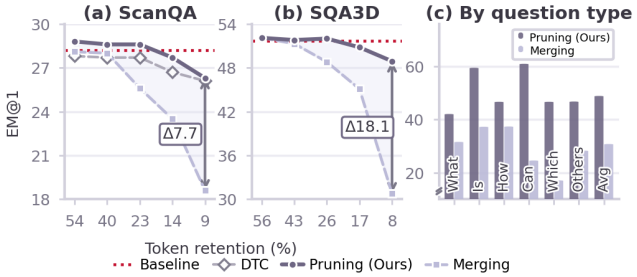


Figure 6. **Coverage initialization analysis** ($\alpha = 1$). Pruning consistently outperforms merging, with gap widening at tight budgets.

Stage 2: Geometric distance. Fig. 7 shows pure 3D distance is consistently strongest across budgets. At the tightest budget, it beats spatial+semantic FPS by 1.7/2.1 on ScanQA/SQA3D and semantic-only FPS by 2.9/4.3, with the largest SQA3D gains on `How` (+6.8) and `What` (+6.2), which require counting or localizing multiple objects. Semantic FPS suppresses distinct objects with similar embeddings, while spatial distance keeps them when 3D locations differ. Without attention or features, it still matches or exceeds SeGPruner [34].

Insight 3. Spatial distance preserves visually similar instances at different locations, while semantic distance can incorrectly suppress them.

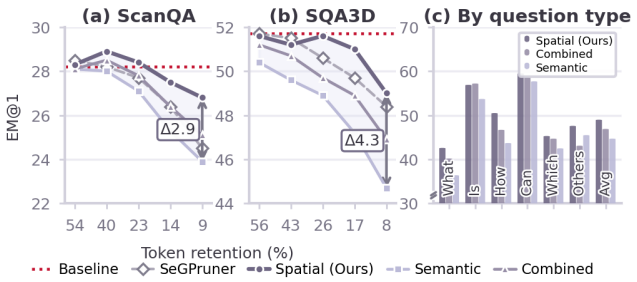


Figure 7. **Coverage expansion analysis** ($\alpha = 0$). Spatial distance outperforms combined and semantic alternatives, with gap widening at tight budgets.

CoVeR ablation. Table 8 ablates the expansion rule and its initialization. Our iterative FPS updates the minimum distance after each pick; we compare it against *Top-K selection* (selecting tokens farthest from the stage 1 tokens without distance updates) and *Random selection* (uniform sampling). Both are weaker: at 9% retention they lose 0.7 and 1.7 average ScanQA points, since only iterative updates keep placing tokens in still-uncovered regions. For initialization, seeding expansion from the stage 1 tokens rather than *From scratch* adds 1.1, 0.2, and 0.5 points at 23%, 14%, and 9% retention, because voxel seeding removes duplicates in parallel first, so expansion extends into new regions instead of rediscovering covered ones.

Coverage initialization is necessary. Table 8 isolates the contribution of stage 1 by comparing CoVeR with FPS only design. Without voxel initialization, FPS must build scene coverage sequentially from scratch.

Table 8. **Design ablations.** Iterative FPS with stage 1 initialization consistently performs best. Results on ScanQA, averaged over EM@1, CIDEr, and ROUGE-L. Pruning time is reported in sec.

Component	Setting	Token Budget		
		23%	14%	9%
Expansion	w/o Iterative FPS (Top-K)	51.8	50.4	49.3
	w/o Iterative FPS (Random)	51.8	50.6	48.3
	w/ Iterative FPS	**52.4**	**50.8**	**50.0**
Initialization	w/o Stage 1 seed (From scratch)	51.3	50.6	49.5
	w/ Stage 1 seed	**52.4**	**50.8**	**50.0**
Pruning time	w/o Stage 1 (FPS only)	0.126	0.078	0.049
	w/ Stage 1 (CoVeR)	**0.082**	**0.049**	**0.034**

Seeding FPS with one representative per occupied voxel instead provides a coarse coverage from the start, which reduces pruning time by approximately $1.5\times$. Across all variants above, higher coverage (NNI, NND_{95}, NND_{100}) accompanies higher accuracy, with full CoVeR best on every measure (Fig. 8). Weaker variants leave coherent regions uncovered (top-K, random) or revisit initialized voxels (from-scratch FPS). This consistent ranking supports coverage as the mechanism behind the downstream gains.

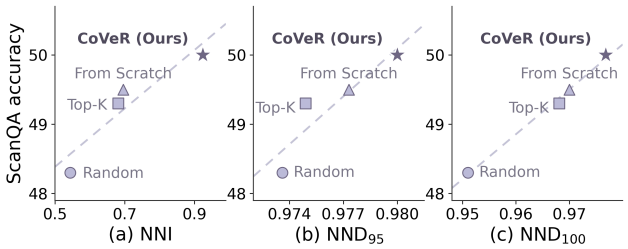
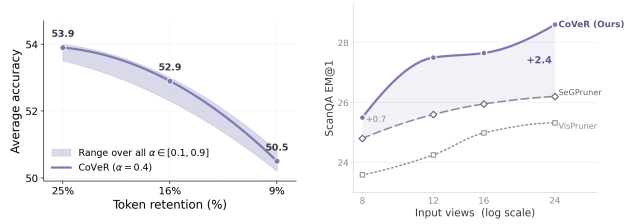


Figure 8. **Better coverage achieves higher performance.** Correlation between coverage and accuracy (on ScanQA at 9% retention).

Voxel ratio. Varying α from 0.1 to 0.9 changes accuracy within a narrow band (Fig. 9a), so the split between the two stages is not narrowly tuned; we use $\alpha = 0.4$ throughout.

Scaling with views. CoVeR leads VisPruner and SeGPruner at every view count and gains more from added views (Fig. 1 and Fig. 9b), because coverage reallocates the fixed budget to newly visible regions rather than repeatedly observed regions favored by attention.



(a) **Voxel ratio.** Avg. accuracy across ScanQA, SQA3D, OpenEQA remains stable across budgets.

(b) **Effect of View count.** CoVeR improves with more views, while VisPruner and SeGPruner saturate.

Figure 9. **Ablation for voxel ratio and scaling view count.**

5. Conclusion

We propose CoVeR, a training-free, deterministic, geometry-only framework for reducing the visual token count of multi-view 2D VLMs. Our analysis identifies key limitations in each prior family: *voxelization*-based methods cannot enforce exact per-scene budgets and are capped by voxel saturation, while *learned importance*-based methods concentrate on prominent regions rather than scene coverage. CoVeR addresses both: it searches per scene for the voxel size that yields an initial selection, then extends beyond the saturation plateau using spatial distance alone. It achieves SOTA on three benchmarks while generalizing across four VLMs from two model families without retraining.

Limitations. Like all prior methods, CoVeR requires depth and camera, and is designed for indoor scenes. Its performance may therefore depend on the quality of the estimated geometry. Future work can combine coverage with reliable depth/pose estimation, and hierarchical or streaming selection for outdoor scenes.

Acknowledgments

The authors thank Saswat Subhajyoti Mallick, Surgan Jandial, Nicholas Mesa-Cucalon, and Yinong Oliver Wang for their insightful discussions, feedback, and assistance with parts of the project. Aviral Chharia was supported in part by the Uber Presidential Fellowship from the Robotics Institute, Carnegie Mellon University. The computational resources were supported in part by PSC Bridges-2 through the Advanced Cyberinfrastructure Coordination Ecosystem: Services and Support (ACCESS) program allocation CIS250962, which is supported by National Science Foundation (NSF) grants #2138259, #2138286, #2138307, #2137603, and #2138296.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 13, 16, 18

[2] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19107–19117, 2022. 2, 5, 13, 18, 19

[3] 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. 2, 5, 14

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. 2, 5, 13, 14, 18

[5] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[6] Wenhao Chai, Enxin Song, Yilun Du, Chenlin Meng, Vashisht Madhavan, Omer Bar-Tal, Jenq-Neng Hwang, Saining Xie, and Christopher D Manning. Auroracap: Efficient, performant video detailed captioning and a new benchmark. In *The Thirteenth International Conference on Learning Representations*, 2025. 18

[7] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2024. 2

[8] Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. Ll3da: Visual interactive instruction tuning for omni-3d understanding, reasoning, and planning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26418–26428, 2024. 2

[9] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 18

[10] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. 18

[11] Jiho Choi, Seonho Lee, Seojeong Park, and Hyunjung Shim. Dense reward for multi-view 3d reasoning with global maps and local views. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026. 2

[12] Philip J. Clark and Francis C. Evans. Distance to nearest neighbor as a measure of spatial relationships in populations. *Ecology*, 35(4):445–453, 1954. 6

[13] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2443, 2017. 13

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 13

[15] Y. Eldar, M. Lindenbaum, M. Porat, and Y.Y. Zeevi. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing*, 6(9):1305–1315, 1997. 3, 4

[16] Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Peihao 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 31054–31065, 2026. 2

[17] Rao Fu, Jingyu Liu, Xilun Chen, Yixin Nie, and Wenhan Xiong. Scene-llm: Extending language model for 3d visual reasoning. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2195–2206, 2025. 2, 18

[18] Ziyu Guo, Renrui Zhang, Xiangyang Zhu, Yiwen Tang, Xianzheng Ma, Jiaming Han, Kexin Chen, Peng Gao, Xianzhi Li, Hongsheng Li, et al. Point-bind & point-llm: Aligning point cloud with multi-modality for 3d understanding, generation, and instruction following. *arXiv preprint arXiv:2309.00615*, 2023. 2

[19] Chanyoung Gwak, Yoonwoo Jeong, Byungwoo Jeon, Hyunseok Lee, Jinwoo Shin, and Minsu Cho. Cog3dmap: Multi-view vision-language reasoning with 3d cognitive maps. *arXiv preprint arXiv:2603.23023*, 2026. 2

[20] Yining Hong, Chunru Lin, Yilun Du, Zhenfang Chen, Joshua B. Tenenbaum, and Chuang Gan. 3d concept learning and reasoning from multi-view images. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9202–9212, 2023. 2

[21] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023. 2, 5, 18

[22] Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. *Advances in Neural Information Processing Systems*, 38:82782–82802, 2026. 18

[23] Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, et al. Chat-scene: Bridging 3d scene and large language models with object identifiers. *Advances in Neural Information Processing Systems*, 37:113991–114017, 2024. 2, 18

[24] Hsiang-Wei Huang, Fu-Chen Chen, Wenhao Chai, Che-Chun Su, Lu Xia, Sanghun Jung, Cheng-Yen Yang, Jenq-Neng Hwang, Min Sun, and Cheng-Hao Kuo. Zero-shot 3d question answering via voxel-based dynamic token compression. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19424–19434, 2025. 2, 5, 7, 13, 15, 16

[25] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024. 2, 18

[26] Ting Huang, Zeyu Zhang, and Hao Tang. 3d-r1: Enhancing reasoning in 3d vlms for unified scene understanding. *arXiv preprint arXiv:2507.23478*, 2025. 2

[27] Xiaohu Huang, Jingjing Wu, Qunyi Xie, and Kai Han. 3drs: Mllms need 3d-aware representation supervision for scene understanding. *Advances in Neural Information Processing Systems*, 38:67961–67988, 2026. 2

[28] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 13, 17

[29] Jerry Jiang, Haowen Sun, Denis Gudovskiy, Yohei Nakata, Tomoyuki Okuno, Kurt Keutzer, and Wenzhao Zheng. Proxy3d: Efficient 3d representations for vision-language models via semantic clustering and alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23816–23825, 2026. 18

[30] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13700–13710, 2024. 18

[31] 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. 18

[32] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025. 2, 5, 13

[33] Han Li, Zehao Huang, Jiahui Fu, Naiyan Wang, and Si Liu. Geometry-guided 3d visual token pruning for video-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9615–9625, 2026. 2, 3, 5, 6, 7

[34] Wenli Li, Kai Zhao, Haoran Jiang, Enquan Yang, Yi Su, and Dan Zeng. Segpruner: Semantic-geometric visual token pruner for 3d question answering, 2026. 2, 5, 6, 7, 13, 15, 17

[35] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, 2004. Association for Computational Linguistics. 5, 13

[36] Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5334–5342, 2025. 2

[37] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llavanext: Improved reasoning, ocr, and world knowledge, 2024. 18

[38] Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. Nvila: Efficient frontier visual language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4122–4134. IEEE, 2025. 18

[39] Jingzhou Luo, Yang Liu, Weixing Chen, Zhen Li, Yaowei Wang, Guanbin Li, and Liang Lin. Dspnet: Dual-vision scene perception for robust 3d question answering. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14169–14178, 2025. 18

[40] Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. SQA3d: Situated question answering in 3d scenes. In *The Eleventh International Conference on Learning Representations*, 2023. 2, 5, 13, 20

[41] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *ACL*, 2024. 18

[42] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, Karmesh Yadav, Qiyang Li, Ben Newman, Mohit Sharma, Vincent Berges, Shiqi Zhang, Pulkit Agrawal, Yonatan Bisk, Dhruv Batra, Mrinal Kalakrishnan, Franziska Meier, Chris Paxton, Alexander Sax, and Aravind Rajeswaran. Openeqa: Embodied question answering in the era of foundation models. In *2024 IEEE/CVF Conference on Computer Vision and Pat-*

tern Recognition (CVPR), pages 16488–16498, 2024. 2, 5, 13, 14, 18, 21

[43] Yunze Man, Liang-Yan Gui, and Yu-Xiong Wang. Situational awareness matters in 3d vision language reasoning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13678–13688, 2024. 2

[44] Zhangyang Qi, Ye Fang, Zeyi Sun, Xiaoyang Wu, Tong Wu, Jiaqi Wang, Dahua Lin, and Hengshuang Zhao. Gpt4point: A unified framework for point-language understanding and generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26407–26417, 2024. 2

[45] 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. 2

[46] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *NeurIPS Datasets and Benchmarks Track*, 2021. 13

[47] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22857–22867, 2025. 2

[48] Dingjie Song, Wenjun Wang, Shunian Chen, Xidong Wang, Michael X. Guan, and Benyou Wang. Less is more: A simple yet effective token reduction method for efficient multi-modal LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7614–7623, Abu Dhabi, UAE, 2025. Association for Computational Linguistics. 2

[49] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. Moviechat: From dense token to sparse memory for long video understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18221–18232, 2024. 18

[50] Yuan Tang, Xu Han, Xianzhi Li, Qiao Yu, Yixue Hao, Long Hu, and Min Chen. Minigpt-3d: Efficiently aligning 3d point clouds with large language models using 2d priors. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6617–6626, 2024. 2

[51] Yuan Tang, Xu Han, Xianzhi Li, Qiao Yu, Jinfeng Xu, Yixue Hao, Long Hu, and Min Chen. More text, less point: Towards 3d data-efficient point-language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7284–7292, 2025. 2

[52] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 18

[53] Anh Thai, Songyou Peng, Kyle Genova, Leonidas Guibas, and Thomas Funkhouser. Splattalk: 3d vqa with gaussian splatting. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4712–4721, 2025. 2, 18

[54] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 18

[55] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 14

[56] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575, 2015. 5, 13

[57] Fengyun Wang, Sicheng Yu, Jiawei Wu, Jinhui Tang, Hanwang Zhang, and Qianru Sun. 3d question answering via only 2d vision-language models. In *International Conference on Machine Learning*, pages 65310–65325. PMLR, 2025. 2

[58] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5294–5306, 2025. 3, 7

[59] Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. *Advances in neural information processing systems*, 38:13569–13597, 2026. 18

[60] Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024. 2

[61] Haomiao Xiong, Yunzhi Zhuge, Jiawen Zhu, Lu Zhang, and Huchuan Lu. 3ur-llm: An end-to-end multimodal large language model for 3d scene understanding. *IEEE Transactions on Multimedia*, 27:2899–2911, 2025. 2

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

[63] Rongtao Xu, Han Gao, Mingming Yu, Dong An, Shunpeng Chen, Changwei Wang, Li Guo, Xiaodan Liang, and Shibiao Xu. 3d-more: Unified modal-contextual reasoning for embodied question answering. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5924–5929, 2025. 2

[64] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, et al. Qwen2 technical report, 2024. 13

[65] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang,

Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 14

[66] Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, Yuquan Deng, and Jianfeng Gao. Magma: A foundation model for multimodal ai agents. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14203–14214, 2025. 18

[67] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19792–19802, 2025. 2

[68] Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 22128–22136, 2025. 2

[69] Hanxun Yu, Wentong Li, Song Wang, Junbo Chen, and Jianke Zhu. Inst3d-lmm: Instance-aware 3d scene understanding with multi-modal instruction tuning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14147–14157, 2025. 2, 18

[70] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6274–6283, 2019. 18

[71] Zhihao Yuan, Yibo Peng, Jinke Ren, Yinghong Liao, Yatong Han, Chun-Mei Feng, Hengshuang Zhao, Guanbin Li, Shuguang Cui, and Zhen Li. Empowering large language models with 3d situation awareness. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19435–19445, 2025. 18

[72] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11941–11952, 2023. 13

[73] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 conference on empirical methods in natural language processing: system demonstrations*, pages 543–553, 2023. 18

[74] Jiahui Zhang, Yurui Chen, Yueming Xu, Ze Huang, Jilin Mei, Chunhui Chen, Yanpeng Zhou, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, et al. From flatland to space: Teaching vision-language models to perceive and reason in 3d. *Advances in Neural Information Processing Systems*, 38, 2026. 18

[75] Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20857–20867, 2025. 2, 3, 5, 7, 13, 17

[76] Sha Zhang, Di Huang, Jiajun Deng, Shixiang Tang, Wanli Ouyang, Tong He, and Yanyong Zhang. Agent3d-zero: An agent for zero-shot 3d understanding. In *European Conference on Computer Vision*, pages 186–202. Springer, 2024. 18

[77] Taolin Zhang, Sunan He, Tao Dai, Zhi Wang, Bin Chen, and Shu-Tao Xia. Vision-language pre-training with object contrastive learning for 3d scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7296–7304, 2024. 18

[78] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *Forty-second International Conference on Machine Learning*, 2025. 2

[79] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun MA, Ziwei Liu, and Chunyuan Li. LLaVA-video: Video instruction tuning with synthetic data. *Transactions on Machine Learning Research*, 2025. 14

[80] Lichen Zhao, Daigang Cai, Jing Zhang, Lu Sheng, Dong Xu, Rui Zheng, Yinjie Zhao, Lipeng Wang, and Xibo Fan. Toward explainable 3d grounded visual question answering: A new benchmark and strong baseline. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(6):2935–2949, 2023. 18

[81] Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. Towards learning a generalist model for embodied navigation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13624–13634, 2024. 18

[82] Duo Zheng, Shijia Huang, and Liwei Wang. Video-3d llm: Learning position-aware video representation for 3d scene understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8995–9006, 2025. 2, 5, 6, 7, 14

[83] Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. Llava-3d: A simple yet effective pathway to empowering lmms with 3d capabilities. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4295–4305, 2025. 2, 5, 18

[84] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. 18

[85] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2899–2909, 2023. 18

CoVeR: Coverage-Based Token Pruning for Multi-View 3D Reasoning in VLMs

Supplementary Material

A. Implementation Details

A.1. Benchmarks

To demonstrate generalization across diverse 3D scene understanding tasks, we evaluate CoVeR on benchmarks covering spatial scene understanding (ScanQA [2]), situated reasoning (SQA3D [40]), and embodied question answering (OpenEQA [42]).

ScanQA [2] and SQA3D [40] are both based on the scenes from the ScanNet [13] dataset. ScanQA requires models to answer questions related to spatial understanding in a 3D environment, while SQA3D requires models to have situated reasoning awareness, in which agents must determine their position and orientation in order to correctly answer the questions. Following previous works [24, 34], we include results on the validation set of ScanQA and the test set of SQA3D.

We additionally show results on the OpenEQA benchmark. OpenEQA [42] is the first open-vocabulary benchmark for embodied question answering (EQA), built on ScanNet [13] and HM3D [46] datasets. The benchmark includes seven test aspects: object recognition, attribute recognition, object state recognition, object localization, spatial reasoning, functional reasoning, and world knowledge. Together, these datasets demonstrate that CoVeR generalizes to 3D spatial understanding. Additional details for these datasets are included in Table A1 and A2.

Table A1. Number of questions and scenes in each benchmark.

Benchmark	# of questions	# of scenes
ScanQA [2]	4,306	71
OpenEQA [42]	1,636	152
SQA3D [40]	3,519	67

Table A2. Number of questions and scenes of ScanNet [13] and HM3D [46] subsets in the OpenEQA [42] benchmark.

Subset	# of questions	# of scenes
ScanNet [13]	1,079	89
HM3D [46]	557	63
Total	1,636	152

A.2. Baselines

To ensure a fair comparison, we reproduce the results using the official implementations of VisPruner [75] and SeG-Pruner [34] in the same environment, with the default importance ratio set to 0.5 (i.e., half of the selected tokens are chosen by their attention-based importance scores, while the remaining tokens are selected by the second stage of each method).

A.3. Evaluation Metrics

EM@1 (Exact Match at top-1) is a binary string matching metric. It gives a score of 1 when the prediction matches the ground truth exactly, and 0 otherwise.

CIDEr [56] measures similarity between the prediction and multiple reference answers for the same question using Term Frequency Inverse Document Frequency (TF-IDF) weighted n-gram matching, where higher scores indicate better alignment with ground truth. By default, n-gram is set to 4.

ROUGE-L [35] measures the overlap between the prediction and the ground truth using Longest Common Subsequence (LCS), rewarding answers that preserve the word order of the ground truth.

LLM-Match [42], is used to evaluate the open-ended answers from VLMs:

$$\text{LLM-Match} = \frac{1}{N}\sum_{i}^{N}\frac{\sigma_i - 1}{4} \times 100\% \tag{A1}$$

where, $\sigma_i \in \{1, \ldots, 5\}$ is the score assigned by an LLM (`GPT-4o` [28] or `GPT-4` [1] [3]). A score of 1 indicates an irrelevant answer, while 5 denotes a correct answer. We follow the official prompt template released by OpenEQA, as shown in Figure A1.

LLM TFLOPs measures the prefill computational cost of the LLM decoder. For a prompt of length n (including text tokens and retained visual tokens) processed by a L-layer transformer decoder, the prefill FLOPs are calculated as:

$$\text{FLOPs} = L\Big(\underbrace{4nd^2 + 4\,n\,d\,d_{kv}}_{\text{attention (GQA)}} + \underbrace{4n^2 d}_{\text{attention}} + \underbrace{6\,n\,d\,m}_{\text{SwiGLU FFN}}\Big) \tag{A2}$$

where d is the hidden size, m is the feed-forward network (FFN) dimension, and $d_{kv} = H_{kv}\,(d/H_q)$ denotes the key/-value width under grouped-query attention (GQA), with H_q query heads and H_{kv} key/value heads.

A.4. Backbones

LLaVA-OneVision-7B [32] consists of a visual encoder (SigLIP [72]), a projection layer (MLP), and a language model (Qwen2 [64]). The number of tokens for one input image is 729 with 384×384 resolution, resulting in 8,748 visual tokens for a 12-view setting.

Qwen2.5-VL-7B [4] contains a native dynamic resolution ViT [14] visual encoder, an MLP-based Vision-Language

[3] As `GPT-4` is being deprecated: https://developers.openai.com/api/docs/deprecations, we report both `GPT-4` (Table A4) and `GPT-4o` (Table 2 and Table A5).

Prompt Template for LLM-Match Scores

You are an AI assistant who will help me to evaluate the response given the question, the correct answer, and extra answers that are also correct. To mark a response, you should output a single integer between 1 and 5 (including 1, 5). 5 means that the response perfectly matches the answer or any of the extra answers. 1 means that the response is completely different from the answer and all of the extra answers.

Example 1:
Question: Is it overcast?
Answer: no
Extra Answers: ['doesn't look like it', 'no',' it's sunny']
Response: yes
Your mark: 1

Example 2:
Question: Who is standing at the table?
Answer: woman
Extra Answers: ['a woman', 'a lady', 'woman']
Response: Jessica
Your mark: 3

Example 3:
Question: Are there drapes to the right of the bed?
Answer: yes
Extra Answers: ['yes, there are drapes', 'yeah', 'the drapes are to the right of the king bed']
Response: yes
Your mark: 5

Your Turn:
Question: {question}
Answer: {answer}
Extra Answers: {extra_answers}
Response: {prediction}

Figure A1. Prompt used for computing LLM-Match scores on the OpenEQA [42] benchmark.

Merger projection layer, and a language model (Qwen2.5). Processing the inputs yields 391 tokens per image, resulting in 4,692 visual tokens for the 12-view setting.

Qwen3-VL-8B [3] contains a visual encoder (SigLIP2 [55]), a projection layer similar to Qwen2.5-VL [4], and a language model (Qwen3 [65]). Processing the inputs yields 300 tokens per image, resulting in 3,600 visual tokens for a 12-view setting.

Video-3D-LLM [82] extends `LLaVA-Video` [79] with sinusoidal 3D positional encoding added to the visual tokens to inject 3D camera geometry. Each frame produces 196 tokens, resulting in 3,136 visual tokens for the 16-frame setting.

A.5. Packages

To ensure the reproducibility of our results, we provide the version of all packages in Table A3.

Table A3. Packages version in the Conda environment.

Name	Version
`python`	`3.10.20`
`torch`	`2.6.0+cu124`
`torchvision`	`0.21.0+cu124`
`numpy`	`2.2.6`
`pillow`	`12.1.1`
`transformers`	`5.8.0.dev0`
`tokenizers`	`0.22.2`
`accelerate`	`1.13.0`
`safetensors`	`0.7.0`
`huggingface-hub`	`1.13.0`
`flash-attn`	`2.8.3`
`qwen-vl-utils`	`0.0.14`

A.6. Coverage Metrics

Coverage metrics. For a point $\mathbf{t}$ and a non-empty index set $\mathcal{A}$, let $\delta(\mathbf{t};\mathcal{A}) = \min_{j\in\mathcal{A}} \|\mathbf{t}-\mathbf{t}_j\|_2$ be the distance from $\mathbf{t}$ to its nearest token in $\mathcal{A}$. We quantify how well the selected set $\mathcal{C}$ covers the full-token set X with two metrics.

1. *Nearest Neighbor Index (NNI)* is the ratio of the observed mean nearest-neighbor distance among selected tokens to that expected under complete spatial randomness. As tokens lie in 3D, we use a 3D homogeneous Poisson process as the baseline:

$$\mathrm{NNI} = \frac{r_A}{r_E}, \qquad r_A = \frac{1}{N}\sum_{i=1}^{N} \min_{j\neq i} \|\mathbf{t}_i - \mathbf{t}_j\|_2, \qquad r_E = \Gamma\left(\frac{4}{3}\right)\left(\frac{4\pi}{3}\lambda\right)^{-1/3} \tag{A3}$$

   where r_A is the observed mean nearest neighbor distance among the selected tokens, r_E is its expectation under the Poisson process, and $\lambda = N/V$ is the density of N selected tokens in a scene of bounding-box volume V. $\Gamma(.)$ denotes the Gamma function.

2. *Nearest Neighbor Distance (NND)* measures scene coverage directly, since a spread-out set can still miss whole regions. For each original token we take its distance to the nearest selected token, and summarize by a percentile, normalized by the scene diagonal:

$$\mathrm{NND}_q = 1 - \frac{Q_q\big(\{\delta(\mathbf{x};\mathcal{C})\}_{\mathbf{x}\in X}\big)}{\mathrm{diag}}, \qquad \text{where } \begin{cases} q = 95: & \text{95th percentile} \\ q = 100: & \max_{\mathbf{x}\in X} \delta(\mathbf{x};\mathcal{C}) = d_H(X,\mathcal{C}) \end{cases} \tag{A4}$$

where Q_q is the q-th percentile over X. We report NND_{95}, which clips the worst 5% for robustness, and NND_{100}, the worst case, whose unnormalized numerator equals the directed Hausdorff distance $d_H(X, \mathcal{C}) = \max_{\mathbf{t} \in X} \delta(\mathbf{t}; \mathcal{C})$ that CoVeR minimizes. Higher is better for both.

Directed distances. To verify that coverage gains do not drop the salient regions chosen by *learned importance*, we compare two selections $\mathcal{C}$ and $\mathcal{S}$ in both directions, normalized by the scene diagonal.

1. *Token Recovery (TR)* checks whether $\mathcal{C}$ keeps the regions $\mathcal{S}$ selects: for each token in $\mathcal{S}$ we take δ to the nearest token in $\mathcal{C}$, then average. A small TR means $\mathcal{C}$ has a token near every region $\mathcal{S}$ selects.

$$\mathrm{TR} = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{s} \in \mathcal{S}} \frac{\delta(\mathbf{s}; \mathcal{C})}{\mathrm{diag}}, \qquad \mathrm{diag} = \sqrt{\sum_{d \in \{x,y,z\}} \left(c_d^{\max} - c_d^{\min}\right)^2} \tag{A5}$$

   where 'diag' is the diagonal of the scene's axis-aligned bounding box.

2. *Token Expansion (TE)* checks whether $\mathcal{C}$ reaches regions $\mathcal{S}$ ignores: for each token in $\mathcal{C}$ we take δ to the nearest token in $\mathcal{S}$, then average. A large TE means $\mathcal{C}$ covers regions $\mathcal{S}$ leaves out.

$$\mathrm{TE} = \frac{1}{|\mathcal{C}|} \sum_{\mathbf{c} \in \mathcal{C}} \frac{\delta(\mathbf{c}; \mathcal{S})}{\mathrm{diag}} \tag{A6}$$

A.7. Ablation Study

Stage 1: Preserving encoder-native tokens. To isolate the representation used in coverage initialization, we set $\alpha = 1$ so stage 1 alone targets the full budget $B_{\text{init}} = B$; the highest budget in this study (54%) stays within the saturation range of all ScanNet scenes (Sec. 3.2). For a fair comparison, we modify the search to ensure exactly B tokens for both strategies: rather than allowing $G(v_s; X) < B$ near saturation, we take the smallest voxel size with $G(v_s; X) \geq B$ and, if more than B voxels form, keep the B densest. This is used only for this ablation. We define:

1. *Pruning*, which keeps the representative token of each voxel (Eq. 3), so the voxel's output feature is a real encoder feature:

$$\mathbf{f}_{\mathcal{V}_k} = \mathbf{f}_{i_k^{\text{rep}}} \tag{A7}$$

2. *Merging*, following VTC [24], which averages all token features in the voxel into a pooled representation:

$$\mathbf{f}_{\mathcal{V}_k} = \frac{1}{|\mathcal{V}_k|} \sum_{i \in \mathcal{V}_k} \mathbf{f}_i \tag{A8}$$

Both yield one feature per occupied voxel on the same voxel partition; they differ only in whether that feature is a real token (*pruning*) or a synthetic average (*merging*).

Stage 2: Geometric distance. To isolate expansion from stage 1, we set $\alpha = 0$, so the stage 2 targets the full budget $B_{\text{expan}} = B$. For tokens i, j with coordinates $\mathbf{t}_i, \mathbf{t}_j$ and ℓ_2-normalized features $\hat{\mathbf{f}}_i, \hat{\mathbf{f}}_j$, we define three distances:

1. *Spatial*: squared Euclidean distance between 3D coordinates, $\mathcal{D}_{\text{spatial}} = \|\mathbf{t}_i - \mathbf{t}_j\|_2^2$.
2. *Semantic*: cosine distance between features, $\mathcal{D}_{\text{semantic}} = 1 - \hat{\mathbf{f}}_i^\top \hat{\mathbf{f}}_j$.
3. *Combined*, following SeGPruner [34], which fuses both. As the terms differ in scale, we propose the normalization for each term: the spatial term by the squared scene diagonal and the semantic term by the cosine range:

$$\mathcal{D}_{\text{combined}} = \frac{\mathcal{D}_{\text{spatial}}}{\mathrm{diag}^2} + \frac{\mathcal{D}_{\text{semantic}}}{2} \tag{A9}$$

where $c_d^{\max}$ and $c_d^{\min}$ are the scene's max/min coordinates along d.

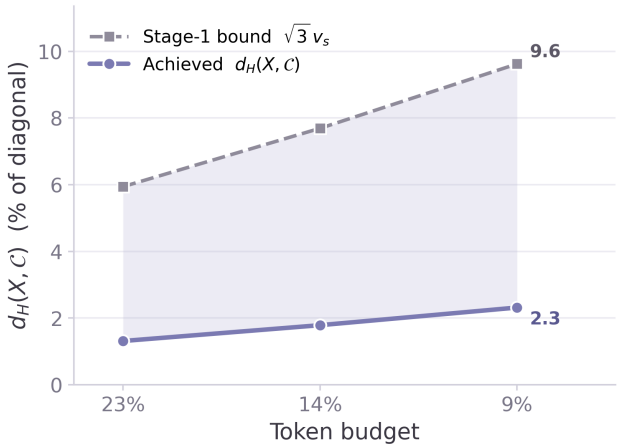


Figure A2. **Empirical validation of the coverage bound.** Even at 9% token retention, the $d_H(X, \mathcal{C})$ is only 2.3% of the scene diagonal, compared with the stage 1 bound of 9.6%.

B. Theoretical Analysis of CoVeR

Setup. Let $X = \{\mathbf{t}_i\}_{i=1}^M$ be the valid back-projected coordinates (Sec. 3.1), and $\mathcal{A} \subseteq \{1, \ldots, M\}$ a selection. Recall $\delta(\mathbf{t}; \mathcal{A})$ and $d_H(X, \mathcal{A})$ above. We have $\mathcal{C}_{\text{init}}$ for the stage 1 selection at voxel size v_s, $m_1^*, \ldots, m_{B_{\text{expan}}}^*$ for the stage 2 picks by FPS in order, $\mathcal{C}^j = \mathcal{C}_{\text{init}} \cup \{m_1^*, \ldots, m_j^*\}$ (so $\mathcal{C}^0 = \mathcal{C}_{\text{init}}$), and $\mathcal{C} = \mathcal{C}^{B_{\text{expan}}}$ for the final selection. We assume $B < M$ and $|\mathcal{C}_{\text{init}}| \leq B$ (cases 1–2 of Sec. 3.3), while case 3 is Remark 2.

Lemma 1 (Stage 1 bound)**.** $d_H(X, \mathcal{C}_{init}) \leq \sqrt{3}\, v_s$.

Proof. Each $\mathbf{t}_i$ lies in one occupied voxel $\mathcal{V}_k$, and stage 1 keeps one token $i_k^{\text{rep}} \in \mathcal{V}_k$. Both $\mathbf{t}_i$ and $\mathbf{t}_{i_k^{\text{rep}}}$ lie in a cube of side v_s, whose greatest internal distance is $\sqrt{3}\, v_s$, hence

$\delta(\mathbf{t}_i; \mathcal{C}_{\text{init}}) \le \sqrt{3}\, v_s$ for all i, and the maximum over X gives the claim. □

Lemma 2 (Stage 2 bound). *$d_H(X, \mathcal{C}^{j+1}) \le d_H(X, \mathcal{C}^j)$, hence $d_H(X, \mathcal{C}) \le d_H(X, \mathcal{C}_{init})$.*

Proof. Since $\mathcal{C}^{j+1} \supseteq \mathcal{C}^j$, the minimum in δ is over a superset, so $\delta(\mathbf{t}; \mathcal{C}^{j+1}) \le \delta(\mathbf{t}; \mathcal{C}^j)$ for all $\mathbf{t}$; the max over X preserves this, and iterating gives the second claim. □

Theorem 1 (Exact budget and coverage). *For any budget B with $B < M$ and $|\mathcal{C}_{init}| \le B$, CoVeR returns $\mathcal{C}$ with*

$$|\mathcal{C}| = B, \quad d_H(X, \mathcal{C}) \le d_H(X, \mathcal{C}_{init}) \le \sqrt{3}\, v_s$$

Proof. Stage 2 runs $B_{\text{expan}} = B - |\mathcal{C}_{\text{init}}| \ge 0$ steps. Selected indices are marked $d_{\min} = -1$, and the update $d_{\min}(m) \leftarrow \min(d_{\min}(m), \|\cdot\|_2^2)$ preserves the mark. Before step j, $|\mathcal{C}_{\text{init}}| + (j-1) \le B - 1 < M$, so an unselected index exists and the $\arg\max$ returns it. Each step adds a new token, giving $|\mathcal{C}| = |\mathcal{C}_{\text{init}}| + B_{\text{expan}} = B$.
Coverage. Chain Lemma 1 and Lemma 2. □

Lemma 3 (Greedy pick realizes d_H). *$\delta(\mathbf{t}_{m^*_{j+1}}; \mathcal{C}^j) = d_H(X, \mathcal{C}^j)$.*

Proof. Stage 2 selects $m^*_{j+1} = \arg\max_{m \notin \mathcal{C}^j} \delta(\mathbf{t}_m; \mathcal{C}^j)^2$ (i.e., $\mathcal{D}_{\text{spatial}}$); as $u \mapsto u^2$ is increasing on $[0, \infty)$, this equals the $\arg\max$ of δ. Since $\delta(\mathbf{t}; \mathcal{C}^j) = 0$ for $\mathbf{t} \in \mathcal{C}^j$, the maximum of $\delta(\cdot; \mathcal{C}^j)$ over X equals its maximum over $X \setminus \mathcal{C}^j$ whenever $d_H(X, \mathcal{C}^j) > 0$, and both equal 0 otherwise. □

Proposition 1 (2-approximation at the expansion budget). *Let $\text{OPT}_k(X) = \min_{|\mathcal{A}|=k} d_H(X, \mathcal{A})$ and $r = d_H(X, \mathcal{C})$. If $B_{expan} \ge 1$, then $d_H(X, \mathcal{C}) \le 2 \cdot \text{OPT}_{B_{expan}}(X)$.*

Proof. Let $R^j = d_H(X, \mathcal{C}^j)$; by Lemma 2, $R^0 \ge \cdots \ge R^{B_{\text{expan}}} = r$. If $r = 0$, the claim is immediate, so assume $r > 0$. For $1 \le i < j \le B_{\text{expan}}$, $m^*_i \in \mathcal{C}^{j-1}$, so by Lemma 3, $\|\mathbf{t}_{m^*_j} - \mathbf{t}_{m^*_i}\|_2 \ge \delta(\mathbf{t}_{m^*_j}; \mathcal{C}^{j-1}) = R^{j-1} \ge r$. Pick $\mathbf{x}^* \in X$ with $\delta(\mathbf{x}^*; \mathcal{C}) = r$ (unselected, since $r > 0$); it is $\ge r$ from every selected token, so $P = \{\mathbf{t}_{m^*_1}, \ldots, \mathbf{t}_{m^*_{B_{\text{expan}}}}, \mathbf{x}^*\}$ is a set of $B_{\text{expan}} + 1$ points pairwise $\ge r$ apart. Let $\mathcal{A}$ be any index set with $|\mathcal{A}| = B_{\text{expan}}$ and $\rho = d_H(X, \mathcal{A})$. Since $|P| > |\mathcal{A}|$, two points $\mathbf{p}, \mathbf{q} \in P$ share a nearest index $a \in \mathcal{A}$, so by the triangle inequality $r \le \|\mathbf{p} - \mathbf{q}\|_2 \le \|\mathbf{p} - \mathbf{t}_a\|_2 + \|\mathbf{t}_a - \mathbf{q}\|_2 \le 2\rho$. Thus, minimizing over $\mathcal{A}$ gives $r \le 2 \cdot \text{OPT}_{B_{\text{expan}}}(X)$. □

Remark 1 (coverage interpretation). Proposition 1 bounds the final selection $\mathcal{C}$, but against $\text{OPT}_{B_{\text{expan}}}$ rather than OPT_B, since the B_{init} voxel seeds need not be mutually separated and the pairwise argument uses the B_{expan} FPS picks together with the worst-covered point. Combined with Theorem 1, CoVeR attains:

$$d_H(X, \mathcal{C}) \le \min\big(\sqrt{3}\, v_s,\ 2 \cdot \text{OPT}_{B_{\text{expan}}}(X)\big)$$

Table A4. **Category-level performance on OpenEQA.** We compare CoVeR with VTC [24] and DTC [24] based on their published scores, computed by `GPT-4` [1]. **LLM-Match.** is the overall score across seven categories and **Rel.** is average percentage of performance maintained. Categories: (a) object recognition, (b) object localization, (c) attribute recognition, (d) spatial understanding, (e) object state recognition, (f) functional reasoning, and (g) world knowledge. `*LLaVA-OV-7B`

Methods	EQA Category							LLM Match.	Rel.
	(a)	(b)	(c)	(d)	(e)	(f)	(g)		
Retain 100% Tokens									
`LLaVA*`	48.6	43.0	74.4	43.6	74.5	53.1	55.0	56.2	100
Retain 43% Tokens									
VTC	–	–	–	–	–	–	–	54.2	96.4
DTC	–	–	–	–	–	–	–	54.3	96.6
CoVeR	49.9	40.3	74.6	45.2	71.6	54.0	54.6	**55.9**	**99.5**
Retain 26% Tokens									
VTC	40.3	35.7	62.9	40.3	71.5	52.5	49.5	50.5	89.9
DTC	44.6	39.4	72.7	43.9	71.3	53.0	53.2	54.1	96.3
CoVeR	47.9	40.0	72.8	45.1	73.0	54.2	54.6	**55.5**	**98.8**
Retain 17% Tokens									
VTC	37.2	33.1	59.9	38.4	65.9	51.0	45.9	47.4	84.3
DTC	41.2	38.1	68.3	42.3	70.5	55.3	50.7	52.5	93.4
CoVeR	46.7	40.1	68.4	45.1	69.9	54.3	53.3	**54.0**	**96.1**
Retain 8% Tokens									
VTC	–	–	–	–	–	–	–	43.6	77.6
DTC	–	–	–	–	–	–	–	49.3	87.7
CoVeR	43.1	34.7	61.9	41.7	65.9	52.0	51.4	**50.1**	**89.1**

These bounds provide complementary guarantees on the final coverage without solving the NP-hard k-center objective.

Remark 2 (case 3). If $|\mathcal{C}_{\text{init}}| > B$, CoVeR keeps the representatives of the B most populated voxels (Sec. 3.3). The budget is still exact, but this subset drops some voxel representatives, so Lemma 1 need not hold. This case does not arise at any budget or split we evaluate.

Summary. The analysis gives (1) an exact per-scene budget $|\mathcal{C}| = B$; (2) under the inactive-safeguard condition, every token within $\sqrt{3}\, v_s$ of a retained token after stage 1, preserved through stage 2; and (3) a 2-approximation to the k-center optimum at the expansion budget. Stage 1 thus contributes the absolute $\sqrt{3}\, v_s$ bound at low cost, while stage 2 fills the rest with a 2-approximation.

Fig. A2 validates the bound: $d_H(X, \mathcal{C})$ rises gradually as the budget shrinks but stays well below $\sqrt{3}\, v_s$, showing the bound is conservative.

C. Results on Token Recovery and Expansion

Figure A3 shows the cumulative distributions on Token Recovery (TR) and Token Expansion (TE) at all budget levels. At every budget, the TR curve reaches one, so for every

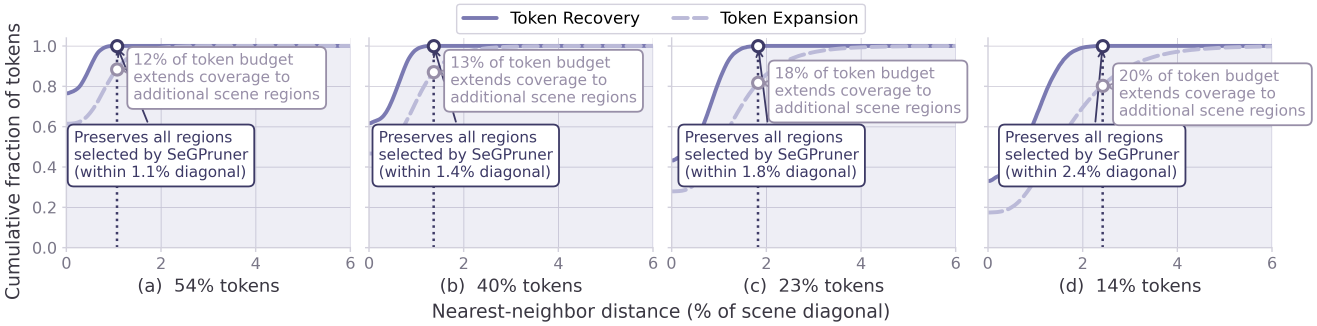


Figure A3. **Cumulative distributions of directed nearest-neighbor distances on ScanQA** across multiple token budgets. At all budgets, the Token Recovery (TR) reaches one, which indicates CoVeR keeps a token near every region SeGPruner selects, within 1.1%, 1.4%, 1.8%, and 2.4% of the scene diagonal, respectively. The Token Expansion (TE) also shows that CoVeR additionally covers regions SeGPruner leaves unrepresented.

Table A5. **Category-level performance on OpenEQA.** We compare CoVeR with VisPruner [75] and SeGPruner [34], computed by `GPT-4o` [28]. **LLM-Match.** is the overall score across seven categories and **Rel.** is the average percentage of performance maintained. Categories: (a) object recognition, (b) object localization, (c) attribute recognition, (d) spatial understanding, (e) object state recognition, (f) functional reasoning, and (g) world knowledge. `*LLaVA-OV-7B`

Methods	EQA Category							LLM Match.	Rel.
	(a)	(b)	(c)	(d)	(e)	(f)	(g)		
Retain 100% Tokens									
`LLaVA*`	53.9	43.4	77.6	49.1	73.1	58.9	57.0	59.1	100
Retain 43% Tokens									
VisPruner	52.3	41.9	76.0	49.0	74.0	58.2	57.2	58.4	98.8
SeGPruner	52.5	43.2	74.3	50.2	73.0	56.2	56.3	58.0	98.1
CoVeR	52.5	41.2	77.1	51.3	71.9	58.9	57.2	**58.6**	**99.2**
Retain 26% Tokens									
VisPruner	51.0	39.2	75.5	51.4	70.8	56.8	55.4	57.1	96.6
SeGPruner	49.9	39.8	76.2	50.1	72.6	57.4	56.7	57.5	97.3
CoVeR	51.0	40.9	75.1	49.9	73.2	57.8	56.0	**57.7**	**97.6**
Retain 17% Tokens									
VisPruner	50.4	38.6	70.9	50.1	69.1	57.6	54.8	55.9	94.6
SeGPruner	48.3	38.7	72.4	50.6	69.1	57.3	56.6	56.0	94.8
CoVeR	49.7	41.4	72.0	50.5	70.3	58.3	55.4	**56.8**	**96.1**
Retain 8% Tokens									
VisPruner	40.5	36.9	65.5	45.5	64.6	56.7	51.4	51.5	87.1
SeGPruner	44.6	38.6	63.4	46.6	65.8	55.0	53.5	52.5	88.8
CoVeR	47.3	36.3	65.6	46.3	66.2	55.5	54.0	**53.0**	**89.7**

region SeGPruner selects, CoVeR retains a token nearby, within 2.4% of the scene diagonal at the tightest budget. The TE further shows that CoVeR also places tokens in regions SeGPruner leaves uncovered. By prioritizing coverage, CoVeR still retains the informative regions that *learned importance* methods select, while additionally covering other regions in the scene. This shows our advantage over *learned importance*, which spends its budget on near-duplicate tokens from a few prominent regions and thus leaves the scene unrepresented.

D. Additional Quantitative Results

OpenEQA Category Analysis. CoVeR is particularly strong on spatial understanding (Tables A4-A5). At 17% retention, its spatial score exceeds the full model when evaluated via `GPT-4` LLM Match (45.1 vs. 43.6) and `GPT-4o` (50.5 vs. 49.1). At 8%, it also gives the best object-recognition score (47.3 vs. 44.6 for SeGPruner and 40.5 for VisPruner), with 89.1-89.7% overall retention across both judges.

Other 3D Reasoning Models. Tables A6 and A7 provide a broader context across task-specific 3D models, video LMMs, and general VLMs. CoVeR reaches 50.1 OpenEQA LLM-Match with 8% of tokens and loses only 0.7 points at 26% while outperforming prior models. On ScanQA, 23% retention yields 28.5 EM@1, exceeding the listed task-specific and open VLM systems. Thus, the pruned model remains competitive in absolute terms, not only relative to its full-token `LLaVA-OV-7B` baseline.

E. Additional Qualitative Results

Figures A4, A5, A6, A7, and A8 provide additional qualitative results from ScanQA, SQA3D, and OpenEQA datasets. Across different token budgets and 3D reasoning tasks, CoVeR preserves sufficient visual evidence to support object understanding, spatial and situated reasoning, and embodied question answering despite significant token pruning. These visualizations complement our quantitative findings by demonstrating that coverage-based token pruning maintains diverse scene information to perform 3D multi-view reasoning.

Table A6. **OpenEQA comparison.** CoVeR remains competitive, relative to both commercial and open VLMs, after removing up to 92% of visual tokens. (·) denote change from the base model. LLM-Match uses `GPT-4` [1].

Models	LLM-Match↑
Blind Text-only LLM baseline	
GPT4 [1]	33.5
LLaMA-2 70B [54]	28.3
Closed VLM models	
Claude-3 Opus	36.3
Gemini 1.0 Pro Vision [52]	44.9
Claude-3.5 Sonnet	48.7
GPT4-V (15 frames) [1]	54.6
GPT4-V (50 frames) [1]	55.3
Open VLM models	
Video-LLaMA [73] [EMNLP 2023]	20.0
LLaMA-2 w/ Concept Graph [42] [CVPR 2024]	28.7
AuroraCap [6] [ICLR 2025]	28.9
Video-ChatGPT [41] [ACL 2024]	32.1
LLaMA-2 w/ Sparse Voxel Map [42] [CVPR 24]	34.3
LLaMA-2 w/ LLaVA-1.5 Caption [42] [CVPR 24]	36.8
Chat-UniVi [30] [CVPR 2024]	42.3
Video-LLaMA2 [10] [arXiv 2024]	49.2
InternVL2.5 (16 frames) [9] [arXiv 2024]	54.4
MovieChat (w/ `LLaVA-OV-7B`) [49] [CVPR 2024]	54.9
LLaVA-3D (32 frames) [83] [ICCV 2025]	53.2
Qwen2.5-VL (16 frames) [4] [arXiv 2025]	50.8
Magma (16 frames) [66] [CVPR 2025]	49.1
NVILA (16 frames) [38] [CVPR 2025]	54.0
InternVL3 (16 frames) [84] [arXiv 2025]	55.5
ThinkAct (16 frames) [22] [NeurIPS 2026]	56.2
Ours (12 frames)	
`LLaVA-OV-7B` (100%)	56.2
w/ CoVeR ↓44% tok	56.0 (-0.2)
w/ CoVeR ↓57% tok	55.9 (-0.3)
w/ CoVeR ↓74% tok	55.5 (-0.7)
w/ CoVeR ↓83% tok	54.0 (-2.2)
w/ CoVeR ↓92% tok	50.1 (-6.1)

Table A7. **ScanQA (val) comparison.** CoVeR remains competitive with task-specific 3D models, video-LMMs, and fine-tuned 3D-LMMs, after removing up to 91% of visual tokens, without 3D-specific training or learned pruning.

Models	EM@1↑
Task-specific models	
VoteNet+MCAN [70] [CVPR 2019]	17.3
ScanRefer+MCAN [70] [CVPR 2019]	18.6
ScanQA [2] [CVPR 2022]	21.1
Jin et al. [31] [CVPR 2023]	21.7
3D-VisTA [85] [ICCV 2023]	22.4
3DVLP [77] [AAAI 2024]	24.0
DSPNet (20 frames) [39] [CVPR 2025]	23.5
Video-LMMs	
Agent3D-Zero [76] [ECCV 2024]	17.5
LLaVA-NeXT-Video [37] [arXiv 2024]	18.7
MovieChat (w/ LLaVA-OV-7B) [49] [CVPR 24]	26.0
AuroraCap [6] [ICLR 2025]	17.2
Task-specific fine-tuned image/video LMMs	
NaviLLM [81] [CVPR 2024]	23.0
Spatial-MLLM-4B [59] [NeurIPS 2025]	26.3
SPAR-mix [74] [NeurIPS 2025]	27.7
SplatTalk-ScanQA-FT (100 frames) [53] [ICCV 2025]	22.3
Proxy3D (32 frames) [29] [CVPR 2026]	25.2
Task-specific fine-tuned 3D-LMMs	
3D-LLM [21] [NeurIPS 2023]	20.5
FE-3DGQA [80] [TCSVT 2022]	22.3
ChatScene [23] [NeurIPS 2024]	21.6
LEO [25] [ICML 2024]	24.5
Scene-LLM [17] [WACV 2025]	27.2
Yuan et al. [71] [CVPR 2025]	22.9
Inst3D-LMM [69] [CVPR 2025]	24.6
Ours (12 frames)	
`LLaVA-OV-7B` (100%)	28.2
w/ CoVeR ↓46% tok	28.7 (+0.5)
w/ CoVeR ↓60% tok	28.9 (+0.7)
w/ CoVeR ↓77% tok	28.5 (+0.3)
w/ CoVeR ↓86% tok	27.9 (-0.3)
w/ CoVeR ↓91% tok	27.1 (-1.1)

F. Broader Impact

Positive impact. CoVeR significantly reduces inference computation and memory, while maintaining the performance of baseline VLMs, making multi-view 3D reasoning more accessible to users with limited computational resources. It also provides precise token reduction control in VLMs. CoVeR also potentially lowers deployment energy costs in robotic and embodied AI applications.

Potential negative impact. Real-world visual reasoning can raise important privacy concerns. In addition, aggressive pruning may discard important cues, potentially leading to incorrect VLM answers in safety-critical systems like healthcare or autonomous driving. Besides, since CoVeR is a general-purpose method, it may inherit the societal risks, biases, and failure modes of its underlying VLMs and data.

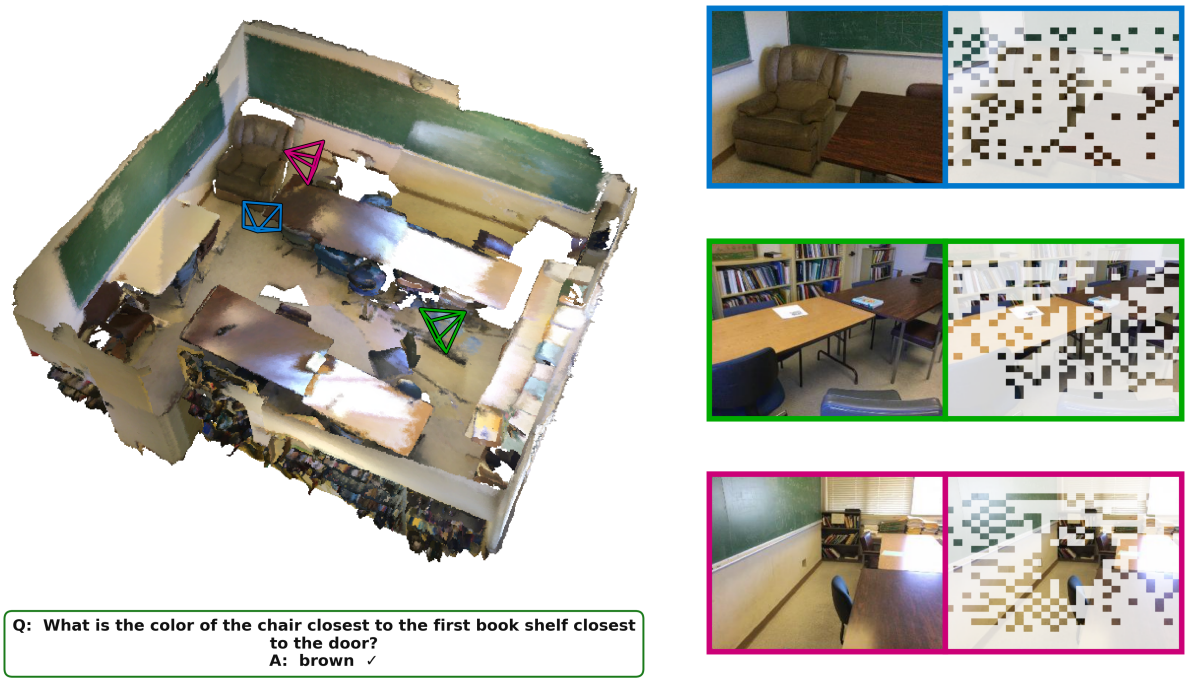
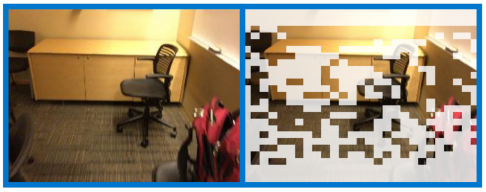


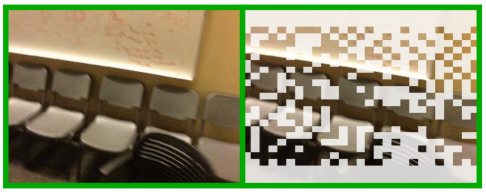
Figure A4. **Visual Results on ScanQA** [2] at 23% token retention.

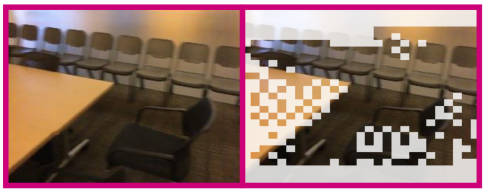


Figure A5. **Visual Results on ScanQA** [2] at 23% token retention.



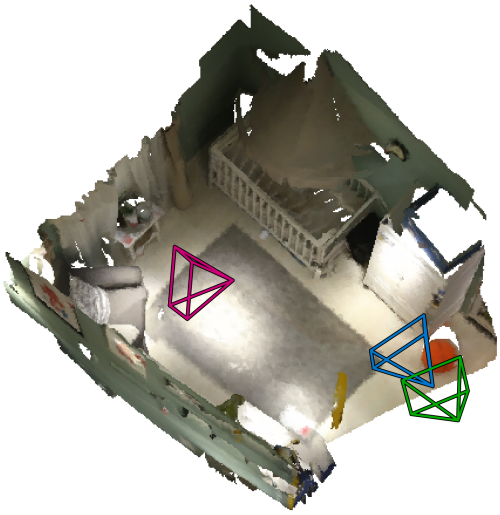




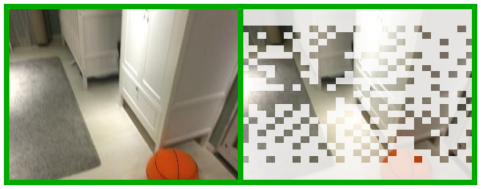


Q: I am facing a chair, while having a whiteboard on the right and another chair with a backpack on top of it behind me. What is the object to my right that rans the span of the wall?
A: whiteboard ✓

Figure A6. **Visual Results on SQA3D** [40] at 43% token retention.









Q: I am facing the wardrobe closet, and the red ball is on the floor to my right. Which direction should I go if I want to open the window?
A: left ✓

Figure A7. **Visual Results on SQA3D** [40] at 26% token retention.

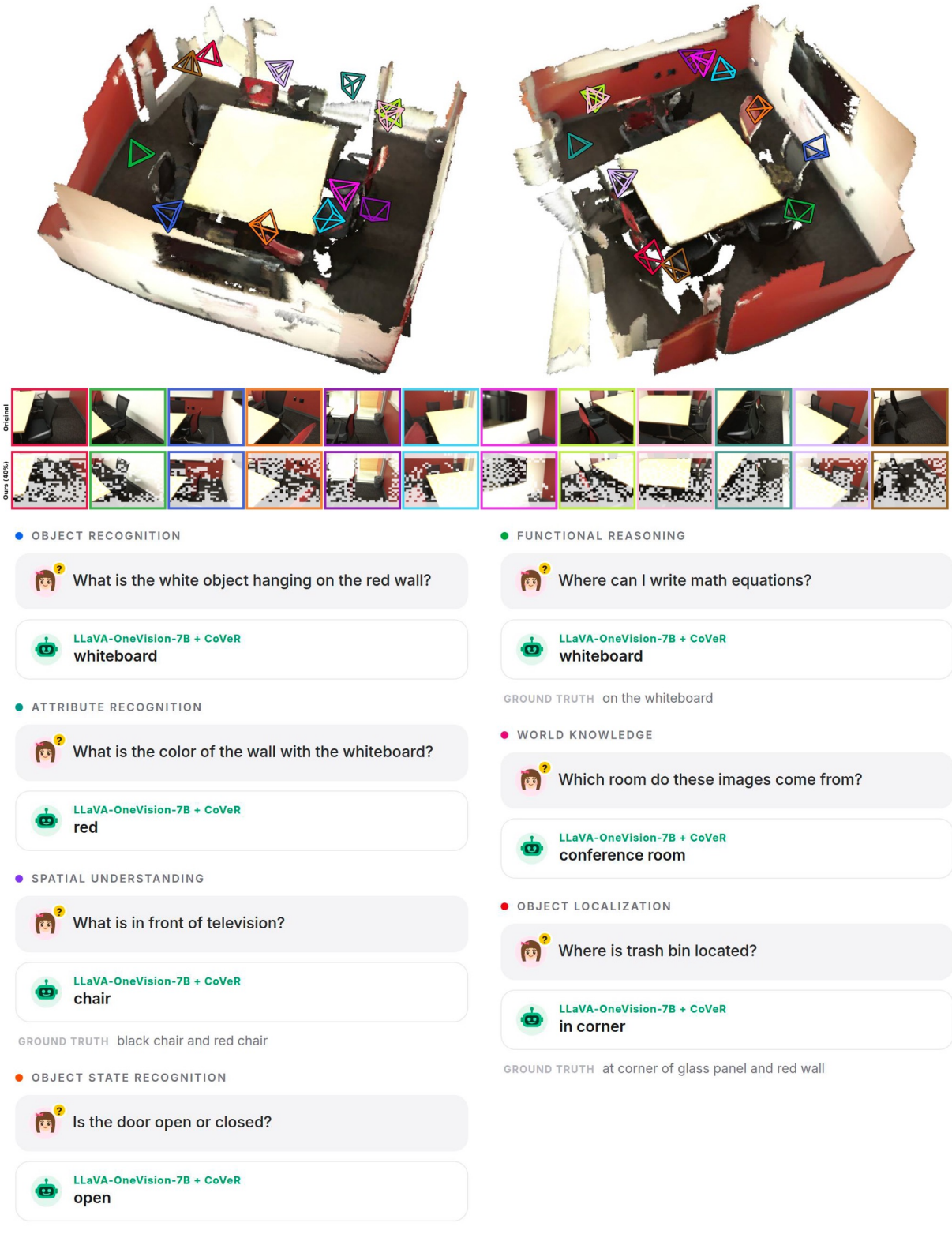


Figure A8. **Visual Results on OpenEQA** [42] at 40% token retention.