GeoGR: Spatial–Semantic Token Compression and Group Routing for Efficient 3D VLMs

Anonymous authors
Paper under double-blind review

Abstract

Multi-view representations enable pretrained image and video vision-language models to understand 3D scenes, but their long visual token sequences make inference costly. Redundancy arises both from repeated observations of the same scene regions across views and from processing all visual tokens at every LLM layer, regardless of their relevance to the current computation. We propose GeoGR, a two-stage framework that jointly compresses scene representations and visual computation. At the projector level, GeoSemZip consolidates cross-view observations into 3D voxel tokens and combines semantic saliency with spatial coverage to construct a compact scene representation. Unselected voxel features are aggregated into contextual anchors to retain scene information that would otherwise be discarded. At the LLM level, we identify an effective visual computation window and observe that query-conditioned token relevance remains locally stable but changes over longer layer intervals. Guided by these findings, GroupRoute combines late entry and early exit with recoverable group-wise skipping. Low-relevance tokens bypass complete LLM layers while retaining their hidden states and can rejoin the active group at subsequent anchor layers selected under a computation budget. For compression-aware post-training, we construct a compact dataset of 67K samples by retaining 30% of the original training pool. With this protocol, GeoGR retains 95.2% and 96.8% of the dense post-trained models' performance on LLaVA-OneVision-7B and Video-3D LLM-7B, respectively, while accelerating LLM prefill by $6.8\times$ and $6.1\times$. These results demonstrate the consistent effectiveness of GeoGR across 3D VLMs with and without explicit 3D spatial encoding, highlighting a favorable performance-efficiency trade-off.

1 Introduction

Enabling multimodal large language models to understand 3D scenes is an important step toward spatial intelligence (Hong et al., 2023). Progress in this direction is constrained by the relative scarcity of 3D-language data compared with large-scale image-text corpora (Zhu et al., 2025; Tang et al., 2025). Recent studies therefore represent 3D scenes as sequences of posed RGB or RGB-D views (Zheng et al., 2025; Zhu et al., 2025), allowing pretrained vision-language models (VLMs) for image and video understanding to be adapted for 3D reasoning. This approach transfers visual and language knowledge learned from 2D data to scene-level question answering, grounding, and captioning (Zhu et al., 2025; Zheng et al., 2025). However, aggregating multiple views produces long visual token sequences that substantially increase LLM inference cost. For example, Video-3D LLM uses 32 views per scene, producing 6,272 visual tokens (Zheng et al., 2025). Efficient visual token compression is therefore essential for multi-view 3D VLMs.

Multi-view 3D VLMs exhibit redundancy in both scene representations and visual computation. At the representation level, overlapping views repeatedly encode the same physical regions (Figure 1(a)). Geometric correspondence allows repeated observations to be consolidated (Huang et al., 2025b), but a compact representation must also preserve the evidence needed for scene understanding. This includes both salient object details and less salient regions that provide spatial context. Compression therefore needs to balance semantic saliency with scene coverage (Li et al., 2026b; Ling et al., 2026; Li et al., 2026a). At the computation level, dense inference processes all visual tokens at every LLM layer. Studies of image-based VLMs suggest that the contribution of visual computation varies with depth (Wu et al., 2026a), motivating a closer examination of its allocation in multi-view 3D VLMs.

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

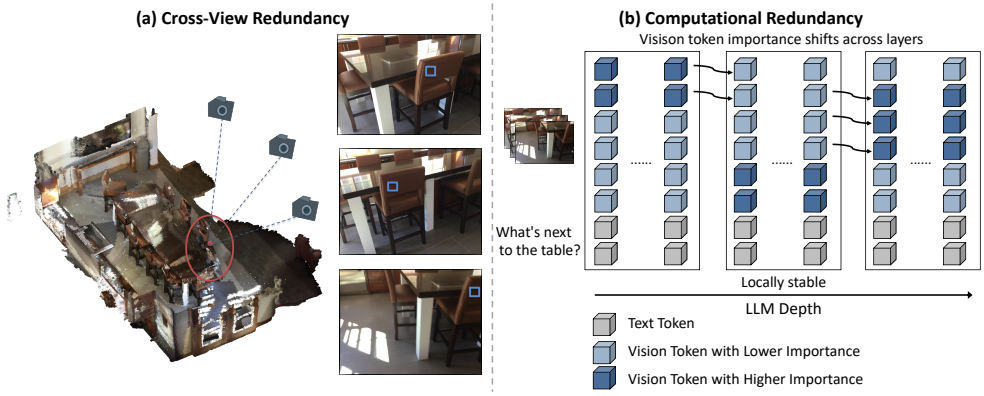


Figure 1: **Redundancy in multi-view 3D VLMs.** (a) Overlapping views repeatedly observe the same 3D region (blue boxes). (b) For the same question, visual token importance remains locally stable but shifts across more distant layers. See Section 3 for quantitative analysis.

Beyond deciding where visual tokens should participate, compression must also account for their changing relevance to the query across layers (Figure 1(b)). Tokens with low relevance at one layer may become useful later, so permanently removing them based on a single-layer ranking can discard evidence needed in subsequent layers.

To address these two forms of redundancy, we propose GeoGR, a two-stage framework that compresses multi-view scene representations and selectively allocates visual computation within the LLM. At the projector level, between the vision encoder and the LLM (Wu et al., 2026b), we introduce GeoSemZip to construct a compact scene representation. GeoSemZip maps cross-view observations into a shared 3D coordinate system and consolidates them into voxel tokens. It retains voxel tokens with high encoder-attention saliency as dominant tokens and selects contextual anchors by sampling the remaining voxels along their 3D Morton order. These anchors complement the salient tokens by covering less salient scene regions. Unselected voxel features are aggregated into the contextual anchors according to feature similarity, retaining scene information that would otherwise be discarded. GeoSemZip thus shortens the visual sequence while preserving salient content and spatial coverage.

Within the LLM, we introduce GroupRoute to reduce visual computation through Late Entry, Early Exit, and Recoverable Group-wise Skipping (RGS). Training-free interventions that separately delay visual token entry and advance visual token exit suggest a layered pattern of visual computation in multi-view 3D VLMs: portions of shallow- and deep-layer processing can be omitted with limited changes to predictions. Using next-token KL divergence relative to full visual computation, we identify candidate window boundaries, then compare the candidates after compression-aware post-training to select the final window. Within this window, query-conditioned token relevance remains locally stable but changes over longer layer intervals (Section 3), motivating group reuse over short intervals with periodic reassessment. At anchor layers, RGS forms high- and low-priority groups using query-conditioned attention. High-priority tokens execute subsequent LLM layers, while low-priority tokens bypass them with their hidden states preserved. All visual tokens execute the next anchor layer and are rescored, allowing previously skipped tokens to rejoin the active group. Budget-constrained Anchor and Regrouping Search (BARS) selects anchor positions to minimize the risk of reusing groups under a computation budget.

To support compression-aware post-training, we use PRISM (Bi et al., 2025) to select the least redundant 30% of samples from the original training pool, yielding a compact dataset of 67K samples. Using this post-training protocol, we evaluate GeoGR on LLaVA-OneVision-7B and Video-3D LLM-7B, without and with explicit 3D spatial encoding, respectively. Evaluation covers two question-answering benchmarks for LLaVA-OneVision and five benchmarks spanning question answering, visual grounding, and dense captioning for Video-3D LLM. GeoGR retains 95.2% and 96.8% of the respective dense post-trained models' performance, with LLM prefill speedups of $6.8\times$ and $6.1\times$ and time-to-first-token (TTFT) speedups of $2.4\times$ and $2.3\times$, respectively.

Our main contributions are as follows: **(1)** We propose GeoSemZip, a projector-level compression module that combines cross-view geometric consolidation with semantic saliency and spatial coverage. It aggregates unselected voxel features into contextual anchors to retain scene information under compression. **(2)** We identify effective visual computation windows in multi-view 3D VLMs and

observe locally stable but evolving query-conditioned token relevance across layers. Guided by these findings, we propose GroupRoute, which combines Late Entry and Early Exit with Recoverable Group-wise Skipping. **(3)** We construct a compact 67K-sample dataset for compression-aware post-training of 3D VLMs and will release it publicly. **(4)** We propose GeoGR, a two-stage compression framework comprising GeoSemZip and GroupRoute that achieves a favorable performance-efficiency trade-off on LLaVA-OneVision-7B and Video-3D LLM-7B.

2 Related Work

Token compression reduces visual sequence length within the vision encoder, at the projector between the encoder and the LLM, or within the LLM (Wu et al., 2026b). Methods select tokens by visual saliency (Shang et al., 2025) or language-conditioned relevance (Xing et al., 2024), merge similar tokens (Bolya et al., 2022), or combine selection with aggregation (Yang et al., 2025). For multi-view 3D inputs, geometric correspondence guides cross-view redundancy reduction, while spatial diversity promotes scene coverage. DTC (Huang et al., 2025b) and ToSA (Huang et al., 2025a) guide merging through voxel constraints and depth-derived spatial tokens, respectively. SeGPruner (Li et al., 2026b) and Geo3DPruner (Li et al., 2026a) select geometrically diverse tokens, while Prune2Drive (Xiong et al., 2026) combines diversity-aware sampling with view-adaptive budgets. Within the LLM, Fast3D (Huang et al., 2025c) assigns sample-adaptive pruning budgets, and AdaToken-3D (Zhang et al., 2025a) schedules layer-wise token retention using spatial information flow. HiSC (Qu et al., 2026) combines spatial graph-based merging before the LLM with hierarchical cluster-based pruning within it. Our GeoSemZip combines salient token selection with spatial coverage and aggregates unselected voxel features into contextual anchors to retain additional scene information.

Operation compression reduces visual computation along LLM width and depth (Wu et al., 2026b). In image-based MLLMs, RedundancyLens (Li et al., 2025) and YOPO (Zhang et al., 2024) exploit redundancy within attention and FFNs, with YOPO also pruning deep-layer visual computation. Along depth, Skip-Vision (Zeng et al., 2025) skips FFN updates for selected visual tokens, whereas ShortV (Yuan et al., 2025) freezes visual token updates in low-contribution layers. HiDrop (Wu et al., 2026a) combines late injection and early exit with progressive token pruning. ViCA (Liu et al., 2026) bypasses visual token updates and provides text access to static visual features through sparse cross-attention. Our GroupRoute extends late entry and early exit to 3D VLMs with recoverable group-wise skipping. Query-conditioned token groups bypass complete LLM layers with their states preserved and are reassessed at subsequent anchor layers for reactivation.

3 Understanding Layer-Wise Visual Token Dynamics in 3D VLMs

Visual tokens play different roles across LLM layers, with vision–language interaction concentrated in the middle layers (Wu et al., 2026a). This motivates late visual entry and early visual exit, which restrict visual computation to a middle-layer window (Section 4.3). Within this window, token rankings may still change as vision–language interaction proceeds, making it unclear whether a single-layer ranking provides a reliable basis for permanent token pruning. We therefore examine the cross-layer stability of attention-based visual token rankings, focusing on their consistency across nearby layers and whether initially low-ranked tokens become highly ranked at later layers.

Tracking Cross-layer Ranking Stability. We conduct our analysis on ScanQA (Azuma et al., 2022) using LLaVA-OneVision, focusing on layers 8–23 within the middle-layer visual computation window. At layer l, we assign each visual token i an attention score using the attention from the last prompt token q before answer generation, averaged over all H attention heads:

$$s_i^{(l)} = \frac{1}{H} \sum_{h=1}^{H} A_{h,q,i}^{(l)},$$

where $A_{h,q,i}^{(l)}$ is the attention weight from q to i in head h. We rank tokens by $s_i^{(l)}$ and split them into top and bottom halves. For each layer pair $(l, l+\Delta)$, we compute Spearman correlation and normalized rank shift, alongside the fraction of

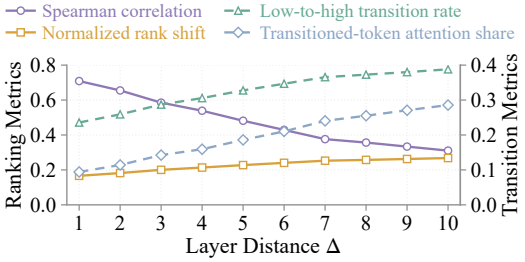


Figure 2: **Cross-layer stability of visual token rankings.** Ranking stability decreases with layer distance, while initially low-ranked tokens increasingly enter the top half and account for a growing share of its attention.

later top-half tokens originating from the earlier bottom half and their attention share within the later top half. We average these metrics over all valid layer pairs at each distance $\Delta = 1, \ldots, 10$. Figure 2 shows two complementary patterns: rankings are more stable across nearby layers, while initially low-ranked tokens increasingly enter the top half and contribute a growing share of its attention at larger layer distances. See Appendix E.1 for full layer-pair heatmaps.

Ranking Stability over Layer Distance. Figure 2 shows ranking stability through two metrics on the left y-axis: Spearman correlation measures agreement between complete token rankings, while normalized rank shift measures the mean absolute rank change divided by the maximum rank span. Nearby layers exhibit higher rank correlation and smaller rank shifts, whereas agreement generally weakens as layer distance increases. These results indicate that token rankings retain local consistency but cannot be assumed to remain stable throughout the visual computation window. This local consistency motivates reusing a token partition over a short layer interval to reduce the overhead of repeated scoring and routing.

Low-to-High Rank Transitions. The two metrics on the right y-axis of Figure 2 quantify how initially low-ranked tokens enter the top half at later layers. The transition rate measures their fraction of the later top half, while the attention share measures their contribution to that group's total attention. As layer distance increases, a growing portion of the later top half originates from initially low-ranked tokens. Crucially, these tokens also account for a growing, non-negligible share of the attention assigned to that group. A low rank at one layer therefore does not guarantee a low rank at subsequent layers. This motivates treating low-ranked tokens as *temporarily inactive at the current stage*, rather than *permanently redundant in all subsequent layers*, and retaining them for later reassessment.

Implications for Group-wise Computation. Together, these observations motivate reusing token partitions over short layer intervals while preserving the opportunity for reassessment. Local ranking stability motivates temporary reuse of a partition, while low-to-high transitions motivate retaining low-ranked tokens for later recovery and regrouping. This leads to the group-wise skip mechanism in Section 4.3, where low-ranked tokens temporarily skip LLM blocks while preserving their hidden states. At the next anchor layer, all visual tokens, including those previously skipped, undergo full LLM-block computation and are rescored to determine their groups for the following interval.

4 METHOD

As illustrated in Figure 3, *GeoGR* combines projector-level and LLM-level compression for multi-view 3D VLMs. *GeoSemZip* consolidates cross-view observations into compact tokens preserving salient content and spatial coverage. *GroupRoute* combines *Late Entry and Early Exit* to define a visual computation window with *Recoverable Group-wise Skipping* to selectively allocate computation according to query relevance while preserving skipped token states for later reactivation. We first present the distillation dataset and post-training protocol, followed by the two compression stages.

4.1 POST-TRAINING PROTOCOL FOR 3D VLM COMPRESSION

We collect 223K training examples from ScanRefer (Chen et al., 2020), Multi3DRefer (Zhang et al., 2023), Scan2Cap (Chen et al., 2021), ScanQA (Azuma et al., 2022), and SQA3D (Ma et al., 2022). To reduce post-training cost, we construct a distilled dataset using PRISM (Bi et al., 2025), a training-free data selection method that estimates redundancy through the average correlation between mean-centered intrinsic visual representations. We retain the least redundant 30% of samples from each source, yielding a 67K-example post-training dataset, as summarized in Table 1. To assess the subset's effectiveness, we post-train dense models without any compression enabled. Relative to their counterparts post-trained on the full dataset, Video-3D LLM-7B and LLaVA-OneVision-7B retain 98.72% and 95.69% of performance, respectively. Detailed results for dense models post-trained on the full and distilled datasets are provided in Appendix D.1. This dataset supports a unified compression-aware post-training protocol for adapting models to compression.

Table 1: Our post-training dataset distilled by selecting 30% of samples from each source using PRISM (top), and performance retained relative to full-data post-training (bottom).

Source	**Full Set**	**Selected**
ScanRefer	36,665	11,000
Multi3DRefer	43,838	13,151
Scan2Cap	36,665	11,000
ScanQA	26,515	7,955
SQA3D	79,445	23,834
Total	223,128	66,940
3D VLMs	**Retention (%)**	
Video-3D LLM-7B	98.72	
LLaVA-OneVision-7B	95.69	

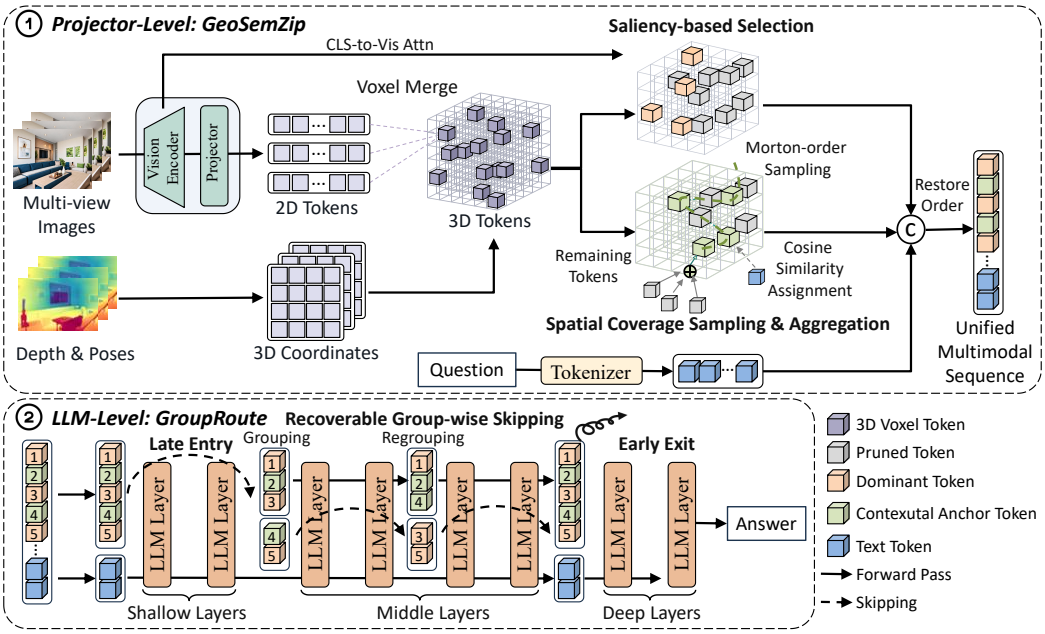


Figure 3: **Overview of GeoGR**. **(1) Projector-level Compression:** GeoSemZip consolidates cross-view observations into 3D voxel tokens using depth and camera poses. Saliency-based selection preserves dominant tokens, while Morton-order sampling selects contextual anchors that aggregate unselected voxel features by cosine similarity. The compressed visual tokens are combined with text tokens to form the LLM input. **(2) LLM-level Compression:** GroupRoute confines visual computation to an intermediate-layer window through Late Entry and Early Exit. Within this window, Recoverable Group-wise Skipping routes query-conditioned token groups through or around complete LLM layers. Skipped tokens retain their hidden states and can resume computation after regrouping at subsequent anchor layers. Text tokens remain active throughout the LLM.

4.2 GeoSemZip: Spatial–Semantic Token Compression

GeoSemZip constructs a compact scene representation preserving salient content and spatial coverage under a fixed token budget. It consolidates cross-view observations into 3D voxel tokens, then divides the budget between salient dominant tokens and spatially distributed contextual anchors. Remaining voxel features are aggregated into the contextual anchors based on feature similarity.

Cross-view Voxel Consolidation. Given T calibrated views with aligned depth, let $\{\mathbf{x}_i\}_{i=1}^{N}$ denote the projected visual tokens, where $\mathbf{x}_i \in \mathbb{R}^d$ and $N = THW$ for an $H \times W$ grid per view. Back-projection using depth, intrinsics, and camera-to-world poses assigns each token a world coordinate $\mathbf{p}_i$. Saliency a_i is computed from received attention in the last executed vision-encoder layer, averaged over heads and query positions and pooled to the projected grid. Voxelization with width s gives $\mathbf{g}_i = \mathrm{round}(\mathbf{p}_i/s)$. Let $\mathcal{V}$ be the occupied voxel set and $\mathcal{G}_v = \{i : \mathbf{g}_i = v\}$ the token indices in voxel v. We aggregate each voxel's features by averaging and saliency by max pooling:

$$\mathbf{z}_v = \frac{1}{|\mathcal{G}_v|} \sum_{i \in \mathcal{G}_v} \mathbf{x}_i, \qquad b_v = \max_{i \in \mathcal{G}_v} a_i$$

Feature averaging consolidates spatially co-located observations across views, while max pooling preserves the strongest saliency response for subsequent token selection.

Saliency- and Coverage-aware Selection. Given $K = |\mathcal{V}|$ occupied voxels and a target keep ratio r, we set the token budget to $M = \min(K, \lceil rN \rceil)$. We allocate a fraction ρ to the highest-saliency dominant tokens and the remainder to contextual anchors:

$$M_d = \mathrm{clip}(\mathrm{round}(\rho M), 1, M-1), \quad M_c = M - M_d, \quad \mathcal{D} = \mathrm{TopK}_{v \in \mathcal{V}}(b_v, M_d)$$

For spatial coverage, we select contextual anchors from the remaining voxels $\mathcal{R} = \mathcal{V} \setminus \mathcal{D}$. Their integer coordinates are shifted to be nonnegative and sorted by 3D Morton code, producing a sequence that approximately preserves spatial locality. We sample M_c anchors at a fixed stride $\lfloor |\mathcal{R}|/M_c \rfloor$ to form $\mathcal{C}$. This selection is independent of saliency scores, encouraging coverage of less salient regions.

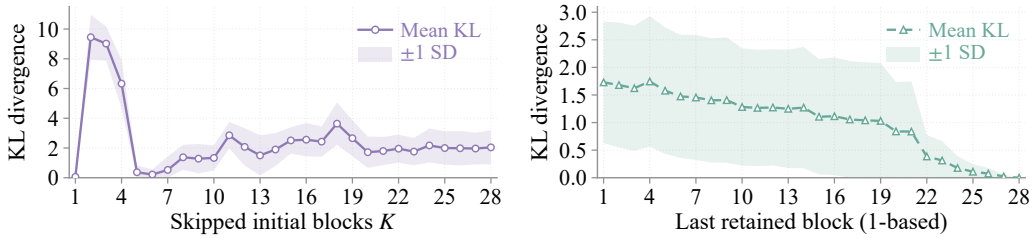


Figure 4: Visual-window diagnosis for LLaVA-OneVision on 256 ScanQA examples: Late Entry (left) and Early Exit (right). Lines show mean next-token KL divergence relative to full visual computation; shading indicates ± 1 standard deviation, with the lower bound clipped at zero.

Residual Merging and Serialization. To retain information from voxels selected by neither branch, we aggregate their features into contextual anchors. Each unselected voxel is assigned to the anchor whose feature has the highest cosine similarity to its own. Each anchor then adds the mean feature of its assigned voxels as a residual, incorporating unselected content without increasing the token budget. Anchors with no assigned voxels and all dominant tokens remain unchanged. For serialization, each retained voxel inherits the index of its earliest contributing input token. We sort the retained tokens by these indices, producing an ordered sequence of M visual tokens for the LLM.

4.3 GroupRoute: Late Entry, Early Exit, and Recoverable Group-wise Skipping

After *GeoSemZip*, *GroupRoute* uses *Late Entry* and *Early Exit* to define a visual computation window, within which *Recoverable Group-wise Skipping (RGS)* selectively updates tokens according to query relevance while preserving skipped token states for later reactivation. *Budget-constrained Anchor and Regrouping Search (BARS)* selects the regrouping anchors under a computation budget.

Late Entry and Early Exit. Following HiDrop (Wu et al., 2026a), we adapt late entry and early exit to multi-view 3D VLMs. Text tokens pass through all LLM layers, while visual tokens participate only within a window $[\ell_{\text{in}}, \ell_{\text{out}}]$. They enter at layer ℓ_{in} with their original full-prompt position IDs and exit after layer ℓ_{out}. Both boundaries use one-based indexing. To identify candidate boundaries, we independently vary the number of initial blocks skipped and the last block retaining visual tokens, measuring next-token KL divergence relative to full visual computation. As shown in Figure 4, skipping the first $K = 5$–7 blocks yields the final consecutive low-mean-KL region before divergence increases. For early exit, KL drops sharply between retained blocks 21 and 22 and remains low from block 23 onward. These pre-adaptation measurements define candidate ranges, which we refine through compression-aware post-training. The resulting window for LLaVA-OneVision is $[8, 23]$, as evaluated in the window ablation study.

Recoverable Group-wise Skipping. Within the visual-computation window, we partition visual tokens at designated anchor layers according to their relevance to the query. The query is the last prompt token for question answering and captioning, or the grounding token for localization. At anchor a, we reuse the layer's query and key projections and positional encoding to compute head-averaged attention scores from its input representations:

$$s_i^{(a)} = \frac{1}{H_q} \sum_{h=1}^{H_q} \left[\text{softmax}\left(\frac{\mathbf{q}_{a,h} \mathbf{K}_{a,h}^{\top}}{\sqrt{d_h}} \right) \right]_i$$

Here H_q is the number of query heads, d_h is the head dimension, and the softmax covers all causally visible prompt tokens. The highest-scoring $\lceil \kappa N_{\text{vis}} \rceil$ visual tokens form the high-priority group $\mathcal{H}_a$, and the remaining tokens form the low-priority group $\mathcal{L}_a$, where N_{vis} is the number of visual tokens retained by GeoSemZip. All visual tokens execute the anchor layer; the partition takes effect only in subsequent layers. Until the next anchor, text tokens and $\mathcal{H}_a$ execute complete LLM layers with their original position IDs, while $\mathcal{L}_a$ bypasses both attention and FFN computation and contributes no keys or values. The skipped tokens retain their hidden states and positions. At the next anchor, all visual tokens are rescored and updated, allowing previously inactive tokens to rejoin the high-priority group. The final partition remains in effect until early exit.

Budget-constrained Anchor Selection. BARS selects anchor positions under a computation budget using layer-wise attention scores normalized over retained visual tokens. For a partition formed at anchor a and reused at layer ℓ, we quantify three risks: low-priority attention mass, migration from the low-priority to high-priority group, and mean normalized rank displacement. Risks are weighted

by the remaining visual-computation depth, averaged within benchmarks, and then equally across benchmarks. For each feasible anchor count m, dynamic programming finds the minimum-risk schedule $\mathcal{A}_m$ under a normalized visual token–layer budget and minimum anchor gap. Starting with $\mathcal{A}_1$, BARS considers increasing anchor counts, accepting a candidate only if it reduces both aggregate risk and each calibration subset's risk by at least $\tau = 5\%$ relative to the current schedule. With $\kappa = 0.5$, BARS selects anchors $[9, 13]$ on both 3D VLMs. The schedule is fixed at inference, while token groups remain query-dependent. Appendix B.1 provides the risk definitions and procedure.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Models. We evaluate GeoGR on two representative multi-view 3D VLMs: LLaVA-OneVision-7B (Li et al., 2024), which independently encodes views without explicit 3D representations, and Video-3D LLM-7B (Zheng et al., 2025), which augments view features with depth- and pose-derived 3D positional embeddings in a shared scene coordinate frame. Both models take 32 uniformly sampled RGB views per scene. Architectural details are provided in Appendix C.1.

Benchmarks. We evaluate GeoGR on five ScanNet-based benchmarks covering visual grounding (ScanRefer (Chen et al., 2020) and Multi3DRefer (Zhang et al., 2023)), dense captioning (Scan2Cap (Chen et al., 2021)), and 3D question answering (ScanQA (Azuma et al., 2022) and SQA3D (Ma et al., 2022)). Video-3D LLM is evaluated on all five, while LLaVA-OneVision is evaluated only on the two QA benchmarks because it does not support the required 3D localization outputs. See Appendix C.2 for details.

Efficiency Evaluation. We estimate LLM-side visual FLOPs. Let n_ℓ be the number of active visual tokens at layer ℓ, d the hidden dimension, m the gated FFN intermediate dimension, and $r = H_{\mathrm{KV}}/H_{\mathrm{Q}}$ the key/value-to-query head ratio. The Q/K/V/O projection widths are d, rd, rd, and d. Counting multiply-adds as two FLOPs and using dense $n_\ell \times n_\ell$ visual attention, we obtain

$$\mathrm{FLOPs}_{\mathrm{vis}} \approx 2 \sum_{\ell=1}^{L} \left[(2 + 2r) n_\ell d^2 + 2 n_\ell^2 d + 3 n_\ell d m \right]$$

Both 3D VLMs use Qwen2-7B with $H_{\mathrm{Q}} = 28$ and $H_{\mathrm{KV}} = 4$, giving $r = 1/7$.

Implementation Details. For LLaVA-OneVision, we pool the projector features of each view to a 14×14 grid, yielding 6,272 visual tokens per scene before compression. Unless otherwise stated, GeoSemZip uses voxel size $s = 0.1$, target keep ratio $r = 0.30$, and dominant-token ratio $\rho = 0.85$, with max pooling for voxel saliency scores. The visual computation windows are $[8, 23]$ for LLaVA-OneVision and $[8, 24]$ for Video-3D LLM, using one-based, inclusive layer indices. BARS selects anchor layers 9 and 13 for both models, with a keep ratio $\kappa = 0.5$ at each anchor. Compression-aware post-training has two stages. The first enables GeoSemZip and trains the projector and full LLM with the vision encoder frozen. The second enables Late Entry and Early Exit and trains only the full LLM, keeping the vision encoder and projector frozen. All evaluations use greedy decoding (temperature 0, one beam) with at most 512 new tokens. Complete configurations are provided in Appendix A.1.

5.2 MAIN RESULTS

Comparison with State-of-the-art Methods. Table 2 compares GeoGR with representative 2D and 3D compression methods at approximately 10% retained LLM-side visual FLOPs. On LLaVA-OneVision, GeoGR retains 95.18% of the dense model's performance, exceeding the strongest baseline, DTC, by 1.88 percentage points while requiring fewer FLOPs (8.65 vs. 9.36 TFLOPs). On Video-3D LLM, we evaluate five benchmarks covering QA, visual grounding, and dense captioning. GeoGR achieves the highest average performance retention of 96.80% at 9.09 TFLOPs. Compared with the strongest baseline, VisionZip†, it improves retention by 0.79 percentage points while reducing FLOPs by 3.2%. The results also suggest that compression effectiveness depends on how spatial information is incorporated. After compression-aware post-training, the 2D-native methods VisionZip and VisPruner retain 96.01% and 95.87% of performance on Video-3D LLM, compared with 88.34% and 92.26% on LLaVA-OneVision. Although these averages cover different benchmark sets, a plausible explanation is that Video-3D LLM embeds depth- and pose-derived 3D positional information into its visual representations, potentially benefiting even compressors designed for 2D inputs. LLaVA-OneVision lacks this explicit spatial encoding, but 3D compression methods can still use geometric information to guide token selection and aggregation. Such external guidance may be

Table 2: **Performance on LLaVA-OneVision-7B and Video-3D LLM-7B at approximately 10% retained LLM-side visual FLOPs.** LLaVA-OneVision is evaluated only on ScanQA and SQA3D, as it does not support the 3D localization outputs required by the other benchmarks. "Avg." measures performance retention relative to the dense model. "Tokens" denotes the average number of active visual tokens per LLM layer. Unshaded baseline rows apply compression without subsequent recovery, while light-red rows use compression-aware post-training. Light-blue rows denote GeoGR. † marks baselines originally proposed without such recovery and evaluated with it here. * marks compression baselines originally designed for 3D inputs, while the remaining baselines were designed for 2D inputs. **Bold** and underlined values indicate the **best** and second-best results among compressed methods, respectively. GeoGR achieves the highest average performance retention on both 3D VLMs at comparable computational budgets.

Method	Budget		ScanQA		SQA3D	Scan2Cap		ScanRefer		Multi3DRefer		Avg.
	Tokens	TFLOPs	CIDEr	EM	EM	B-4@.5	C@.5	Acc@.25	Acc@.5	F1@.25	F1@.5	(%)
LLaVA-OneVision-7B												
Dense	6272	97.64	94.84	28.09	57.06							100.00
VisionZip (Yang et al., 2025)	640	8.52	78.54	23.14	51.41	-	-	-	-	-	-	85.10
VisPruner (Zhang et al., 2025b)	640	8.52	80.37	23.81	52.09	-	-	-	-	-	-	86.93
SeGPruner* (Li et al., 2026b)	640	8.52	85.15	25.43	53.88	-	-	-	-	-	-	91.59
VTC* (Huang et al., 2025b)	735	9.81	85.64	25.33	54.36	-	-	-	-	-	-	91.91
DTC* (Huang et al., 2025b)	702	9.36	87.22	25.81	54.78	-	-	-	-	-	-	93.30
VisionZip† (Yang et al., 2025)	640	8.52	81.36	23.59	54.33	-	-	-	-	-	-	88.34
VisPruner† (Zhang et al., 2025b)	640	8.52	84.67	25.09	56.01	-	-	-	-	-	-	92.26
SeGPruner*,† (Li et al., 2026b)	640	8.52	84.52	24.86	55.61	-	-	-	-	-	-	91.69
DTC*,† (Huang et al., 2025b)	702	9.36	86.06	25.50	54.96	-	-	-	-	-	-	92.61
GeoGR (ours)	638	8.65	**88.25**	**26.20**	**56.61**	-	-	-	-	-	-	**95.18**
Video-3D LLM-7B												
Dense	6272	97.64	98.19	27.74	57.43	40.71	80.02	57.79	51.48	56.95	51.75	100.00
VisionZip (Yang et al., 2025)	704	9.39	89.60	25.11	55.30	37.00	59.56	55.72	49.78	53.40	48.67	91.59
VisPruner (Zhang et al., 2025b)	704	9.39	88.19	24.64	54.67	36.71	58.55	55.63	49.76	53.01	48.39	90.74
SeGPruner* (Li et al., 2026b)	640	8.52	88.91	25.07	55.10	36.04	57.99	56.11	49.96	53.99	49.11	91.30
VTC* (Huang et al., 2025b)	708	9.44	89.14	25.26	54.16	36.36	57.07	55.92	49.65	52.67	47.89	90.56
DTC* (Huang et al., 2025b)	704	9.39	87.06	24.11	54.11	36.60	58.28	55.50	49.45	51.57	47.00	89.55
VisionZip† (Yang et al., 2025)	704	9.39	93.30	26.42	**56.72**	38.78	70.42	56.69	50.37	55.76	50.71	96.01
VisPruner† (Zhang et al., 2025b)	704	9.39	92.60	26.31	56.49	39.23	70.15	56.52	50.21	55.86	50.66	95.87
SeGPruner*,† (Li et al., 2026b)	640	8.52	94.23	**26.46**	55.90	38.77	70.66	56.33	50.11	55.62	50.41	95.79
DTC*,† (Huang et al., 2025b)	704	9.39	91.28	25.67	56.29	39.13	70.98	56.65	50.39	55.41	50.29	95.41
GeoGR (ours)	671	9.09	**94.85**	26.22	56.46	**39.45**	**73.27**	**56.87**	**50.69**	**55.89**	**50.84**	**96.80**

particularly valuable for preserving cross-view evidence on LLaVA-OneVision, consistent with the stronger results of 3D-native baselines on this model. GeoGR achieves strong performance retention on both 3D VLMs, supporting its applicability to models with different scene representations.

Efficiency Analysis. As shown in Table 3, GeoGR speeds up LLM prefill by $6.8\times$ on LLaVA-OneVision and $6.1\times$ on Video-3D LLM, with TTFT speedups of $2.4\times$ and $2.3\times$. It reduces the visual token KV cache from 359.7 MB to 36.6 and 38.5 MB, respectively. Compression overhead is critical to realizing these gains. DTC spends over 2000 ms in the projector stage, outweighing its prefill savings, whereas GeoGR takes only 99.0 and 75.6 ms. These efficiency gains should also be considered alongside the performance retained after compression. SeGPruner is faster, but GeoGR retains more performance, especially when a 2D VLM is used for 3D scene understanding. On LLaVA-OneVision, GeoGR retains 95.2% of performance compared with 91.7% for SeGPruner, a gain of 3.5 percentage points for an additional 56.5 ms of TTFT. These results show a favorable trade-off between performance retention and inference efficiency.

Table 3: **Efficiency comparison on two 3D VLMs.** Parentheses indicate speedups relative to the dense model. "Projector" covers all operations between the vision encoder and the LLM, including token compression. Encoder latency is similar across methods and is omitted from the breakdown but included in TTFT. GeoGR offers a favorable trade-off between performance retention and inference efficiency on both 3D VLMs.

Method	KV Cache (MB)	Latency (ms)			Avg. (%)
		Projector	LLM Prefill	TTFT	
LLaVA-OneVision-7B					
Dense	359.7	53.3	1021.4 (1.0×)	1426.7 (1.0×)	100.0
SeGPruner†	36.7	100.8	103.0 (9.9×)	548.5 (2.6×)	91.7
DTC†	40.3	2025.0	147.2 (6.9×)	2522.2 (0.6×)	92.6
GeoGR	36.6	99.0	151.2 (6.8×)	605.0 (2.4×)	95.2
Video-3D LLM-7B					
Dense	359.7	65.2	874.1 (1.0×)	1257.8 (1.0×)	100.0
SeGPruner†	36.7	76.4	86.4 (10.1×)	479.7 (2.6×)	95.8
DTC†	40.3	2301.1	132.4 (6.6×)	2750.6 (0.5×)	95.4
GeoGR	38.5	75.6	143.4 (6.1×)	537.4 (2.3×)	96.8

5.3 Ablation studies

We conduct ablation studies on LLaVA-OneVision-7B to examine the components of GeoSemZip, the grouping signal, and recoverable skipping. Avg. reports performance retention relative to the dense post-trained model (100%). Additional ablations on voxel size, saliency aggregation, the dominant-token ratio, and the visual computation window are provided in Appendix D.

GeoSemZip Component Analysis. Table 4 evaluates GeoSemZip at a fixed 30% token budget. Spatial coverage alone retains 93.53% of performance, while semantic saliency alone reaches 96.99%. Combining both branches with voxel consolidation raises retention to 98.19%, supporting the joint use of salient content and spatial coverage. Residual merging adds 1.14 points, reaching 99.33% without increasing the token budget. This gain highlights the value of aggregating unselected voxel features into contextual anchors rather than discarding them.

Table 4: **Component ablation of GeoSemZip.** All variants retain 30% of the visual tokens and are post-trained independently from the same initialization. The semantic-only variant omits cross-view voxel consolidation. Residual merging requires contextual anchors from the spatial coverage branch.

Semantic Saliency	Spatial Coverage	Residual Merge	ScanQA CIDEr	ScanQA EM	SQA3D EM	Avg.
	✓		85.97	26.16	55.24	93.53%
✓			92.04	26.50	56.81	96.99%
✓	✓		92.67	26.95	57.57	98.19%
✓	✓	✓	93.95	27.51	57.63	**99.33%**

Grouping Signal. Table 5 compares projector-derived and query-conditioned grouping under the same execution schedule. The former reuses the static ordering from GeoSemZip, while the latter ranks visual tokens at each anchor using attention from the final prompt token. Query-conditioned grouping improves all three metrics and raises average performance retention from 92.73% to 95.18%. This gain supports updating token priorities according to the question and current LLM layer rather than reusing the projector-derived ordering.

Table 5: **Grouping signal** with anchor layers 9 and 13 and $\kappa = 0.5$. Both variants share the same checkpoint, schedule, and decoding protocol and differ only in the ranking used to form the groups.

Grouping Signal	ScanQA CIDEr	ScanQA EM	SQA3D EM	Avg.
Projector-derived	85.34	25.30	55.98	92.73%
Query-conditioned	88.25	26.20	56.61	**95.18%**

Skipping vs. Pruning. Table 6 compares recoverable group-wise skipping with permanent pruning at a matched average token budget. We evaluate pruning at layer 9, layer 13, or both layers. With two anchors, tokens removed at layer 9 remain unavailable at layer 13. At each anchor, skipping selects half of the visual tokens for subsequent computation while preserving the hidden states of the others. At layer 13, all visual tokens are rescored, allowing previously skipped tokens to rejoin the active group. Pruning uses higher keep ratios to match the average number of active visual tokens per LLM layer. Single-anchor pruning retains 94.65% of performance at layer 9 and 95.03% at layer 13, while pruning at both anchors retains 94.73%. Skipping at layers 9 and 13 achieves the highest retention of 95.18%, outperforming all pruning configurations. These results support the observation in Section 3 that low token relevance can be temporary and that preserving tokens for later reactivation is beneficial.

Table 6: **Skipping vs. pruning within the window** $[8, 23]$**.** All variants use the same window-adapted checkpoint without further training and average 638 active visual tokens per LLM layer.

Method	Anchors	ScanQA CIDEr	ScanQA EM	SQA3D EM	Avg.
Pruning	9	87.66	25.97	56.52	94.65%
	13	87.20	26.07	57.23	95.03%
	9, 13	87.82	25.88	56.75	94.73%
Skipping	9, 13	88.25	26.20	56.61	**95.18%**

6 Conclusion

We presented GeoGR, a two-stage compression framework for multi-view 3D VLMs that addresses redundancy in scene representations and visual computation. GeoSemZip consolidates cross-view observations while preserving salient content, spatial coverage, and information from unselected regions. Guided by effective visual computation windows and locally stable but evolving token relevance, GroupRoute combines Late Entry and Early Exit with Recoverable Group-wise Skipping, allowing temporarily inactive tokens to resume computation. A compact, selected dataset supports compression-aware post-training. Experiments demonstrate that GeoGR achieves a favorable performance–efficiency trade-off across 3D VLMs with and without explicit 3D spatial encoding.

AI USE STATEMENT

We used generative AI tools for translation, language editing, and figure preparation based on author-provided code and data. The authors manually reviewed and revised all AI-assisted content for accuracy and consistency with the study. We take full responsibility for the final manuscript, including its text, figures, scientific claims, and any other materials prepared with the assistance of generative AI.

REFERENCES

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)*, pp. 19107–19117. IEEE, 2022.

Jinhe Bi, Zengjie Jin, Yifan Wang, Danqi Yan, Wenke Huang, Xiaowen Ma, Sikuan Yan, Artur Hecker, Mang Ye, Xun Xiao, et al. Prism: Self-pruning intrinsic selection method for training-free multimodal data selection. *arXiv preprint arXiv:2502.12119*, 2025.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European conference on computer vision*, pp. 202–221. Springer, 2020.

Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3193–3203, 2021.

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.

H.-W. Huang, W. Chai, K.-M. Chen, C.-Y. Yang, and J.-N. Hwang. ToSA: Token merging with spatial awareness, 2025a.

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)*, pp. 19424–19434. IEEE, 2025b.

Wencan Huang, Daizong Liu, and Wei Hu. Fast3d: Accelerating 3d multi-modal large language models for efficient 3d scene understanding. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10935–10944, 2025c.

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

Han Li, Zehao Huang, Jiahui Fu, Naiyan Wang, and Si Liu. Geometry-guided 3d visual token pruning for video-language models. *arXiv preprint arXiv:2604.18260*, 2026a.

Hongliang Li, Jiaxin Zhang, Wenhui Liao, Dezhi Peng, Kai Ding, and Lianwen Jin. Redundancylens: Revealing and exploiting visual token processing redundancy for efficient decoder-only mllms. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 24056–24067, 2025.

Wenli Li, Kai Zhao, Haoran Jiang, Enquan Yang, Yi Su, and Dan Zeng. Segpruner: Semantic-geometric visual token pruner for 3d question answering. *arXiv preprint arXiv:2603.29437*, 2026b.

Peng Ling, Yingda Yin, Lingting Zhu, Weikai Chen, Shengju Qian, Zeyu Hu, Xin Wang, and Wenming Yang. Coverprune: Coverage-driven token pruning for 3d vlms via optimal transport. In *European Conference on Computer Vision*, pp. 499–516. Springer, 2026.

Wenjie Liu, Hao Wu, Xin Qiu, Xudong Wang, Yingqi Fan, Yihan Zhang, Anhao Zhao, Yunpu Ma, and Xiaoyu Shen. Vica: Efficient multimodal llms with vision-only cross-attention. *arXiv preprint arXiv:2602.07574*, 2026.

Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022.

Jiuhe Qu, Yingping Liang, and Ying Fu. Hisc: Hierarchical spatial clustering token compression for efficient 3d scene understanding. *arXiv preprint arXiv:2608.04610*, 2026.

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)*, pp. 22857–22867. IEEE, 2025.

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*, volume 39, pp. 7284–7292, 2025.

Hao Wu, Yingqi Fan, Dai Jinyang, Junlong Tong, Yunpu Ma, and Xiaoyu Shen. Hidrop: Hierarchical vision token reduction in mllms via late injection, concave pyramid pruning, and early exit. In *International Conference on Learning Representations*, volume 2026, pp. 97296–97316, 2026a.

Hao Wu, Junlong Tong, Xudong Wang, Yang Tan, Changyu Zeng, Anastasia Antsiferova, and Xiaoyu Shen. From data to model: A survey of the compression lifecycle in mllms. *TechRxiv*, 2026b.

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.

Minhao Xiong, Zichen Wen, Zhuangcheng Gu, Xuyang Liu, Rui Zhang, Hengrui Kang, Jiabing Yang, Junyuan Zhang, Weijia Li, Conghui He, et al. Prune2drive: A plug-and-play framework for accelerating vision-language models in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25215–25224, 2026.

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)*, pp. 19792–19802. IEEE, 2025.

Qianhao Yuan, Qingyu Zhang, Yanjiang Liu, Jiawei Chen, Yaojie Lu, Hongyu Lin, Jia Zheng, Xianpei Han, and Le Sun. Shortv: Efficient multimodal large language models by freezing visual tokens in ineffective layers. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 329–339. IEEE, 2025.

Weili Zeng, Ziyuan Huang, Kaixiang Ji, and Yichao Yan. Skip-vision: Efficient and scalable acceleration of vision-language models via adaptive token skipping. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21384–21397. IEEE, 2025.

Kai Zhang, Xingyu Chen, and Xiaofeng Zhang. Adatoken-3d: dynamic spatial gating for efficient 3d large multimodal-models reasoning. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 16702–16709. IEEE, 2025a.

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)*, pp. 20857–20867. IEEE, 2025b.

Yiming Zhang, ZeMing Gong, and Angel X Chang. Multi3drefer: Grounding text description to multiple 3d objects. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15179–15179. IEEE, 2023.

Zeliang Zhang, Phu Pham, Wentian Zhao, Kun Wan, Yu-Jhe Li, Jianing Zhou, Daniel Miranda, Ajinkya Kale, and Chenliang Xu. Treat visual tokens as text? but your mllm only needs fewer efforts to see. *arXiv preprint arXiv:2410.06169*, 2024.

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)*, pp. 8995–9006. IEEE, 2025.

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

This appendix supplements the main text with implementation and experimental details, additional results, and analysis:

- Section A: additional experimental details, including the training and evaluation protocol and the exact GroupRoute execution schedule.
- Section B: implementation details, including BARS anchor selection.
- Section C: experimental setup, including the two 3D VLM backbones and the benchmark datasets.
- Section D: additional experimental results, including post-training data selection, projector-level token-budget comparisons, GeoSemZip parameter sensitivity, visual-computation window selection, and anchor keep-ratio allocation.
- Section E: additional analysis, including layer-pair visual-token ranking and top-200 overlap heatmaps.

A Additional Experimental Details

A.1 Training and Evaluation Protocol

Input construction. For each of the 32 sampled RGB views, we bilinearly pool projector features and vision-encoder attention scores to a 14×14 grid. GeoSemZip reconstructs world coordinates from aligned depth maps, camera intrinsics, and camera-to-world poses.

Evaluation. Evaluation metrics are summarized in Table 9. For ScanQA, predictions and references are lowercased, processed with the benchmark caption-text cleaner, whitespace-normalized, and stripped of trailing punctuation. SQA3D EM follows benchmark-specific normalization: case and punctuation normalization, digit-to-word conversion, article removal, and spelling correction. A prediction is correct if its normalized form matches any normalized reference answer. Evaluation uses batch size 1.

Post-training. For both 3D VLMs, compression-aware post-training consists of projector adaptation followed by late-entry/early-exit adaptation. Each stage runs for one epoch with a global batch size of 16 and learning rate 10^{-5}. The second stage starts from the projector-adapted checkpoint. The vision encoder remains frozen in both stages; the projector is updated only in the first stage, while the LLM backbone is updated in both. We use zero weight decay, cosine learning rate decay, a warm-up ratio of 0.03, BF16 precision, and DeepSpeed ZeRO-3 in both stages.

A.2 Exact GroupRoute Execution Schedule in the LLM Decoder

Table 7 summarizes the GroupRoute execution schedule under the adopted LLaVA-OneVision configuration.Visual tokens participate in computation from blocks 8 through 23.The schedule uses grouping anchors at blocks 9 and 13, both with keep ratio $\kappa = 0.5$. An anchor block processes the complete visual token sequence and recomputes the partition based on its block-input representations, which then governs subsequent blocks until the next anchor. "All" denotes the full token sequence, including text, visual tokens; $\mathcal{H}_a$ denotes the visual content tokens selected at anchor a.

Table 7: **GroupRoute execution schedule on LLaVA-OneVision.**

Blocks	Active sequence	Event
1–7	Text only	Visual tokens not yet injected
8	All	Visual tokens enter
9	All	Score and group at anchor 9
10–12	Text, $\mathcal{H}_9$	$\mathcal{L}_9$ bypasses
13	All	Anchor 13; score and regroup
14–23	Text, $\mathcal{H}_{13}$	$\mathcal{L}_{13}$ bypasses
24–28	Text only	All visual tokens have exited

Blocks 8 and 9 both process all retained visual tokens. The anchor-9 partition governs blocks 10–12, where only $\mathcal{L}_9$ bypasses computation; block 13 then processes all visual tokens and forms a new partition. The anchor-13 partition governs blocks 14–23, where only $\mathcal{L}_{13}$ bypasses computation. Blocks 1–7 and 24–28 contain no visual tokens because of late entry and early exit.

B More Implementation Details

B.1 Budget-Aware Anchor Selection (BARS)

BARS determines the number and positions of regrouping anchors under fixed GeoSemZip configurations, visual computation windows, and token keep ratios. The algorithm first builds a risk and cost model based on attention scores. It then generates candidate schedules for all feasible anchor numbers and selects the optimal schedule from these candidates following a screening rule.

Query-Guided Visual Token Grouping. We sample some of examples from each of the $D = 5$ datasets; $\mathcal{S}_d$ denotes the examples used from dataset d. Inputs follow inference-time frame sampling, feature ordering, prompt construction, and positional encoding. The query is the last prompt token for question answering and captioning, or the grounding token for localization. Let $\mathcal{V}_x$ contain the positions of the N_x visual content tokens in the complete prompt. At layer ℓ, we compute and normalize their head-averaged attention scores as

$$s_{x,\ell,i} = \frac{1}{H_q} \sum_{h=1}^{H_q} \mathrm{softmax}\Big(\mathbf{q}_{x,\ell,h}\mathbf{K}_{x,\ell,h}^{\top}/\sqrt{d_h}\Big)_i,$$
$$p_{x,\ell,i} = s_{x,\ell,i} \Big/ \sum\nolimits_{j\in\mathcal{V}_x} s_{x,\ell,j}, \qquad i \in \mathcal{V}_x.$$

Here H_q and d_h are the query-head count and head dimension, and $\mathbf{K}_{x,\ell,h}$ contains all causally visible prompt keys.Stable descending sorting of $p_{x,\ell,i}$ definesrank$_{x,\ell}(i)$. At anchor a, the top$\lceil \kappa N_x \rceil$ tokens form $\mathcal{H}_{x,a}$; the rest form$\mathcal{L}_{x,a}$. Independently ranking layer ℓ gives$\mathcal{H}_{x,\ell}$, used to assess whether the anchor partition has become stale.

Segment risk. Reusing the partition from anchor a at layer ℓ incurs three proxy risks:

$$R_{\mathrm{mass},x}(a,\ell) = \sum\nolimits_{i\in\mathcal{L}_{x,a}} p_{x,\ell,i},$$
$$R_{\mathrm{mig},x}(a,\ell) = \frac{|\mathcal{L}_{x,a} \cap \mathcal{H}_{x,\ell}|}{|\mathcal{H}_{x,a}|},$$
$$R_{\mathrm{rank},x}(a,\ell) = \frac{1}{N_x^2} \sum\nolimits_{i\in\mathcal{V}_x} |\,\mathrm{rank}_{x,a}(i) - \mathrm{rank}_{x,\ell}(i)|.$$

These terms measure the bypassed attention mass, high-priority tokens missed by the anchor partition, and normalized rank displacement, respectively.

In segment $[a, b]$, block a processes all visual tokens, while blocks $a+1, \ldots, b$ reuse its partition; b is the next anchor or $\ell_{\mathrm{out}} + 1$. The segment risk is

$$R_x(a,b) = \sum\nolimits_{\ell=a+1}^{b-1} w(\ell)\big[\alpha R_{\mathrm{mass},x}(a,\ell) + \beta R_{\mathrm{mig},x}(a,\ell) + \gamma R_{\mathrm{rank},x}(a,\ell)\big],$$

where $w(\ell) = (\ell_{\mathrm{out}} - \ell + 1)^p$ assigns more weight to earlier skipped blocks. We use $\alpha = \beta = 0.35$, $\gamma = 0.2$, and $p = 1$. For each dataset, we average over its sampled examples, $R_d(a,b) = |\mathcal{S}_d|^{-1} \sum_{x\in\mathcal{S}_d} R_x(a,b)$, and then average equally across datasets, $R(a,b) = D^{-1} \sum_{d=1}^{D} R_d(a,b)$.

Visual token–layer budget. Let C_x^{full} and C_x^{group} denote the actual numbers of active visual-sequence positions in a full and grouped block for example x. For schedule $\mathcal{A} = \{a_1 < \cdots < a_m\}$ over the window $[\ell_{\mathrm{in}}, \ell_{\mathrm{out}}]$, the prefix before a_1 and the m anchors execute all visual tokens; the remaining blocks use the groups. Thus,

$$C_x(\mathcal{A}) = (a_1 - \ell_{\mathrm{in}} + m)C_x^{\mathrm{full}} + (\ell_{\mathrm{out}} - a_1 + 1 - m)C_x^{\mathrm{group}}.$$

We average costs within and across datasets, then normalize by the corresponding full-window cost $(\ell_{\mathrm{out}} - \ell_{\mathrm{in}} + 1)C_x^{\mathrm{full}}$ to obtain $\Gamma(\mathcal{A})$.

Candidate schedules by dynamic programming. A schedule $\mathcal{A} = (a_1 < \cdots < a_m)$ is scored by its total segment risk $\sum_{j=1}^{m} R(a_j, a_{j+1})$, where $a_{m+1} = \ell_{\mathrm{out}} + 1$ closes the last segment and the blocks before a_1 carry cost but no skipping risk. The cost is simpler than the risk: the full-visual blocks are the $a_1 - \ell_{\mathrm{in}}$ prefix blocks and the m anchors, so Γ is determined by a_1 and m alone. The

Algorithm 1 Budget-constrained Anchor and Regrouping Search (BARS)

Input: Sampled examples $\{\mathcal{S}_d\}_{d=1}^{D}$; fixed checkpoint and GeoSemZip configuration
Input: Visual window $\mathcal{W}$, keep ratio κ, budget B, anchor limit $M_{\max}$, gap g, threshold τ
Output: Selected anchor schedule $\mathcal{A}^*$

for all sampled examples and layers in $\mathcal{W}$ **do**
    Collect query-to-visual attention; normalize over visual content tokens
    Store stable token ranks and the high- and low-priority groups
end for
for all legal segments (a, b) **do**
    Evaluate attention mass, group migration, and rank displacement on intervening blocks
    Apply depth weights; average within datasets to obtain $\{R_d(a, b)\}$
    Average across datasets to obtain $R(a, b)$; compute the segment cost
end for
Initialize DP states for possible first anchors with full-prefix costs and zero risk
for all reachable DP states in increasing boundary order **do**
  for all legal next segments **do**
      Extend the state with the segment cost, risk, and one additional anchor
    if the extended state satisfies the budget and anchor limit **then**
        Keep the lowest-risk predecessor for its boundary, cost, and anchor count
    end if
  end for
end for
for each feasible anchor count $m = 1, \ldots, M_{\max}$ **do**
    Backtrack the lowest-risk terminal state to obtain $\mathcal{A}_m$
    Sum its segment risks for each dataset to obtain $\{R_d(\mathcal{A}_m)\}$
end for
$\mathcal{A}^* \leftarrow \mathcal{A}_1$
for each larger feasible anchor count m, in increasing order **do**
  if $R(\mathcal{A}_m) < R(\mathcal{A}^*)$ and every dataset's relative risk reduction is at least τ **then**
      $\mathcal{A}^* \leftarrow \mathcal{A}_m$
  end if
end for
return $\mathcal{A}^*$

budget is therefore a closed-form condition on (a_1, m) instead of a quantity accumulated along the schedule, and for each anchor count $m \le M_{\max}$ the search minimizes risk over a feasible set $\mathcal{F}_m$:

$$\mathcal{A}_m = \operatorname*{arg\,min}_{\mathcal{A} \in \mathcal{F}_m} \sum_{j=1}^{m} R(a_j, a_{j+1}), \qquad \mathcal{F}_m = \big\{\mathcal{A} :\ |\mathcal{A}| = m,\ a_{j+1} - a_j \ge g,\ \Gamma(a_1, m) \le B\big\}.$$

Let $V(\ell, j)$ denote the smallest accumulated risk of placing j anchors with the last one at block ℓ. With the segment risks precomputed, $V(a_1, 1) = 0$ for every budget-feasible a_1, and

$$V(\ell', j+1) = \min_{\ell \le \ell' - g} \big[V(\ell, j) + R(\ell, \ell')\big].$$

Minimizing $V(\ell, m) + R(\ell, \ell_{\text{out}} + 1)$ over the last anchor and backtracking gives $\mathcal{A}_m$ and its per-dataset risk $R_d(\mathcal{A}_m) = \sum_j R_d(a_j, a_{j+1})$. The window spans at most 17 blocks and $M_{\max} = 4$.

Selecting the anchor count. Starting from the feasible one-anchor candidate $\mathcal{A}_1$, BARS examines larger feasible counts in increasing order. A candidate $\mathcal{A}_m$ replaces the current selection $\mathcal{A}^*$ only when

$$R(\mathcal{A}_m) < R(\mathcal{A}^*), \qquad R_d(\mathcal{A}_m) \le (1 - \tau) R_d(\mathcal{A}^*) \quad \text{for every } d,$$

where $R(\mathcal{A}) = D^{-1} \sum_d R_d(\mathcal{A})$ and $\tau = 5\%$.

Both 3D VLMs share the search parameters: keep ratio $\kappa = 0.5$, minimum gap $g = 2$, anchor limit $M_{\max} = 4$, and margin $\tau = 5\%$, with no maximum-span constraint. They differ only in the visual computation window, $[8, 23]$ for LLaVA-OneVision and $[8, 24]$ for Video-3D LLM. The optimal anchor schedule for both 3D VLMs is $[9, 13]$. The schedule is fixed at inference, while the token groups are recomputed for each query. Algorithm 1 summarizes the overall procedure.

C More Experimental Setup Details

C.1 3D VLMs

LLaVA-OneVision-7B. LLaVA-OneVision (Li et al., 2024) provides a unified framework for single-image, multi-image, and video understanding. Its SigLIP visual encoder extracts patch-level visual features, which are further projected into the language embedding space via a two-layer MLP. Each sampled scene view is independently encoded, and the output visual tokens are concatenated with textual instructions to form the multimodal input sequence. Under our 32-view setting, each view yields a 14×14 pooled visual grid, producing $32 \times 14 \times 14 = 6{,}272$ visual content tokens before compression. This 3D VLM takes only RGB-based visual features as input, while depth maps and camera calibration parameters are exclusively fed into our GeoSemZip module.

Video-3D LLM-7B. Built upon LLaVA-Video-7B, Video-3D LLM (Zheng et al., 2025) explicitly integrates 3D positional information into multi-view visual representations. Specifically, depth maps, camera intrinsics, and camera poses project observations from different views into a unified world coordinate system. The obtained patch-level spatial coordinates are encoded and fused with visual features, endowing the model with robust spatial awareness. In addition to standard visual question answering, Video-3D LLM supports spatial reasoning tasks, including object-conditioned captioning and visual grounding. For object-conditioned captioning, the center coordinates of target objects are adopted to specify query instances. For visual grounding, the hidden representations of a dedicated grounding token are matched with object-level features to locate target objects.

Table 8: **Architectures and scene inputs of the evaluated 3D VLMs.** Token counts use the 32-view setting and exclude formatting tokens. Q/KV denotes query and key/value attention heads.

Component	LLaVA-OneVision-7B	Video-3D LLM-7B
Pretrained VLM family	LLaVA-OneVision	LLaVA-Video
Vision encoder	SigLIP SO400M/14	SigLIP SO400M/14
Encoder input resolution	384×384	384×384
Visual feature dimension	1,152	1,152
Multimodal projector	Two-layer MLP, GELU	Two-layer MLP, GELU
LLM	Qwen2-7B	Qwen2-7B
Decoder blocks	28	28
Attention heads (Q/KV)	28 / 4	28 / 4
Hidden / FFN dimension	3,584 / 18,944	3,584 / 18,944
Scene view sampling	32 uniform RGB views	32 uniform RGB views
Pooled grid per view	14×14	14×14
Uncompressed content tokens	6,272	6,272
Role of depth and poses	Geometric compression	3D encoding and compression
Evaluated task interfaces	Question answering	Question answering, captioning, grounding

C.2 Benchmarks and Datasets

The five ScanNet-based datasets serve both as sources of post-training examples and as benchmarks for 3D question answering, object captioning, and language grounding. Table 9 reports, for each dataset, the number of examples in our full post-training pool before PRISM selection and the separate split used for evaluation. The full pool contains 223,128 examples across the five sources; its selected subset is reported in Table 1. LLaVA-OneVision is evaluated on the two question-answering benchmarks, whereas Video-3D LLM is evaluated on all five.

ScanQA. ScanQA (Azuma et al., 2022) contains human-written questions and answers grounded in indoor 3D scenes. Questions require identifying objects, recognizing their attributes, and reasoning about their spatial relationships.We report CIDEr for generated-answer agreement and normalized exact match for answer correctness on the validation split.

SQA3D. SQA3D (Ma et al., 2022) adds a textual description of an agent's situation, including its position and orientation, to a question about the surrounding scene. Answering requires interpreting

Table 9: **Training sources and evaluation splits for five ScanNet-based benchmarks.** Train counts examples before PRISM selection; Eval counts held-out examples. Evaluation and scene counts follow the Video-3D LLM annotations (Zheng et al., 2025).

Benchmark	Train	Eval split	Eval	Scenes	Reported metrics
ScanQA (Azuma et al., 2022)	26,515	Validation	4,675	71	CIDEr, normalized EM
SQA3D (Ma et al., 2022)	79,445	Test	3,519	67	Normalized EM
Scan2Cap (Chen et al., 2021)	36,665	Validation	2,068	141	BLEU-4, CIDEr (IoU 0.5)
ScanRefer (Chen et al., 2020)	36,665	Validation	9,508	141	Acc@0.25, Acc@0.5
Multi3DRefer (Zhang et al., 2023)	43,838	Validation	11,120	141	F1@0.25, F1@0.5

the question from that situated perspective. The benchmark therefore tests whether the compressed representation preserves scene layout and spatial relations as well as object appearance. We evaluate the test split using normalized EM, with the situation description and question supplied together in the task prompt.

Scan2Cap. Scan2Cap (Chen et al., 2021) evaluates dense captioning in 3D scenes, using object descriptions from ScanRefer. A caption must describe the localized object and may refer to its attributes and surrounding objects. For Video-3D, the target location is supplied through its coordinate-conditioned captioning prompt. We report BLEU-4 and CIDEr at a 3D IoU threshold of 0.5,which couples language quality with adequate object localization.

ScanRefer. ScanRefer (Chen et al., 2020) asks a model to localize one object in a 3D scene from a natural-language referring expression. Descriptions use appearance and relational cues to distinguish the target from nearby objects. We report the fraction of predictions whose 3D bounding-box intersection over union with the target reaches 0.25 or 0.5, denoted Acc@0.25 and Acc@0.5.

Multi3DRefer. Multi3DRefer (Zhang et al., 2023) extends language grounding to expressions that may refer to zero, one, or multiple objects. A model must identify the appropriate set of objects and handle descriptions with distractors or no valid target. We use the benchmark's set-level F1 metrics at 3D IoU thresholds of 0.25 and 0.5, including its zero-target evaluation convention.

D Additional Experimental Results

D.1 Effect of Data Selection on Post-Training Performance

Table 10 provides per-metric evaluation results for our data selection ablation study. For each 3D VLM, we compare two sets of dense checkpoints: one post-trained on the complete five-source dataset, and the other post-trained on the PRISM-selected subset (approximately 30% from each source). Retention is defined as the unweighted mean of the ratios between the results obtained on the selected subset and those on the full dataset across evaluation metrics of the corresponding model. Our results show that using this 30% subset maintains strong overall performance: Video-3D LLM achieves 98.72% performance retention, while LLaVA-OneVision retains 95.69% of the full-data post-trained performance.

D.2 GeoSemZip Budget Selection and Baseline Comparison

We examine GeoSemZip at 20%, 30%, and 40% to select the Stage-I budget (Table 11). We compare post-trained projector-level compression methods on LLaVA-OneVision at target budgets of approximately 10%, 20%, and 30% of the original 6,272 visual tokens (Table 12).

Table 10: **Dense post-training on full data and PRISM-selected data.** PRISM retains approximately 30% from each source; retention averages the metric-wise ratios to the full-data post-trained checkpoint.

(a) LLaVA-OneVision-7B

Training data	ScanQA CIDEr	ScanQA EM	SQA3D EM	Retention (%)
Full set (223,128)	100.27	29.07	59.53	100.00
PRISM 30% (66,940)	94.84	28.09	57.06	95.69

(b) Video-3D LLM-7B

Training data	ScanQA		SQA3D	Scan2Cap		ScanRefer		Multi3DRefer		Retention (%)
	CIDEr	EM	EM	B-4@.5	C@.5	Acc@.25	Acc@.5	F1@.25	F1@.5	
PRISM 30% (66,940)	98.19	27.74	57.43	40.71	80.02	57.79	51.48	56.95	51.75	98.72
Full set (223,128)	99.51	28.72	57.71	39.12	80.66	59.88	53.39	57.59	52.50	100.00

Choice of the 30% budget. Table 11 isolates GeoSemZip's Stage-I accuracy–token trade-off. The 30% setting retains 99.33% of Dense performance with a mean of 1,881.73 visual tokens, compared with 97.26% retention at 20%. Increasing the budget to 40% requires 593.74 more tokens and yields a lower retention of 98.38%. We therefore use the post-trained 30% checkpoint as the input to GroupRoute.

Table 11: **GeoSemZip budget selection on LLaVA-OneVision.** Each budget uses a separately post-trained checkpoint. Vis. tokens is the ScanQA mean entering the LLM; Avg. is retention relative to Dense.

Budget	Vis. tokens	Avg. (%)
$\approx 20\%$	1255.00	97.26
$\approx 30\%$	1881.73	**99.33**
$\approx 40\%$	2475.47	98.38

Table 12: **GeoSemZip and projector-level baselines across token budgets on LLaVA-OneVision.** All reported configurations are post-trained. Vis. tokens is the ScanQA mean entering the LLM; Avg. is retention relative to Dense. **Bold** marks the highest reported Avg. within each budget group.

Method	Vis. tokens	ScanQA		SQA3D	Avg. (%)
		CIDEr	EM	EM	
Dense	6272.00	94.84	28.09	57.06	100.00
Token budget $\approx 10\%$					
VisionZip	640.00	81.36	23.59	54.33	88.34
VisPruner	640.00	84.67	25.09	56.01	92.26
SeGPruner	640.00	84.52	24.86	55.61	91.69
DTC	702.00	86.06	25.50	54.96	92.61
GeoSemZip (ours)	628.00	87.44	25.41	56.69	**94.00**
Token budget $\approx 20\%$					
VisionZip	1248.00	88.10	25.97	56.12	94.57
VisPruner	1248.00	86.29	25.24	56.78	93.45
SeGPruner	1248.00	87.43	25.73	57.26	94.71
DTC	1285.00	87.99	25.41	57.00	94.38
GeoSemZip (ours)	1255.00	90.01	26.78	57.94	**97.26**
Token budget $\approx 30\%$					
VisionZip	1856.00	91.27	27.40	56.86	97.81
VisPruner	2048.00	90.03	26.52	57.09	96.46
SeGPruner	1888.00	90.34	26.87	57.23	97.07
DTC	1756.67	93.25	27.25	56.72	98.25
GeoSemZip (ours)	1881.73	93.95	27.51	57.63	**99.33**

Comparison with baselines. At the approximately 10%, 20%, and 30% target budgets in Table 12, GeoSemZip achieves the highest average retention, exceeding the strongest baseline in each group by 1.39, 2.55, and 1.08 percentage points, respectively. The results show that its combination of salient-token selection and spatially distributed anchors is effective across these tested budgets.

D.3 GeoSemZip Parameter Sensitivity

Voxel size. The voxel width s sets the granularity of geometric consolidation. A small s keeps many distinct voxels and leaves part of the cross-view duplication unresolved, whereas a large s merges more aggressively and starts to fuse surfaces that belong to different objects. Table 13 sweeps s at the same 30% budget.The default $s = 0.1$, which leaves 3574 occupied voxels per scene on average,gives the best trade-off between removing duplicates and preserving object boundaries.

Table 13: **GeoSemZip voxel-size ablation at a 30% token budget.** LLaVA-OneVision variants are post-trained independently; K is the mean number of occupied voxels per scene.

Voxel size s	K	Keep ratio	ScanQA		SQA3D	Avg.
			CIDEr	EM	EM	
0.05	4778.6	30%	92.10	27.40	57.26	98.34%
0.10	3574.0	30%	93.95	27.51	57.63	**99.33%**
0.15	2520.8	30%	91.31	27.42	57.63	98.30%

Score aggregation. A voxel inherits a single score from the patches it contains, and the pooling rule decides whose score that is. Max pooling lets the best-scoring view of a region determine its rank, whereas mean pooling dilutes that view with the others, which matters because a region is often salient in only a few of the views that observe it. Table 14 confirms the effect: mean pooling costs 0.27 points of Avg. at the same token budget, so we keep max pooling.

Table 14: **Voxel-score aggregation for GeoSemZip at a 30% token budget.** LLaVA-OneVision variants are post-trained independently and differ only in the within-voxel score-pooling rule.

Aggregation	ScanQA		SQA3D	Avg. (%)
	CIDEr	EM	EM	
Mean	93.15	27.76	57.12	99.06
Max (default)	93.95	27.51	57.63	**99.33**

Budget Split. The ratio ρ partitions the budget between semantically salient dominant tokens and spatially distributed contextual anchors. Since the total token budget is independent of ρ, all configurations in Table 15 process the same number of tokens and only differ in token allocation. We observe performance degradation when ρ deviates from the optimal value in either direction. Reducing ρ to 0.75 and 0.65 decreases the average metric by 0.34 and 1.10 points, respectively, while increasing ρ to 0.95 incurs a 0.81-point drop. This indicates that most of the budget should be assigned to salient features, with a small yet non-negligible portion reserved to maintain full scene coverage.

Table 15: **GeoSemZip budget-split ablation at a 30% token budget.** On LLaVA-OneVision, independently post-trained variants differ in ρ, the fraction assigned to dominant tokens; residual merging remains enabled.

Dominant ratio ρ	ScanQA		SQA3D	Avg. (%)
	CIDEr	EM	EM	
0.65	91.46	27.40	57.46	98.23
0.75	92.89	27.66	57.37	98.99
0.85 (default)	93.95	27.51	57.63	**99.33**
0.95	92.36	27.61	57.00	98.52

D.4 Visual-Computation Window Selection

The KL interventions in Figure 4 guide the candidate boundaries before post-training. Skipping the first $K = 5$–7 blocks gives the last consecutive low-divergence region for late entry, corresponding to entry near blocks 6–8. For early exit, divergence drops between retaining visual tokens through blocks 21 and 22 and remains low from block 23 onward. We then compare nearby windows using the same input and post-training protocol.

We determine the two boundaries sequentially. First, we compare candidate late-entry layers without early exit. Entry at block 8 achieves an average performance of 93.53%, which is only 0.46 percentage points lower than the optimal entry scheme at block 5, while reducing visual computation for three additional blocks. We therefore fix the late-entry position at block 8. With the late-entry position fixed, we then compare candidate early-exit layers. Exiting after block 23 yields the highest average performance (95.43%), surpassing the schemes that exit after blocks 22 and 24 by 4.46 and 3.37 percentage points, respectively. The resulting computation window is $[8, 23]$, and we adopt the post-trained checkpoint of this window for group-wise skipping experiments on LLaVA-OneVision.

Table 16: **Visual-computation window selection on LLaVA-OneVision.** We first vary late entry with visual tokens retained through block 28, then vary early exit with entry fixed at block 8. Avg. is relative to the dense reference in Table 2.

Window	ScanQA		SQA3D	Avg.
	CIDEr	EM	EM	
Step 1: late entry (visual tokens through block 28)				
$[5, 28]$	87.60	25.56	56.27	93.99%
$[6, 28]$	87.87	25.13	56.15	93.51%
$[7, 28]$	85.37	25.41	55.61	92.65%
$[8, 28]$	86.61	25.35	56.49	93.53%
$[9, 28]$	86.88	25.33	55.98	93.30%
Step 2: early exit (late entry at block 8)				
$[8, 22]$	83.38	24.79	55.19	90.97%
$[8, 23]$	88.13	26.25	57.00	**95.43%**
$[8, 24]$	84.86	24.94	55.87	92.06%
$[8, 28]$	86.61	25.35	56.49	93.53%

D.5 Keep-Ratio Allocation Across Anchors

At each anchor a, the keep ratio κ_a specifies the fraction of visual tokens assigned to the high-priority group. The remaining tokens form the low-priority group and skip computation in the subsequent non-anchor blocks. We compare three pairs of ratios (κ_9, κ_{13}). The groups formed at blocks 9 and 13 govern blocks 10–12 and 14–23, respectively; therefore, the two ratios affect different numbers of blocks and lead to different computational costs.

Table 17: **Keep-ratio allocation at GroupRoute anchors 9 and 13.** On LLaVA-OneVision with window $[8, 23]$, "Visual tokens" is the mean active count per LLM layer; Avg. is relative to the dense model.

Keep ratios (9 / 13)	Visual tokens	ScanQA		SQA3D	Avg.
		CIDEr	EM	EM	
0.75 / 0.25	520.79	87.28	25.99	56.38	94.46%
0.50 / 0.50 (default)	638.38	88.25	26.20	56.61	**95.18%**
0.25 / 0.75	755.98	87.77	26.07	56.07	94.55%

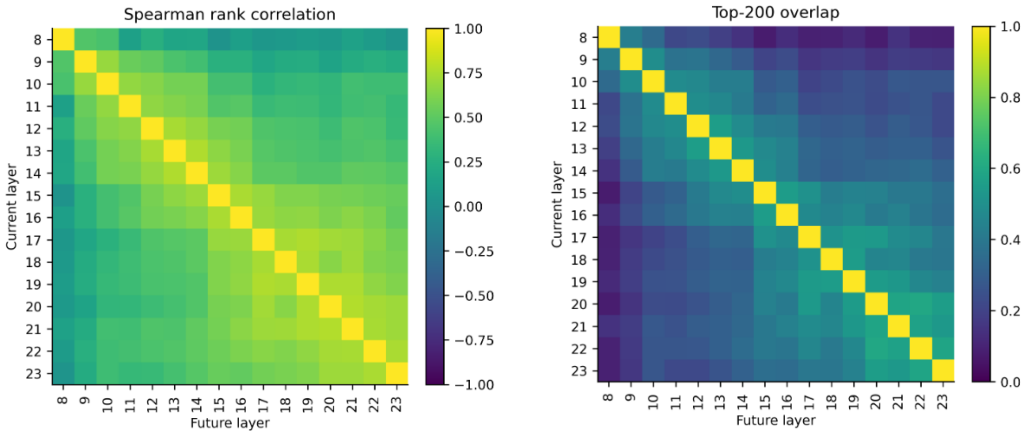
Among the tested allocations, equal keep ratios $(0.50, 0.50)$ yield the highest average retention (95.18%). The $(0.75, 0.25)$ setting uses fewer active visual tokens but retains 94.46% of performance; reversing the ratios increases the active-token count while retaining 94.55%. We therefore use $\kappa_9 = \kappa_{13} = 0.50$.

E Additional Analysis

E.1 Layer-Pair visual token Relevance Stability

We examine visual-token relevance across every pair of layers 8–23 in LLaVA-OneVision on ScanQA, complementing the layer-distance averages in Section 3. As in the main analysis, a token's relevance at each layer is its attention from the final prompt token, averaged over attention heads. Figure 5(a) measures the Spearman correlation between the complete visual-token rankings at two layers. Figure 5(b) focuses on the highest-ranked tokens: for each layer pair, it counts the visual tokens shared by their top-200 sets and divides by 200.

Both heatmaps show the strongest agreement near the diagonal, with agreement generally decreasing as the distance between layers increases. Adjacent layers thus tend to preserve the overall token ranking and share many of the same high-relevance tokens, whereas the composition of the top-200 set changes more substantially across distant layers. Taken together, these observations motivate recoverable group-wise skipping: low-priority visual tokens temporarily skip computation, and all visual tokens are regrouped at later anchors.



(a) Spearman rank correlation **(b)** Top-200 overlap

Figure 5: **Layer-pair stability of visual-token relevance on ScanQA.** For LLaVA-OneVision decoder layers 8–23, (a) gives Spearman correlation between complete visual-token rankings, and (b) gives the fraction shared by the two layers' top-200 sets.

E.2 Complementarity of Projector-Level and LLM-Level Compression

Table 18 compares projector-only, LLM-only, and two-stage compression on LLaVA-OneVision. GeoSemZip-only retains approximately 10% of projector tokens, which are processed by every decoder block. GroupRoute-only keeps all 6,272 projector tokens and uses GeoGR's window $[8, 23]$, anchors $[9, 13]$, and keep ratio $\kappa = 0.5$. GeoGR combines a 30% projector budget with the same GroupRoute schedule.

Table 18: **Projector-only, LLM-only, and two-stage compression on LLaVA-OneVision.** Vis. tokens is the mean active count per decoder block; Avg. is retention relative to the dense model.

Configuration	Proj. keep	Vis. tokens	ScanQA		SQA3D	Avg. (%)
			CIDEr	EM	EM	
GeoSemZip only	10%	628	87.44	25.41	56.69	94.00
GroupRoute only	100%	2128	88.48	26.07	55.84	94.65
GeoGR (both)	30%	638	88.25	26.20	56.61	**95.18**

GeoGR retains 95.18% of dense performance with 638 active visual tokens per decoder block. Compressing only at the projector to a similar active-token count (628) lowers retention to 94.00%, a 1.18-point difference. GroupRoute-only retains 94.65% while processing 2,128 active tokens per block. These results support the effectiveness of combining spatial–semantic token compression before the LLM with query-conditioned group-wise skipping within the LLM.