Lie Routing Vision Language Models for View-Robust 3D Understanding

Jingjun Yi[1,2]†,* **Chen Hu**[1]†, **Qi Bi**[2]✉, **Hao Zheng**[3]✉, **Huimin Huang**[3], **Yongkang Li**[2], **Wei Ji**[3], **Yawen Huang**[3], **Xian Wu**[3], **Yefeng Zheng**[1]✉
†: equal contribution; ✉: correspondence
[1]Westlake University, China, [2]University of Amsterdam, the Netherland,
[3]Tencent, China
q.bi@ieee.org, howzheng@tencent.com
zhengyefeng@westlake.edu.cn

Abstract

Multimodal Large Language Models (MLLMs) have made substantial progress in 3D scene understanding from multi-view RGB observations. Yet a fixed object and query can elicit inconsistent target-related responses across camera viewpoints, a phenomenon we characterize as viewpoint-induced semantic drift. We introduce LieVLM, a Lie routing framework for view-robust 3D understanding. Motivated by the rotational component of viewpoint change, Lie routing adapts visual representations by applying local $\mathrm{SO}(3)$ transformations to projected features within learned three-dimensional subspaces. Geometry tokens extracted by VGGT predict local rotation parameters and channel-wise routing coefficients. A closed-form Padé–Cayley map transforms projected visual features, whose identity-relative corrections are projected back and gated into the original visual and geometry features. This design retains the pretrained semantic and direct spatial context while learning view-dependent feature adjustments. Experiments on 3D captioning, grounding, detection, and spatial reasoning show consistent improvements over additive fusion across training scales and backbones. Controlled ablations examine the roles of geometric conditioning and transformation structure. Viewpoint analyses further reveal systematic variation in the predicted routing states alongside more localized target responses and stronger task performance across viewpoint changes.

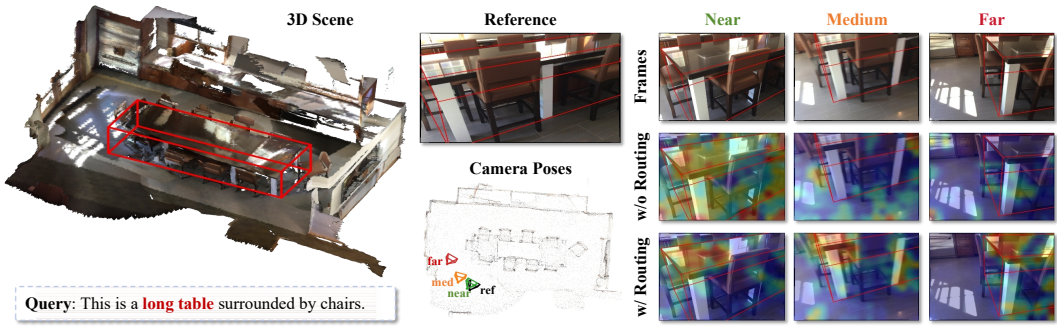
1 Introduction

3D Multimodal Large Language Models (MLLMs) extend visual-language reasoning to scene-level tasks such as object captioning, language-guided localization, and spatial question answering. Early approaches primarily connect language models to explicit 3D representations, including point clouds and object-level scene features [37, 6, 22]. These representations provide direct spatial structure, while their reliance on reconstructed scene inputs motivates complementary approaches that learn 3D understanding from RGB observations.

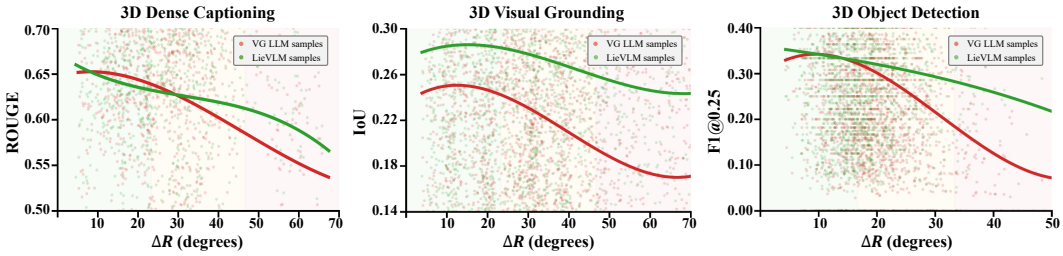
Recent 3D MLLMs derive geometry directly from multi-view RGB observations, enabling 3D understanding without requiring preconstructed 3D scene models. Pretrained geometry encoders such as VGGT [34] extract scene structure and cross-view correspondences, and recent methods inject these features into MLLM visual representations for 3D scene understanding and spatial reasoning [42, 24, 39]. This learned geometry injection paradigm preserves the scalability of video-based MLLMs while providing strong spatial context. Despite these advances, geometry is primarily

*Jingjun Yi is currently affinitied with University of Amsterdam. This work was conducted when he was affinitied with Westlake University.

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



(a) Controlled Semantic Responses across Viewpoints



(b) 3D Task Robustness across Viewpoints

Figure 1: **3D scene viewpoint variation exposes semantic instability and downstream performance degradation.** (a) Without routing, query-relevant activations become diffuse and less target centered across views, while Lie routing preserves stable target responses. Warmer regions indicate stronger support for the target phrase. (b) Fitted task trends show that performance degrades as sample-level viewpoint variation ΔR increases, while LieVLM maintains stronger performance across higher levels of viewpoint variation. For grounding, raw IoU visualizes the continuous quantity underlying Acc@0.5. Additional examples are provided in Appendix D.1.

used to augment visual representations, while how it should drive viewpoint-adaptive changes in visual semantics remains underexplored.

We observe that a fixed scene entity and query can yield substantially different target-related responses across viewpoints. As camera pose, apparent scale, and visibility change, the response may weaken, become diffuse, or shift toward irrelevant regions, a phenomenon we term *viewpoint-induced semantic drift*. We visualize it using query-relevant activations, obtained by projecting final visual tokens through the vocabulary head and averaging their positive support over the target-phrase tokens. Viewpoint change between frames i and j is measured by the $\mathrm{SO}(3)$ geodesic distance $d_{\mathrm{rot}}(i,j) = \arccos\big((\mathrm{tr}(R_i^\top R_j) - 1)/2\big)$ [16, 26]. Taking the first frame containing the referred target as reference r, we define $\Delta R = |V_s|^{-1} \sum_{t \in V_s} d_{\mathrm{rot}}(r,t)$ over the remaining views V_s, with both distances reported in degrees. Figure 1(a) varies viewpoint along a fixed camera trajectory while holding the scene, target, query, checkpoint, and reference view constant, showing that Lie routing preserves more localized responses. Figure 1(b) relates task performance to ΔR, with LieVLM maintaining stronger performance as viewpoint variation increases.

To address this challenge, we introduce LieVLM, which adapts visual semantics through geometry-conditioned Lie routing. Camera rotations are described by $\mathrm{SO}(3)$, motivating us to use the same structure for feature adaptation across viewpoints. Lie routing applies learned rotations to three-dimensional groups of projected visual features, changing their directions while preserving their lengths. Concretely, geometry tokens predict local $\mathfrak{so}(3)$ generators and channel-wise routing coefficients. A Padé–Cayley realization converts the generators into local rotations, and only the resulting identity-relative corrections are projected back into the MLLM representation. The corrections are gated and fused with the original visual features and direct geometry features before language generation. This design combines structured feature adaptation with retained semantic and spatial-context paths, without requiring relative camera poses as inputs to the routing module.

We evaluate LieVLM on 3D captioning, grounding, video object detection, and spatial reasoning [9, 4, 36, 38]. LieVLM consistently improves over matched additive geometry fusion, with gains that generalize across spatial benchmarks, training scales, and stronger multimodal backbones. Controlled

analyses further connect these improvements to structured geometry conditioned transformations and robustness to viewpoint variation.

Our contributions are summarized as follows:

- We characterize viewpoint-induced semantic drift in 3D MLLMs through cross-view target-response diagnostics and its association with downstream task performance.
- We develop Lie Routing Vision Language Models (LieVLM), combining local $\mathfrak{so}(3)$ prediction, rotation-preserving residual transformations, and channel-wise gating to adapt pretrained visual semantics across viewpoints.
- We demonstrate consistent gains across 3D scene understanding and spatial reasoning, with controlled evaluations across training scales and backbones and targeted analyses of routing structure and viewpoint robustness.

2 Related Work

3D Vision Language Models. 3D VLMs incorporate explicit spatial inputs such as point clouds and object representations [37, 6, 22, 44], or derive geometric priors from image sequences [42, 34, 35]. Reconstruction-aware models further couple video understanding with 3D reconstruction [5, 11, 20], establishing geometry as an important representation for 3D reasoning. Recent RGB-based systems additionally combine semantic and geometry encoders through camera-guided fusion [41]. Our work builds on these interfaces and studies geometry-conditioned routing of pretrained visual semantics across viewpoints.

3D Scene Understanding and Spatial Reasoning. Scene understanding benchmarks evaluate localization, grounding, and captioning [4, 9, 36]. Spatial reasoning benchmarks assess depth, distance, viewpoint, and multiview relations [38, 33, 12, 39], while instruction datasets develop these capabilities from embodied observations [3, 40, 31]. Multi-view grounding methods further aggregate view-dependent evidence through inter-view attention and learned view prototypes [14]. These efforts motivate our focus on semantic stability and task robustness under viewpoint variation.

Geometry in Neural Representations. Prior work spans conditional feature modulation through FiLM [30], explicit geometry-induced transformations in multi-view attention via GTA [28], and equivariant representations under $\mathrm{SO}(3)$ or Euclidean symmetries [10, 25, 13, 32]. Orthogonal and Cayley parameterizations further provide structure-preserving learned transformations [17]. We instead study geometry-conditioned Lie routing in pretrained 3D MLLMs, where learned geometry tokens predict token-wise local $\mathrm{SO}(3)$ actions and channel-wise gates over latent subspaces for viewpoint-adaptive routing without explicit relative pose inputs.

3 Method

3.1 Preliminaries

3D Geometry in MLLMs. Consider a video or multi-view observation $\{I_t\}_{t=1}^{T}$ of a 3D scene. A standard MLLM maps each frame into semantic visual tokens $F_t^{2d} \in \mathbb{R}^{N\times H}$, while the geometry encoder provides aligned geometry tokens $F_t^{3d} \in \mathbb{R}^{N\times H}$, where H denotes the visual token width at the MLLM interface. Existing 3D-enhanced VLMs commonly combine the two streams through additive or attention-based feature injection,

$$\widetilde{F}_t = F_t^{2d} + B(F_t^{3d}), \tag{1}$$

where $B(\cdot)$ is a learned geometry projection. The 3D perception fundamentally concerns how scene observations transform across viewpoints. Beyond supplying spatial context, our goal is to adapt visual semantics to viewpoint variation through structured latent transformations.

Lie Groups. We briefly review Lie groups, Lie algebras, and the Special Orthogonal group $\mathrm{SO}(3)$. For more in-depth details, please refer to [15, 16].

Definition 3.1 (Lie Group). A Lie group is a smooth manifold G equipped with a group operation $\odot$ such that the multiplication map $m : (x, y) \mapsto x \odot y$ and the inverse map $i : x \mapsto x^{-1}$ are smooth.

Definition 3.2 (Lie Algebra). Let G be a Lie group with identity e. Its Lie algebra is the tangent space at e, *i.e.*, $\mathfrak{g} = T_eG$, which locally linearizes G through the exponential map $\exp : \mathfrak{g} \to G$.

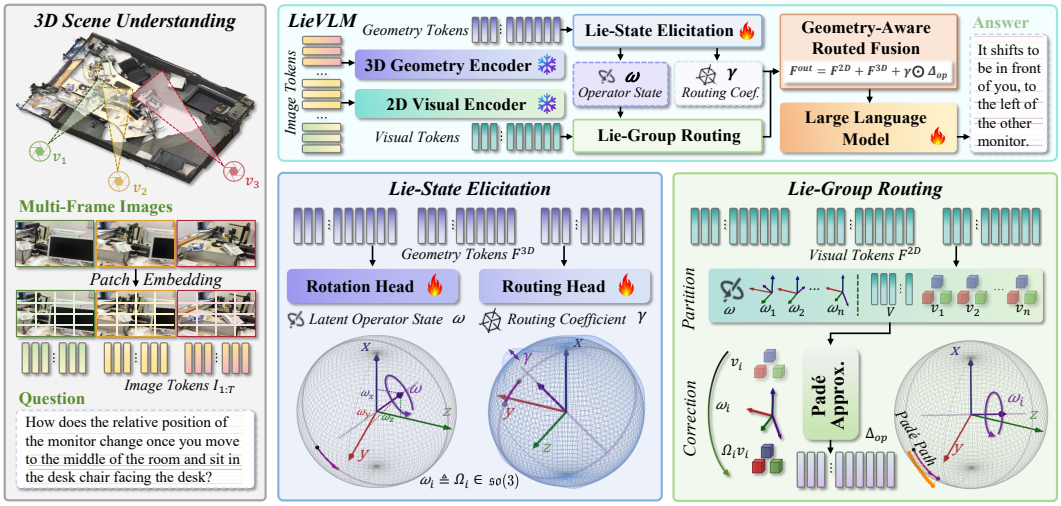


Figure 2: **Framework of the proposed LieVLM.** Lie-State Elicitation extracts latent operator states and routing coefficients from geometry tokens. Lie Group Routing transforms projected visual tokens, and routed fusion combines semantic, geometric, and operator paths before language generation.

The set of 3D rotations, known as the Special Orthogonal group $\mathrm{SO}(3)$, forms a Lie group

$$\mathrm{SO}(3) = \{R \in \mathbb{R}^{3\times 3} \mid R^\top R = I, \det(R) = 1\}, \quad \mathfrak{so}(3) = \{\Omega \in \mathbb{R}^{3\times 3} \mid \Omega^\top = -\Omega\}, \tag{2}$$

where $\mathfrak{so}(3)$ is the Lie algebra of $\mathrm{SO}(3)$, namely the space of 3×3 skew-symmetric matrices. The standard hat operator $(\cdot)^\wedge : \mathbb{R}^3 \to \mathfrak{so}(3)$ identifies an axis-angle vector ω with its skew-symmetric matrix $\Omega = \omega^\wedge$, whose exponential $\exp(\Omega)$ gives the corresponding rotation in $\mathrm{SO}(3)$.

Lie Routing. Lie groups provide a natural framework for modeling geometric transformations. Since the geometry tokens F_t^{3d} encode scene structure, viewpoint variation, and cross-view correspondence, they should not only serve as additional geometric context, but also indicate how semantic states should be adjusted under viewpoint changes. Conventional additive fusion injects a geometry-dependent feature, such as $B(F_t^{3d})$, into the visual representation. Although this enriches tokens with 3D information, it does not explicitly define an operation on the semantic state. Lie routing utilizes geometry features to predict local rotations of projected visual representations. The routed branch retains only the change induced by these transformations relative to the identity,

$$F_t^{2d} \mapsto F_t^{2d} + \Delta_{\mathrm{op}}\left(F_t^{2d};\, F_t^{3d}\right). \tag{3}$$

Here, F_t^{2d} denotes the semantic state and F_t^{3d} provides the geometric conditioning. The term Δ_{op} denotes only the semantic change induced by the operator, rather than the complete transformed state. By construction, this correction vanishes when the induced operator is the identity. We implement this correction using local $\mathfrak{so}(3)$ generators and a rational $\mathrm{SO}(3)$ parameterization, preserving the pretrained semantic path while geometry governs viewpoint dependent semantic transformations.

3.2 Lie-State Elicitation

For a video or multi-view observation $\{I_t\}_{t=1}^T$ of a 3D scene, VGGT [34] extracts 3D geometry features. These features are aligned to the visual token grid and projected to the MLLM interface width H, yielding geometry tokens $F^{3d} \in \mathbb{R}^{N\times H}$. Alignment details are provided in Appendix C.1. Layer normalization and two prediction heads then produce the latent rotation parameters and channel-wise routing coefficients,

$$Z = \mathrm{LN}(F^{3d}), \quad \omega = \tanh(W_\omega Z) \in \mathbb{R}^{N\times D_o}, \quad \gamma = \sigma(W_\gamma Z + b_\gamma) \in (0,1)^{N\times H}. \tag{4}$$

Here, $W_\omega : \mathbb{R}^H \to \mathbb{R}^{D_o}$ predicts latent operator states, while $W_\gamma : \mathbb{R}^H \to \mathbb{R}^H$ predicts token and channel wise routing coefficients. The learnable routing bias b_γ is initialized to -4, biasing γ toward a low activation regime, since $\sigma(-4) \approx 0.018$ when $W_\gamma Z = 0$. We set $D_o = 3\lfloor H/3 \rfloor$ and define $M = D_o/3$ triplet groups, with γ gating the routed transformation at each token and output channel.

To construct structured generators, we reshape the predicted state as $\omega \in \mathbb{R}^{N\times M\times 3}$, with $\omega_{n,m} = (\omega^x_{n,m}, \omega^y_{n,m}, \omega^z_{n,m})$, where $n \in \{1,\dots,N\}$ indexes visual tokens and $m \in \{1,\dots,M\}$ indexes

triplet subspaces. The partition is applied after projection to the operator space, rather than directly to the backbone hidden dimension. Each triplet defines a learned local latent subspace. The triplet partition is defined over learned projected channels and assumes no predefined semantic ordering of channels. Each $\omega_{n,m}$ parameterizes an element of $\mathfrak{so}(3)$ acting on the corresponding semantic triplet, with camera-dependent geometry provided by the aligned geometry tokens, and its exponential defines an $\mathrm{SO}(3)$ structured transformation in the latent subspace.

The latent operator state $\omega_{n,m}$ is mapped to a triplet-wise $\mathfrak{so}(3)$ generator through the hat map:

$$\Omega_{n,m} = \begin{bmatrix} 0 & -\omega^z_{n,m} & \omega^y_{n,m} \\ \omega^z_{n,m} & 0 & -\omega^x_{n,m} \\ -\omega^y_{n,m} & \omega^x_{n,m} & 0 \end{bmatrix} \in \mathfrak{so}(3). \tag{5}$$

The geometry encoder produces triplet-wise $\mathfrak{so}(3)$ generators over the visual tokens. The output is defined as the Lie state $S_{\mathrm{Lie}} = (\Omega, \gamma),\ \Omega = (\Omega_{n,m}) \in \mathfrak{so}(3)^{N\times M}$, where $\Omega_{n,m}$ specifies the local operator for the m-th latent group of token n, and γ controls the routed contribution at each token.

3.3 Lie-Group Routing

Given the Lie state $\mathcal{S}_{\mathrm{Lie}}$ elicited from geometry tokens, we use it to route the 2D semantic representation through geometry conditioned operators with $SO(3)$ structure. We first project the visual tokens into the operator space $V = W_v(\mathrm{LN}(F^{2d})) \in \mathbb{R}^{N\times D_o}$, where $W_v : \mathbb{R}^H \to \mathbb{R}^{D_o}$. We then partition V into $M = D_o/3$ semantic triplets $\{v_{n,m} \in \mathbb{R}^3\}_{n,m}$, and pair each $v_{n,m}$ with its corresponding $\mathfrak{so}(3)$ generator $\Omega_{n,m}$. The exponential map defines a standard $\mathrm{SO}(3)$ action for each generator and admits the Rodrigues closed form [29]. We adopt a diagonal Padé realization [18] that preserves the local rotation structure and yields a closed-form, rationally normalized correction. Its structural properties and local sensitivity are analyzed in Appendix B. Its action is expressed through the first- and second-order generator terms:

$$a_{n,m} = \Omega_{n,m} v_{n,m} = \omega_{n,m} \times v_{n,m}, \tag{6}$$

$$b_{n,m} = {\Omega_{n,m}}^2 v_{n,m} = \omega_{n,m} \times (\omega_{n,m} \times v_{n,m}) = \omega_{n,m}\big(\omega^\top_{n,m} v_{n,m}\big) - \|\omega_{n,m}\|_2^2 v_{n,m}. \tag{7}$$

Here, $a_{n,m}$ and $b_{n,m}$ are the first- and second- order generator actions on the semantic triplet. Let $R_{1,1}(\Omega_{n,m})$ denote the Padé 1/1 approximation to $\exp(\Omega_{n,m})$, which is equivalent to the Cayley action reviewed in Appendix A. LieVLM retains only the residual action relative to the identity,

$$c^{\mathrm{pade}}_{n,m} = \left(R_{1,1}(\Omega_{n,m}) - I\right) v_{n,m} = \frac{a_{n,m} + \frac{1}{2} b_{n,m}}{1 + \frac{1}{4}\|\omega_{n,m}\|_2^2}. \tag{8}$$

The corresponding transformed triplet is $v_{n,m} + c^{\mathrm{pade}}_{n,m}$, while only the correction $c^{\mathrm{pade}}_{n,m}$ enters the routed branch. We reshape all triplet-wise corrections into $\delta^{\mathrm{pade}} \in \mathbb{R}^{N\times D_o}$. A bias-free output projection maps the routed representation back to the MLLM token space,

$$\Delta_{\mathrm{op}}(F^{2d}; F^{3d}) = W_o(\delta^{\mathrm{pade}}) \in \mathbb{R}^{N\times H}, \tag{9}$$

where $W_o : \mathbb{R}^{D_o} \to \mathbb{R}^H$. The visual tokens F^{2d} provides the content to be routed, while F^{3d} conditions the $\mathfrak{so}(3)$ generators and routing strengths. This produces a geometry-conditioned operator correction that is later fused with the original semantic path and geometry anchor.

3.4 Geometry-Aware Routed Fusion

The Lie Group Routing module produces the operator correction $\Delta_{\mathrm{op}}(F^{2d}; F^{3d})$, which captures geometry-conditioned semantic transformation under the induced $\mathfrak{so}(3)$ generators. However, geometry tokens also contain 3D contextual information beyond rotational variation, such as scene layout, structural priors, and cross-view correspondence. We therefore fuse the original 2D semantic path with two complementary geometry paths: a direct 3D context path and a Lie-routed transformation path. The final geometry-aware visual representation is computed as

$$F^{out} = F^{2d} + F^{3d} + \gamma \odot \Delta_{\mathrm{op}}(F^{2d}; F^{3d}), \tag{10}$$

where $\odot$ denotes element-wise multiplication and $\gamma \in (0,1)^{N\times H}$ is the token- and channel-wise routing coefficient predicted from the geometry tokens in Eq. 4. The three terms in Eq. 10 play

complementary roles. The first term F^{2d} preserves the pretrained 2D visual-semantic representation. The second term F^{3d} provides direct 3D contextual information, and $\gamma \odot \Delta_{\mathrm{op}}(F^{2d}; F^{3d})$ introduces a geometry-conditioned correction to the semantic representation. The direct geometry path keeps 3D evidence available to the language model, while the Lie-routed path uses the geometry-conditioned operators from Sec. 3.3 to model transformation-aware semantic variation. The routing coefficient γ adaptively controls the Lie routed contribution for each token and channel.

This fusion design preserves the pretrained semantics and direct geometry paths throughout optimization, while the bounded routing coefficient γ modulates only the Lie-routed transformation. The routing head is learned jointly with the task objective, allowing the transformation strength to adapt across visual tokens and output channels.

Finally, F^{out} is used as the visual representation passed to the MLLM backbone for language generation. For a video or multi-view input, the same fusion rule is applied to each frame's aligned token grid, and the resulting fused visual tokens are concatenated with text embeddings following the standard MLLM interface. The model is trained end-to-end with the task objective without requiring explicit supervision on ω, Ω, or γ.

4 Experiments

4.1 3D Scene Understanding

Datasets & Evaluation Protocols. We evaluate LieVLM on 3D visual grounding, 3D dense captioning, and 3D video object detection. For grounding, we use the ScanRefer dataset [4] under the protocol of SPAR [40], with Mask3D proposal refinement applied consistently to methods SPAR, VG LLM, and LieVLM. For dense captioning, we use the Scan2Cap dataset [9] and generate object descriptions conditioned on Mask3D proposals from LEO [23], following prior work [23, 43, 44]. For 3D video object detection, we adopt the VG LLM protocol on the EmbodiedScan dataset [36]. We report Acc@0.25/0.5 for ScanRefer, CIDEr, BLEU-4, METEOR, and ROUGE under the @0.5 criterion for Scan2Cap, and precision, recall, and F1 at IoU 0.25 over 20 common classes for detection. All models are trained for one epoch with Adam, a batch size of 64, a warmup ratio of 0.03, and a peak learning rate of 1×10^{-5} followed by linear decay. The 2D visual encoder, 3D geometry encoder, and multimodal connector are frozen, while the learnable routing bias is initialized to $b_\gamma = -4$ and fixed as the default across all main experiments.

Baselines. The baselines are grouped according to how 3D geometry is incorporated. For ScanRefer and Scan2Cap, we compare with explicit 3D-input methods, including ScanRefer [4], Scan2Cap [9], 3D-LLM [19], LL3DA [6], Chat-3D v2 [21], Grounded 3D-LLM [7], LEO [23], Chat-Scene [22], LLaVA-3D [44], and Video-3D LLM [43]. We also include geometry-injected MLLMs, including SPAR [40], 3DRS [24], and VG LLM [42]. For 3D video object detection, we compare against Qwen2.5-VL [1] and VG LLM [42]. Tables 1–3 use published VG LLM results, while controlled analyses use our reproduced 8B baselines.

Results on 3D Dense Captioning. Table 1 reports the Scan2Cap results. LieVLM achieves the best CIDEr score of 82.7, surpassing Video-3D LLM by 2.7 points and LLaVA-3D by 3.5 points. Compared with geometry-injected RGB/video baselines, LieVLM improves over VG LLM-8B from 80.0 to 82.7 in CIDEr while remaining competitive across the other captioning metrics. These results support the effectiveness of Lie routing for object-level captioning.

Results on 3D Visual Grounding. Table 2 shows the ScanRefer results. LieVLM reaches 64.5 Acc@0.25 and 57.2 Acc@0.5. Under the same proposal refinement protocol, it improves over the published VG LLM-8B results by 6.9 and 6.3 points, respectively, demonstrating stronger grounding performance with geometry conditioned operator routing. Results in Fig. 3 show that the predictions closely overlap with the ground-truth boxes, demonstrating accurate language-guided 3D localization.

Results on 3D Video Object Detection. Table 3 evaluates 3D video object detection under the 4-frame and 6-frame settings. LieVLM improves F1 from 41.2 to 45.8 in the 4-frame setting and from 40.0 to 44.1 in the 6-frame setting. Matched three-seed results in Table 7 further confirm consistent improvements, with detailed statistics in Appendix D.2. The category-level effects vary across objects, with particularly strong gains on table, bed and bathtub.

Table 1: The quantitative results of 3D dense captioning on Scan2Cap [9]. Underline denotes overall best result, and bold denotes best results among RGB-based methods.

Method	C@0.5↑	B-4@0.5↑	M@0.5↑	R@0.5↑
Explicit 3D Input				
Scan2Cap [9]	39.1	23.3	22.0	44.8
LL3DA [6]	65.2	36.8	26.0	55.0
Chat-3D-v2 [21]	63.9	31.8	-	-
Grounded 3D-LLM [7]	70.2	35.0	-	-
LEO [23]	72.4	38.2	27.9	58.1
Chat-Scene [22]	77.1	36.5	-	-
LLaVA-3D [44]	79.2	41.1	30.2	63.4
Video-3D LLM [43]	80.0	40.2	28.5	61.7
RGB-Based Models				
3DRS [24]	76.4	39.6	27.3	57.1
VG LLM-4B [42]	78.6	40.9	28.6	62.4
VG LLM-8B [42]	80.0	41.5	28.9	62.6
LieVLM-8B (Ours)	**82.7**	**41.7**	**28.9**	**62.9**

Table 2: The results of 3D Visual grounding on ScanRefer. [4]. Underline denotes overall best, and bold denotes best among RGB-based methods.

Model	Acc@0.25	Acc@0.5
Explicit 3D Input		
ScanRefer [4]	37.3	24.3
Chat-3D v2 [21]	35.9	30.4
Grounded 3D-LLM [7]	47.9	44.1
ChatScene [22]	55.5	50.2
LLaVA-3D [44]	54.1	42.4
Video-3D LLM [43]	58.1	51.7
RGB-Based Models		
SPAR [40]	48.8	43.1
3DRS [24]	55.2	51.1
VG LLM-4B [42]	53.5	47.5
VG LLM-8B [42]	57.6	50.9
LieVLM-8B (Ours)	**64.5**	**57.2**

Table 3: The quantitative results on 3D video object detection on EmbodiedScan [36].

Method	chair	cabinet	table	bin	couch	bed	bathtub	toilet	20 Common Classes		
									Prec.@0.25	Recall@0.25	F1@0.25
4-Frame Setting											
Qwen2.5-VL [1]	41.2	11.6	36.5	30.2	41.1	68.2	36.6	68.7	34.6	31.0	32.5
VG LLM-8B [42]	**54.0**	17.1	46.5	39.8	47.0	74.1	42.1	82.5	43.4	39.6	41.2
LieVLM-8B (Ours)	53.5	**21.3**	**56.8**	**43.0**	**55.1**	**82.9**	**75.9**	**86.2**	**48.6**	**44.1**	**45.8**
6-Frame Setting											
Qwen2.5-VL [1]	36.1	10.6	32.7	25.0	40.7	64.6	38.4	68.6	31.8	28.0	29.6
VG LLM-8B [42]	48.7	17.9	44.8	**38.5**	46.4	75.8	40.4	83.2	43.5	38.7	40.0
LieVLM-8B (Ours)	**50.0**	**19.6**	**54.2**	35.5	**51.7**	**83.7**	**79.9**	**86.5**	**47.9**	**41.9**	**44.1**

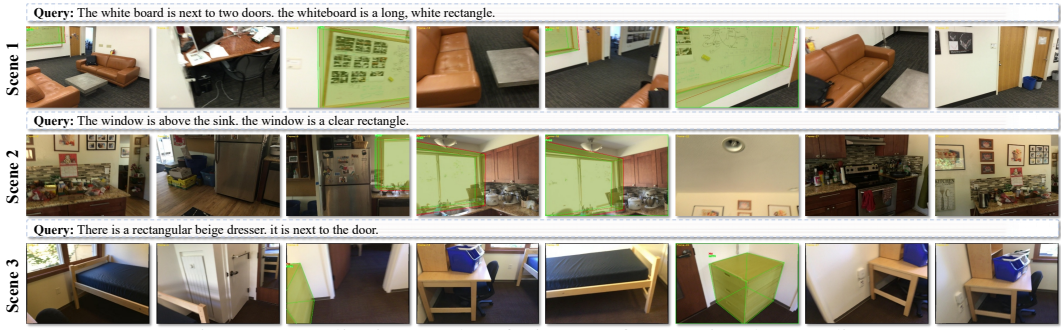


Figure 3: Qualitative results of LieVLM for 3D visual grounding.

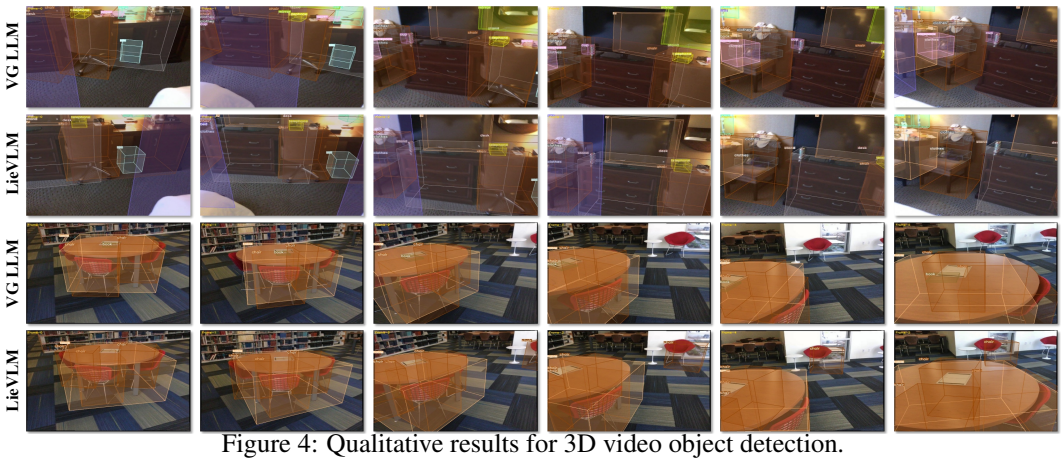


Figure 4: Qualitative results for 3D video object detection.

4.2 General Spatial Reasoning

Datasets & Evaluation Protocols. We train on a 3.05M example generalist mixture constructed from the public data sources released with CA-MLLM [39], covering video instruction following,

Table 4: Comparison on SPAR-Bench Full under the 3.05M setting. Prior model results follow published evaluations, and controlled rows use identical training and evaluation.

Method	Data Size	Avg.	Low	Medium	High	CamMotion	ViewChgI	DistI-OO-MV	ObjRel-OC-MV	SpImag-OO-MV
Prior MLLMs and Spatial Models										
InternVL2.5-8B [8]	N/A	36.28	29.46	31.88	43.80	28.00	6.32	56.85	51.75	39.50
LLaVA-OV-7B [27]	N/A	31.20	21.79	26.13	40.14	30.25	9.47	55.06	37.25	35.01
Qwen2.5-VL-7B [1]	N/A	33.07	28.75	22.97	40.27	28.75	6.83	51.49	44.75	31.93
SPAR-8B [40]	2M	63.25	65.53	63.01	61.29	73.00	37.14	79.76	64.00	53.50
VG LLM-4B [42]	297K	60.36	52.81	51.35	72.92	63.00	25.82	63.69	81.25	71.99
CA-MLLM-4B [39]	3.05M	68.35	59.94	60.42	81.74	73.00	31.01	68.75	87.50	81.23
Controlled 3.05M Setting										
VG LLM-8B [42]	3.05M	67.18	59.08	59.31	80.07	70.18	32.11	70.42	84.93	77.65
LieVLM-8B (Ours)	3.05M	**68.57** +1.39	59.81 +0.73	60.74 +1.43	81.21 +1.14	70.92 +0.74	35.26 +3.15	71.58 +1.16	85.34 +0.41	78.27 +0.62

Table 5: Comparison on CV-Bench-3D and BLINK-Spatial under the 3.05M setting.

Method	Data Size	CV-Bench-3D			BLINK-Spatial			
		Avg.	Depth	Dist.	Avg.	Depth	Spatial	Multi View
Prior MLLMs and Spatial Models								
GPT-4o	N/A	83.0	-	-	67.6	74.2	69.2	59.4
Gemini-2.5-Pro	N/A	90.8	91.0	90.7	71.6	**87.9**	**91.6**	35.3
Qwen2.5-VL-7B	N/A	77.3	84.5	76.5	73.8	79.0	86.7	55.6
Qwen2.5-VL-72B	N/A	81.0	88.7	81.0	68.6	79.8	85.3	40.6
SPAR-8B	2M	89.1	-	-	-	-	-	-
VG LLM-4B	297K	91.3	-	-	68.4	79.8	71.3	54.1
CA-MLLM-4B	3.05M	90.7	89.8	91.5	77.0	77.4	66.4	**87.2**
Controlled 3.05M Setting								
VG LLM-8B	3.05M	92.6	92.1	93.1	75.9	83.5	77.4	66.6
LieVLM-8B (Ours)	3.05M	**94.3** +1.7	**93.9** +1.7	**94.6** +1.6	**77.1** +1.2	84.0 +0.5	78.1 +0.7	69.1 +2.5

Table 6: VSI-Bench comparison across backbone and scales with multiple choice categories.

Method	Data Size	Multiple Choice				Overall Avg.
		Rel. Dist.	Rel. Dir.	Route	Appr.	
Prior MLLMs and Spatial Models						
VG LLM-4B	297K	44.6	45.6	33.5	36.4	47.3
CA-MLLM-4B	3.05M	50.7	45.5	31.4	20.2	46.8
Qwen3-VL-8B	N/A	-	-	-	-	57.9
SenseNova-SI Qwen3-VL-8B	8M	-	-	-	-	**64.8**
Controlled Lie Routing Comparisons						
VG LLM-8B	297K	46.6	40.7	32.4	59.2	50.6
LieVLM-8B (Ours)	297K	49.1 +2.5	42.9 +2.2	33.5 +1.1	59.8 +0.6	52.4 +1.7
VG LLM-Qwen3	3.05M	57.8	52.2	45.0	69.9	60.6
LieVLM-Qwen3 (Ours)	3.05M	**60.6** +2.8	**55.5** +3.3	**48.8** +3.8	**73.7** +3.8	62.8 +2.2

spatial question answering, 3D grounding and captioning, 3D detection, and object centric spatial reasoning. We evaluate Qwen2.5-VL-7B variants on SPAR-Bench Full [40], CV-Bench-3D [33], and BLINK-Spatial [12], and Qwen3-VL-8B variants on VSI-Bench [38], without benchmark-specific fine tuning. SPAR-Bench reports difficulty and camera sensitive categories, CV-Bench-3D and BLINK-Spatial report accuracy by spatial subtask, and VSI-Bench combines numerical Mean Relative Accuracy with multiple choice accuracy. The training composition is provided in Appendix C.3.

Baselines. We compare with representative generalist MLLMs and spatially specialized models, and include SenseNova-SI [2] as an external Qwen3 scale reference. Within each controlled comparison, VG LLM and LieVLM share the same language and geometry backbones, training data, optimization, and evaluation, and differ only in additive geometry fusion or Lie routing.

Generalization across Benchmarks. Tables 4 and 5 show consistent matched gains across every reported metric on SPAR-Bench Full, CV-Bench-3D, and BLINK-Spatial. LieVLM reaches the highest SPAR-Bench overall score with its largest gain on viewpoint change inference, and achieves the strongest CV-Bench-3D results across both depth and distance. The improvement further transfers to BLINK-Spatial, providing evidence beyond the original ScanNet centered evaluation suite.

Scalability across Backbones and Data. Table 6 shows that the matched benefit of Lie routing persists from the original Qwen2.5 setting to the scaled Qwen3 setting. With Qwen3-VL-8B and the 3.05M mixture, LieVLM raises the VSI-Bench average from 60.6 to 62.8 and improves seven of eight categories, including all four multiple choice categories shown in the table. The 8M SenseNova-SI results provide additional scale context, while the controlled Qwen3 comparison shows that the benefit of Lie routing persists with a stronger backbone and 10.3 times more training data. Full VSI-Bench category results and multi-seed statistics are provided in Appendix D.3.

4.3 Understanding Lie Routing

Routing Viewpoint Correspondence. A premise of Lie routing is that latent transformations should vary systematically with viewpoint while preserving target-related semantics. For same-scene frame pairs, with tokens matched by shared indices on the aligned visual grid, we relate the $\mathrm{SO}(3)$ camera rotation distance d_{rot} to the mean ℓ_2 change in within-space normalized geometry features, cross-attention residuals, and latent operator states. As shown in Figure 5, operator state variation shows the strongest monotonic association with camera rotation among the representations considered, reaching Spearman $\rho = 0.870$ compared with 0.452 for geometry features and 0.576 for cross-attention residuals. The stronger rank association indicates that the predicted transformation states vary systematically with viewpoint under this diagnostic. This characterizes the adaptive behavior of Lie routing and complements the target-localization and task-level evidence in Figure 1.

Operator Structure. To disentangle transformation structure, conditioning source, and response control, we compare variants under the same backbone, direct geometry path, training, and evalu-

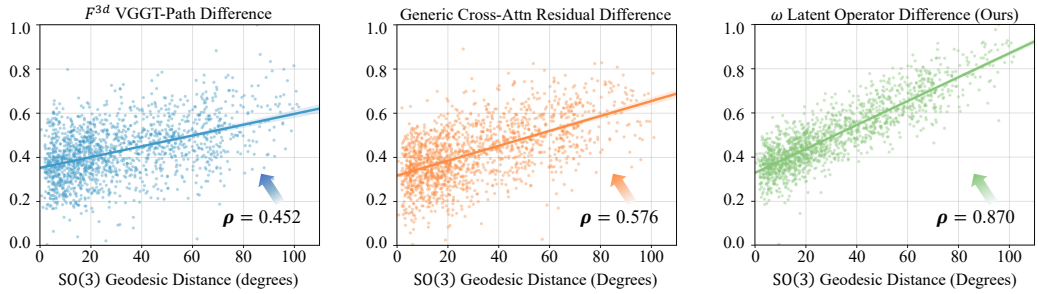


Figure 5: Routing viewpoint correspondence. Spearman correlation ρ between physical camera rotation and internal representation change across same scene frame pairs.

Table 7: Operator structure ablation under matched training. Results are mean $\pm$ std over three seeds.

Variant	Scan2Cap C@0.5	ScanRefer Acc@0.25	EmbodiedScan 4F F1@0.25	EmbodiedScan 6F F1@0.25
Additive Only	80.07±0.12	57.62±0.18	41.18±0.24	40.05±0.13
Visual-Conditioned Routing	80.71±0.18	59.56±0.23	42.48±0.19	41.69±0.22
Generic Cross-Attention	80.62±0.17	59.08±0.24	42.36±0.18	41.48±0.21
Learnable 3×3 Generator	80.11±0.14	57.91±0.20	41.56±0.15	40.71±0.18
Spectral-Controlled 3×3 Generator	80.76±0.15	59.34±0.22	42.61±0.16	41.72±0.20
Rational-Controlled 3×3 Generator	81.48±0.16	61.27±0.25	43.83±0.17	42.76±0.21
$\mathfrak{so}(3)$–Padé (Ours)	**82.69±0.15**	**64.56±0.31**	**45.76±0.13**	**44.10±0.26**

Table 8: Comparison of numerical realizations for the proposed Lie Group Routing operator.

Method	Scan2Cap		3D Video Detection		
	C@0.5	B-4@0.5	P_{25}	R_{25}	$F1_{25}$
Rodrigues	81.7	41.4	47.8	42.7	44.5
Taylor-1	81.2	41.3	47.2	42.3	43.9
Taylor-2	81.6	41.5	47.6	42.8	44.2
Taylor-3	81.9	41.4	47.9	42.9	44.6
Padé-2/2	82.4	41.6	48.4	43.3	45.3
Padé-1/1 (Ours)	**82.7**	**41.7**	**48.6**	**44.1**	**45.8**

Table 9: Sensitivity to the initial routing bias b_γ in Lie State Elicitation.

b_γ	Scan2Cap		3D Video Detection		
	C@0.5	B-4@0.5	P_{25}	R_{25}	$F1_{25}$
0	81.8	41.4	47.7	43.7	45.2
-1	82.0	41.5	47.9	43.9	45.4
-3	82.5	41.7	48.4	**44.2**	45.7
-4	**82.7**	**41.8**	**48.6**	44.1	**45.8**
-6	82.3	41.6	48.2	44.0	45.6

ation, with implementation details in Appendix C.1. *Additive Only* removes routing branch, while *Visual-Conditioned Routing* predicts the operator state and gate from F^{2d} instead of F^{3d}. *Generic cross-attention* replaces the structured transformation with cross-attention from F^{2d} to F^{3d}, and *Learnable* 3×3 *Generator* applies an unconstrained geometry-conditioned matrix $G_{n,m}$ through $c^{\text{gen}}_{n,m} = G_{n,m} v_{n,m}$, with spectral- and rational-controlled variants isolating response regulation. As shown in Table 7, cross-attention yields modest gains and the unconstrained generator remains comparable to additive fusion, while spectral and rational controls provide progressive but limited improvements. Visual-conditioned routing likewise recovers only part of the gain. The geometry-conditioned $\mathfrak{so}(3)$–Padé formulation remains consistently strongest, supporting the joint role of geometric conditioning and structured operator realization.

Operator Realization and Routing. We further examine the numerical realization of the structured operator and its routing bias initialization. The Rodrigues baseline uses the closed-form action with small-angle series for coefficient evaluation near the identity, providing an exact exponential-action reference. As shown in Tables 8 and 9, the Padé variants achieve the strongest overall results among the evaluated realizations. Padé 1/1 gives the highest CIDEr and detection F1, while $b_\gamma = -4$ yields the best CIDEr and detection precision and $b_\gamma = -3$ gives higher recall. Appendix B characterizes its rotation structure, response normalization, and local sensitivity.

5 Conclusion

We presented LieVLM, a Lie routing framework for view-robust 3D MLLMs. Rather than treating geometry as auxiliary context alone, LieVLM uses geometry tokens to predict local $\mathfrak{so}(3)$ states and routing coefficients that govern structured transformations of pretrained visual semantics across viewpoints. Through Lie-State Elicitation, Lie-Group Routing, and Geometry-Aware Routed Fusion, the model injects gated corrections while preserving the original visual-semantic, thereby mitigating viewpoint-induced semantic drift. Experiments across 3D scene understanding and spatial reasoning show consistent gains over additive geometry fusion, with benefits that generalize across benchmarks, training scales, and stronger backbones. Controlled ablations and viewpoint analyses further connect these gains to geometry-conditioned routing and structured transformation design.

Acknowledgements This work is supported by Zhejiang Leading Innovative and Entrepreneur Team Introduction Program (2024R01007). This work is also supported by The "Pioneer" and "Leading Goose" Research and Development Program of Zhejiang (2025C02077). Jingjun Yi sincerely thanks Prof. Pascal Mettes (University of Amsterdam) for his extensive help in revising the manuscript, greatly improving its clarity and presentation.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv:2502.13923*, 2025.

[2] Zhongang Cai, Ruisi Wang, Chenyang Gu, Fanyi Pu, Junxiang Xu, Yubo Wang, Wanqi Yin, Zhitao Yang, Chen Wei, Tongxi Zhou, et al. Scaling spatial intelligence with multimodal foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7879–7890, 2026.

[3] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465, 2024.

[4] 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*, pages 202–221. Springer, 2020.

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

[6] Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. LL3DA: Visual interactive instruction tuning for omni-3D understanding reasoning and planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26428–26438, 2024.

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

[8] 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.

[9] 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*, pages 3193–3203, 2021.

[10] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector Neurons: A general framework for SO(3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12200–12209, 2021.

[11] Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Shijie Zhou, Dilin Wang, et al. VLM-3R: Vision-language models augmented with instruction-aligned 3D reconstruction. *arXiv preprint arXiv:2505.20279*, 2025.

[12] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer, 2024.

[13] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. SE(3)-Transformers: 3D roto-translation equivariant attention networks. *Advances in Neural Information Processing Systems*, 33:1970–1981, 2020.

[14] Zoey Guo, Yiwen Tang, Ray Zhang, Dong Wang, Zhigang Wang, Bin Zhao, and Xuelong Li. Viewrefer: Grasp the multi-view knowledge for 3d visual grounding. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15326–15337. IEEE, 2023.

[15] Brian C Hall. Lie groups, Lie algebras, and representations. In *Quantum Theory for Mathematicians*, pages 333–366. Springer, 2013.

[16] Richard Hartley, Andrew Zisserman, et al. *Multiple view geometry in computer vision*, volume 2. Cambridge university press Cambridge, 2003.

[17] Kyle Helfrich, Devin Willmott, and Qiang Ye. Orthogonal recurrent neural networks with scaled Cayley transform. In *International Conference on Machine Learning*, pages 1969–1978. PMLR, 2018.

[18] Nicholas J Higham. *Functions of matrices: theory and computation*. SIAM, 2008.

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

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

[21] Haifeng Huang, Zehan Wang, Rongjie Huang, Luping Liu, Xize Cheng, Yang Zhao, Tao Jin, and Zhou Zhao. Chat-3D v2: Bridging 3D scene and large language models with object identifiers. *arXiv preprint arXiv:2312.08168*, 2(3):6, 2023.

[22] 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.

[23] 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. *arXiv preprint arXiv:2311.12871*, 2023.

[24] Xiaohu Huang, Jingjing Wu, Qunyi Xie, and Kai Han. 3DRS: Mllms need 3D-aware representation supervision for scene understanding. *arXiv preprint arXiv:2506.01946*, 2025.

[25] Michael J Hutchinson, Charline Le Lan, Sheheryar Zaidi, Emilien Dupont, Yee Whye Teh, and Hyunjik Kim. LieTransformer: Equivariant self-attention for Lie groups. In *International Conference on Machine Learning*, pages 4533–4543. PMLR, 2021.

[26] Du Q Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.

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

[28] Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. GTA: A geometry-aware attention mechanism for multi-view transformers. In *International Conference on Learning Representations*, 2024.

[29] Richard M Murray, Zexiang Li, and S Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC Press, 2017.

[30] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[31] Arijit Ray, Jiafei Duan, Ellis Brown, Reuben Tan, Dina Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A Plummer, Ranjay Krishna, et al. SAT: Dynamic spatial aptitude training for multimodal language models. *arXiv preprint arXiv:2412.07755*, 2024.

[32] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In *International Conference on Machine Learning*, pages 9323–9332. PMLR, 2021.

[33] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai C Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024.

[34] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5294–5306, 2025.

[35] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUST3R: Geometric 3D vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

[36] Tai Wang, Xiaohan Mao, Chenming Zhu, Runsen Xu, Ruiyuan Lyu, Peisen Li, Xiao Chen, Wenwei Zhang, Kai Chen, Tianfan Xue, et al. EmbodiedScan: A holistic multi-modal 3D perception suite towards embodied AI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19757–19767, 2024.

[37] 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.

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

[39] Gongjie Zhang, Wenhao Li, Quanhao Qian, Jiuniu Wang, Deli Zhao, Shijian Lu, and Ran Xu. On the generalization capacities of MLLMs for spatial intelligence. *arXiv preprint arXiv:2603.06704*, 2026.

[40] 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*, 38:171909–171941, 2025.

[41] Ruosen Zhao, Zhikang Zhang, Jialei Xu, Jiahao Chang, Dong Chen, Lingyun Li, Weijian Sun, and Zizhuang Wei. SpaceMind: Camera-guided modality fusion for spatial reasoning in vision-language models. *arXiv preprint arXiv:2511.23075*, 2025.

[42] Duo Zheng, Shijia Huang, Yanyang Li, and Liwei Wang. Learning from videos for 3D world: Enhancing MLLMs with 3D vision geometry priors. *arXiv preprint arXiv:2505.24625*, 2025.

[43] Duo Zheng, Shijia Huang, and Liwei Wang. Video-3D LLM: Learning position-aware video representation for 3D scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8995–9006, 2025.

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

Supplementary Material

Contents

A Background: Exponential Map on $\mathrm{SO}(3)$ **14**

A.1 Rodrigues Formula 14

A.2 Taylor Expansion 14

A.3 Padé Approximation 14

B Theoretical Analysis **15**

B.1 Direct Rodrigues Evaluation Near the Identity 15

B.2 Local Jacobian Bounds for Padé Routing 15

B.3 Structure and Response of Taylor and Padé Actions 17

C Experimental Setting and Details **18**

C.1 Implementation Details 18

C.2 Data Preparation 19

C.3 Large Scale Spatial Reasoning Setting 19

C.4 Evaluation Details 20

D Additional Experimental Results **21**

D.1 Controlled Viewpoint Responses 21

D.2 3D Video Object Detection on EmbodiedScan 22

D.3 Detailed VSI-Bench Results 24

D.4 Robustness to Sparse Views and Geometry Perturbations 24

E More Visualization **25**

F Broader Impact **26**

G Limitations **26**

H Usage of Large Language Models **26**

A Background: Exponential Map on $\mathrm{SO}(3)$

This section provides background on the numerical realization of the exponential action on $\mathrm{SO}(3)$. We first recall the exact Rodrigues formula, and then review Taylor and Padé approximations, which are commonly used to implement matrix exponentials in practice.

A.1 Rodrigues Formula

Let $\Omega = \omega^\wedge \in \mathfrak{so}(3)$ and $\theta = \|\omega\|_2$. The exponential map from $\mathfrak{so}(3)$ to $\mathrm{SO}(3)$ admits the classical Rodrigues formula,

$$\exp(\Omega) = I + A(\theta)\Omega + B(\theta)\Omega^2, \tag{11}$$

where, for $\theta > 0$, the scalar coefficients are

$$A(\theta) = \frac{\sin\theta}{\theta}, \qquad B(\theta) = \frac{1-\cos\theta}{\theta^2}. \tag{12}$$

For a vector $v \in \mathbb{R}^3$, the corresponding action is

$$\exp(\Omega)v = v + A(\theta)\Omega v + B(\theta)\Omega^2 v. \tag{13}$$

The Rodrigues formula is exact in real arithmetic, but direct evaluation of its coefficient ratios requires care near $\theta = 0$. Both ratios have removable singularities with limits

$$\lim_{\theta\to 0} A(\theta) = 1, \qquad \lim_{\theta\to 0} B(\theta) = \frac{1}{2}. \tag{14}$$

We set $A(0) = 1$ and $B(0) = 1/2$ by continuous extension. Taylor expansions provide regular local approximations for evaluating the coefficients near zero. The apparent singularities therefore belong to the ratio expressions, while the exponential map itself remains smooth.

A.2 Taylor Expansion

Taylor expansion [18] provides the most direct power-series realization of the matrix exponential. It follows from the analytic definition of the exponential map, which represents $\exp(\Omega)$ as an infinite sum of weighted powers of the generator Ω. Truncating this series yields a polynomial approximation whose accuracy improves near the identity as more orders are retained. For $\Omega \in \mathfrak{so}(3)$ and $v \in \mathbb{R}^3$, the exponential action can be written as

$$\exp(\Omega)v = \sum_{k=0}^{\infty} \frac{1}{k!}\Omega^k v = v + \Omega v + \frac{1}{2}\Omega^2 v + \frac{1}{6}\Omega^3 v + \cdots. \tag{15}$$

A finite Taylor approximation of order K is therefore

$$\Phi_{\text{Taylor-}K}(\Omega, v) = \sum_{k=0}^{K} \frac{1}{k!}\Omega^k v. \tag{16}$$

This polynomial form is smooth around the identity and avoids the explicit trigonometric ratios in Rodrigues formula, making it suitable for small-angle implementations. However, finite Taylor truncations control the response only through the truncation order and $\|\Omega\|$, without the rational normalization introduced by Padé approximations.

A.3 Padé Approximation

Padé approximation [18] provides a rational-function realization of a given analytic function. Given a scalar function $f(x)$, its (m, n) Padé approximant takes the form

$$R_{m,n}(x) = \frac{P_m(x)}{Q_n(x)} = \frac{a_0 + a_1 x + \cdots + a_m x^m}{1 + b_1 x + \cdots + b_n x^n}, \tag{17}$$

where the coefficients are chosen such that the Taylor expansion of $R_{m,n}(x)$ matches that of $f(x)$ up to the highest possible order around the expansion point. Compared with Taylor truncation, Padé

approximation uses both numerator and denominator polynomials, and therefore provides a rational rather than purely polynomial approximation.

For the matrix exponential, Padé approximation replaces the infinite exponential series with a rational matrix function. The commonly used diagonal Padé forms approximate $\exp(\Omega)$ by a ratio of two matrix polynomials. In particular, the Padé-$1/1$ form is

$$R_{1,1}(\Omega) = \left(I - \frac{1}{2}\Omega\right)^{-1} \left(I + \frac{1}{2}\Omega\right), \tag{18}$$

where $(\cdot)^{-1}$ denotes the matrix inverse, usually implemented by solving a linear system. This form is exactly the Cayley map applied to $\Omega/2$. A higher-order diagonal variant is the Padé-$2/2$ form:

$$R_{2,2}(\Omega) = \left(I - \frac{1}{2}\Omega + \frac{1}{12}\Omega^2\right)^{-1} \left(I + \frac{1}{2}\Omega + \frac{1}{12}\Omega^2\right). \tag{19}$$

Both Padé-$1/1$ and Padé-$2/2$ are rational approximations to $\exp(\Omega)$. When $\Omega \in \mathfrak{so}(3)$ is skew-symmetric, these diagonal Padé forms preserve the rotation structure because the denominator polynomial is the transpose of the numerator polynomial. Thus, they define valid $\mathrm{SO}(3)$ actions while approximating the exact exponential map.

B Theoretical Analysis

This section analyzes the local sensitivity and structural properties of the operator realizations. We examine direct Rodrigues evaluation near the identity, establish Jacobian bounds for Padé actions, and compare the rotation structure and response magnitude of finite Taylor and Padé forms.

B.1 Direct Rodrigues Evaluation Near the Identity

Rodrigues formula expresses the exact exponential action through the coefficients $A(\theta)$ and $B(\theta)$. Their direct derivative expressions involve subtracting nearly equal quantities near $\theta = 0$, which can amplify rounding errors during evaluation. We examine this computational form while retaining the continuous limits of the underlying functions.

Specifically, differentiating $A(\theta) = \frac{\sin\theta}{\theta}, B(\theta) = \frac{1-\cos\theta}{\theta^2}$ gives

$$dA = \frac{\theta\cos\theta - \sin\theta}{\theta^3}(\omega^\top d\omega), \quad dB = \frac{\theta\sin\theta - 2(1-\cos\theta)}{\theta^4}(\omega^\top d\omega). \tag{20}$$

The scalar coefficients in Eq. 20 have well-defined limits:

$$\lim_{\theta\to 0}\frac{\theta\cos\theta - \sin\theta}{\theta^3} = -\frac{1}{3}, \quad \lim_{\theta\to 0}\frac{\theta\sin\theta - 2(1-\cos\theta)}{\theta^4} = -\frac{1}{12}. \tag{21}$$

B.2 Local Jacobian Bounds for Padé Routing

We establish a local Jacobian bound for the full Padé 1/1 action underlying the operator correction. The derivative is taken with respect to the predicted operator state while holding the visual feature fixed. The argument extends to general Padé forms under a uniform nonsingularity condition on the denominator.

Proposition B.1 (Local Jacobian bound)**.** *Let $\Omega = \omega^\wedge \in \mathfrak{so}(3)$ and define the Padé-$1/1$ routing action as*

$$\Phi_{1,1}(\omega, v) = v + \frac{\Omega v + \frac{1}{2}\Omega^2 v}{1 + \frac{1}{4}\|\omega\|_2^2}. \tag{22}$$

For any $\rho > 0$, there exists a constant $C_\rho < \infty$ such that, for all $\|\omega\|_2 \le \rho$ and $v \in \mathbb{R}^3$,

$$\left\| \frac{\partial \Phi_{1,1}(\omega, v)}{\partial \omega} \right\| \le C_\rho \|v\|_2. \tag{23}$$

The Jacobian norm is the operator norm induced by the Euclidean norm.

Proof. Let

$$q(\omega) = 1 + \frac{1}{4}\|\omega\|_2^2, \quad N(\omega, v) = \Omega v + \frac{1}{2}\Omega^2 v. \tag{24}$$

Then $\Phi_{1,1}(\omega, v) = v + N(\omega, v)/q(\omega)$. For any perturbation direction $h \in \mathbb{R}^3$, since $\Omega = \omega^\wedge$ depends linearly on ω, we have

$$D\Omega[h] = h^\wedge. \tag{25}$$

Therefore, the differentials of the numerator and denominator are

$$\begin{aligned} DN(\omega, v)[h] &= h^\wedge v + \frac{1}{2}\left(h^\wedge \Omega v + \Omega h^\wedge v\right), \\ Dq(\omega)[h] &= \frac{1}{2}\omega^\top h. \end{aligned} \tag{26}$$

Since $\|x^\wedge\|_2 = \|x\|_2$ for any $x \in \mathbb{R}^3$, when $\|\omega\|_2 \le \rho$, we have

$$\begin{aligned} \|N(\omega, v)\|_2 &\le \left(\rho + \frac{1}{2}\rho^2\right)\|v\|_2, \\ \|DN(\omega, v)[h]\|_2 &\le (1+\rho)\|h\|_2\|v\|_2, \\ q(\omega) \ge 1, \quad |Dq(\omega)[h]| &\le \frac{1}{2}\rho\|h\|_2. \end{aligned} \tag{27}$$

Using the above relations, the directional derivative of $\Phi_{1,1}$ can be bounded as follows:

$$\begin{aligned} \|D\Phi_{1,1}(\omega, v)[h]\|_2 &\overset{(1)}{=} \left\| \frac{DN(\omega, v)[h]q(\omega) - N(\omega, v)Dq(\omega)[h]}{q(\omega)^2} \right\|_2 \\ &\overset{(2)}{\le} \frac{\|DN(\omega, v)[h]\|_2}{q(\omega)} + \frac{\|N(\omega, v)\|_2 |Dq(\omega)[h]|}{q(\omega)^2} \\ &\overset{(3)}{\le} (1+\rho)\|h\|_2\|v\|_2 + \frac{1}{2}\rho\left(\rho + \frac{1}{2}\rho^2\right)\|h\|_2\|v\|_2 \\ &= \left((1+\rho) + \frac{1}{2}\rho\left(\rho + \frac{1}{2}\rho^2\right)\right)\|h\|_2\|v\|_2 \\ &= C_\rho \|h\|_2\|v\|_2, \end{aligned} \tag{28}$$

where $C_\rho = (1+\rho) + \frac{1}{2}\rho(\rho + \frac{1}{2}\rho^2) < \infty$.

The derivation of Eq. 28 follows.

(1) follows from the quotient rule applied to $\Phi_{1,1}(\omega, v) = v + N(\omega, v)/q(\omega)$.

(2) follows from the triangle inequality and norm sub-multiplicativity.

(3) follows from the bounds in Eq. 27 and the fact that $q(\omega) \ge 1$.

Taking the supremum over all perturbation directions with $\|h\|_2 = 1$ gives

$$\left\| \frac{\partial \Phi_{1,1}(\omega, v)}{\partial \omega} \right\| \le C_\rho \|v\|_2. \tag{29}$$

This completes the proof. □

Remark B.2 (Extension to Padé-m/n). The above proof uses Padé-$1/1$ as the default case in our main model, but the same argument applies to a general Padé-m/n routing action under a local nonsingularity condition. Let

$$P_m(\Omega) = \sum_{i=0}^{m} p_i \Omega^i, \quad Q_n(\Omega) = \sum_{j=0}^{n} q_j \Omega^j, \quad q_0 = 1, \tag{30}$$

and define

$$\Phi_{m,n}(\omega, v) = Q_n(\Omega)^{-1} P_m(\Omega) v. \tag{31}$$

Assume that, for all $\|\omega\|_2 \le \rho$, the denominator matrix is locally nonsingular:

$$\sigma_{\min}(Q_n(\Omega)) \ge \alpha_{m,n,\rho} > 0. \tag{32}$$

We further define the coefficient-dependent quantities

$$P_{m,\rho} = \sum_{i=0}^{m} |p_i|\rho^i, \quad P'_{m,\rho} = \sum_{i=1}^{m} i|p_i|\rho^{i-1}, \quad Q'_{n,\rho} = \sum_{j=1}^{n} j|q_j|\rho^{j-1}. \tag{33}$$

For any perturbation direction $h \in \mathbb{R}^3$, since $D\Omega[h] = h^\wedge$ and $\|h^\wedge\|_2 = \|h\|_2$, we have

$$\|D\Omega^k[h]\|_2 = \left\| \sum_{\ell=0}^{k-1} \Omega^\ell h^\wedge \Omega^{k-1-\ell} \right\|_2 \leq k\rho^{k-1}\|h\|_2. \tag{34}$$

Therefore,

$$\begin{aligned} \|P_m(\Omega)v\|_2 &\leq P_{m,\rho}\|v\|_2, \\ \|DP_m(\Omega)[h]v\|_2 &\leq P'_{m,\rho}\|h\|_2\|v\|_2, \\ \|DQ_n(\Omega)[h]\|_2 &\leq Q'_{n,\rho}\|h\|_2, \\ \|Q_n(\Omega)^{-1}\|_2 &\leq \alpha_{m,n,\rho}^{-1}. \end{aligned} \tag{35}$$

Using these bounds, the directional derivative of $\Phi_{m,n}$ can be bounded as

$$\begin{aligned} \|D\Phi_{m,n}(\omega, v)[h]\|_2 &\overset{(1)}{=} \left\| -Q_n^{-1}(DQ_n[h])Q_n^{-1}P_m v + Q_n^{-1}(DP_m[h])v \right\|_2 \\ &\overset{(2)}{\leq} \|Q_n^{-1}\|_2\|DP_m[h]v\|_2 + \|Q_n^{-1}\|_2^2\|DQ_n[h]\|_2\|P_m v\|_2 \\ &\overset{(3)}{\leq} \left(\frac{P'_{m,\rho}}{\alpha_{m,n,\rho}} + \frac{Q'_{n,\rho}P_{m,\rho}}{\alpha_{m,n,\rho}^2} \right) \|h\|_2\|v\|_2 \\ &= C_{m,n,\rho}\|h\|_2\|v\|_2, \end{aligned} \tag{36}$$

where

$$C_{m,n,\rho} = \frac{P'_{m,\rho}}{\alpha_{m,n,\rho}} + \frac{Q'_{n,\rho}P_{m,\rho}}{\alpha_{m,n,\rho}^2}. \tag{37}$$

The derivation of Eq. 36 follows.

(1) follows from the product rule and the inverse differential identity $D(Q^{-1}) = -Q^{-1}(DQ)Q^{-1}$.

(2) follows from the triangle inequality and norm sub-multiplicativity.

(3) follows from the bounds in Eq. 35.

Taking the supremum over all perturbation directions with $\|h\|_2 = 1$ gives

$$\left\| \frac{\partial \Phi_{m,n}(\omega, v)}{\partial \omega} \right\| \leq C_{m,n,\rho}\|v\|_2. \tag{38}$$

Thus, the Padé action has a locally bounded Jacobian under the stated denominator condition. For Padé 1/1, the correction used in the main model has the strictly positive scalar denominator $1 + \|\omega\|_2^2/4$.

Remark B.3. The bound characterizes the sensitivity of the exact Padé action to the predicted operator state. The implemented triplet correction is $c_{n,m}^{\text{pade}} = \Phi_{1,1}(\omega_{n,m}, v_{n,m}) - v_{n,m}$. For fixed v, subtracting v leaves the Jacobian with respect to ω unchanged, so the same bound applies to the implemented correction. This result concerns input sensitivity in exact arithmetic, independently of finite precision rounding error.

B.3 Structure and Response of Taylor and Padé Actions

Finite Taylor and diagonal Padé forms provide smooth realizations of the operator action near the identity. We compare their rotation structure and the magnitude of the induced semantic change.

First, finite Taylor truncations do not generally preserve the $\mathrm{SO}(3)$ constraint. Let $T_K(\Omega) = \sum_{k=0}^{K} \frac{1}{k!}\Omega^k$. Since $\Omega^\top = -\Omega$, its orthogonality defect is

$$T_K(\Omega)^\top T_K(\Omega) - I = T_K(-\Omega)T_K(\Omega) - I. \tag{39}$$

This defect is composed of higher-order powers of Ω. For example,

$$\begin{aligned} T_1(\Omega)^\top T_1(\Omega) - I &= -\Omega^2, \\ T_2(\Omega)^\top T_2(\Omega) - I &= \frac{1}{4}\Omega^4, \\ T_3(\Omega)^\top T_3(\Omega) - I &= -\frac{1}{12}\Omega^4 - \frac{1}{36}\Omega^6. \end{aligned} \tag{40}$$

These expressions quantify the orthogonality defect of finite Taylor truncations. The defect depends on both the truncation order and generator magnitude, and finite truncation does not guarantee a rotation matrix for arbitrary generators. As $\|\Omega\|_2 = \|\omega\|_2$ increases, these residual high-order terms become more visible, so Taylor truncations do not strictly define rotation matrices unless additional projection or re-orthogonalization is applied.

In contrast, diagonal Padé forms preserve the orthogonal structure for skew-symmetric generators. A diagonal Padé approximation can be written as

$$R_{\ell,\ell}(\Omega) = Q_\ell(\Omega)^{-1} P_\ell(\Omega), \quad Q_\ell(\Omega) = P_\ell(-\Omega). \tag{41}$$

Since $\Omega^\top = -\Omega$, we have $Q_\ell(\Omega) = P_\ell(\Omega)^\top$. Therefore,

$$R_{\ell,\ell}(\Omega)^\top R_{\ell,\ell}(\Omega) = Q_\ell(\Omega) P_\ell(\Omega)^{-1} Q_\ell(\Omega)^{-1} P_\ell(\Omega) = I, \tag{42}$$

where the last equality follows because $P_\ell(\Omega)$ and $Q_\ell(\Omega)$ are polynomials of the same matrix and hence commute. If $Q_\ell(t\Omega)$ remains nonsingular for all $t \in [0, 1]$, continuity from $R_{\ell,\ell}(0) = I$ further gives $\det R_{\ell,\ell}(\Omega) = 1$. In particular, the Padé 1/1 and Padé 2/2 forms used here have nonsingular denominators for real skew-symmetric generators and therefore yield matrices in $\mathrm{SO}(3)$.

Second, Taylor truncations are purely polynomial. For the finite Taylor approximation in Eq. 16, the operator induced change satisfies

$$\|\Phi_{\text{Taylor-}K}(\Omega, v) - v\|_2 \le \left(\sum_{k=1}^{K} \frac{\|\omega\|_2^k}{k!} \right) \|v\|_2. \tag{43}$$

By contrast, the Padé-$1/1$ routing form has the bound

$$\|\Phi_{1,1}(\omega, v) - v\|_2 \le \frac{\|\omega\|_2 + \frac{1}{2}\|\omega\|_2^2}{1 + \frac{1}{4}\|\omega\|_2^2} \|v\|_2. \tag{44}$$

The denominator in Eq. 44 provides a rational normalization of the linear and quadratic generator actions. Finite Taylor forms avoid direct evaluation of the Rodrigues coefficient ratios, while Padé 1/1 additionally preserves the rotation structure and introduces rational normalization of the operator response.

C Experimental Setting and Details

C.1 Implementation Details

LieVLM uses VGGT-1B as the geometry encoder and Qwen2.5-VL-7B as the default language backbone. For the scaled spatial reasoning experiments in Sec. 4.2, we additionally instantiate matched VG LLM and LieVLM variants with Qwen3-VL-8B-Instruct. Within each matched setting, the two models use identical initialization, training data, optimization, and evaluation, and differ only in additive geometry fusion or Lie routing. For the original setting, we train for one epoch using Adam with a batch size of 64. The learning rate is warmed up with a ratio of 0.03 to a peak value of 1×10^{-5}, and then linearly decayed to zero. At each iteration, one data source is sampled from the mixed training corpus to form the batch. During training, the 2D visual encoder, the 3D geometry encoder, and the multimodal connector are kept frozen. We optimize the language backbone together with the proposed LieVLM modules, including the rotation head, routing head, and routed fusion layers. The output projection is applied to the correction $c_{n,m}^{\text{pade}} = R_{1,1}(\Omega_{n,m}) v_{n,m} - v_{n,m}$, rather than to the full transformed triplet. The routing head $W_\gamma : \mathbb{R}^H \to \mathbb{R}^H$ predicts a token and channel wise gate $\gamma \in (0, 1)^{N \times H}$, which modulates the routed feature after the output projection W_o. The

learnable routing bias b_γ is initialized to -4 and optimized jointly with the routing head. For the spectral-controlled 3×3 generator in Table 7, we retain the prediction head and parameter count of the unconstrained dynamic-matrix baseline but normalize each predicted residual matrix as

$$\bar{G}_{n,m} = \frac{\tau G_{n,m}}{\max(\tau, \|G_{n,m}\|_2)}, \qquad c^{\mathrm{spec}}_{n,m} = \bar{G}_{n,m} v_{n,m}. \tag{45}$$

We set $\tau = 2\sqrt{3/7} \approx 1.309$, matching the maximum spectral norm of $R_{1,1}(\Omega) - I$ under the component-wise bounded state $\omega = \tanh(W_\omega Z)$. The baseline otherwise shares the direct geometry path, routing gate, output projection, initialization, and training protocol with the unconstrained generator. For the rational-controlled 3×3 generator, we retain the same unconstrained prediction head but replace the hard spectral cap with

$$c^{\mathrm{rat}}_{n,m} = \frac{G_{n,m} v_{n,m}}{1 + \frac{1}{8}\|G_{n,m}\|_F^2}. \tag{46}$$

The factor $1/8$ matches the Padé-1/1 denominator for a skew-symmetric generator because $\|\omega^\wedge\|_F^2 = 2\|\omega\|_2^2$. This variant retains the same direct geometry path, routing gate, output projection, parameter count, and training protocol. For visual-conditioned routing, we retain the complete $\mathfrak{so}(3)$–Padé routing architecture and direct geometry path, but replace the conditioning input in Eq. 4 with $Z^{2d} = \mathrm{LN}(F^{2d})$. Both ω and γ are therefore predicted from visual semantics rather than geometry tokens, with all other components and training settings unchanged. The MLLM interface width is $H = 3584$ for Qwen2.5-VL-7B and $H = 4096$ for Qwen3-VL-8B. We set $D_o = 3\lfloor H/3 \rfloor$, giving $(D_o, M) = (3582, 1194)$ and $(4095, 1365)$ for the two backbones, respectively. The projections W_v and W_ω map H to D_o, W_γ maps H to H, and the bias free W_o maps D_o back to H. The four projection matrices contain $3HD_o + H^2$ weights, corresponding to 51.36M and 67.10M parameters for the two settings, excluding biases. Following the geometry fusion interface of VG LLM [42], we adopt the same VGGT-to-MLLM token alignment and projection procedure without modification. The aligned VGGT features are projected to the interface width H before fusion. The resulting geometry tokens F^{3d} share the token indexing and channel dimension of F^{2d}. The same aligned geometry representation supplies both the direct geometry path and Lie State Elicitation, so geometry injection and operator prediction operate on a common token grid. All experiments are conducted on 8 H100 80G GPUs. For 3D scene understanding, training takes approximately 12 hours for LieVLM. For the original 297K VG LLM spatial reasoning setting, training takes approximately 9 hours.

C.2 Data Preparation

We use publicly available datasets and follow the VG LLM [42] preprocessing protocol to convert them into the input-output format required by LieVLM.

3D Scene Understanding. For ScanRefer and Scan2Cap, we sample 32 RGB frames per scene. For 3D video object detection, each sample contains a 4-frame clip at 1 FPS, with annotations in the first-frame coordinate system.

SPAR-7M. We process SPAR-7M with the official implementation[2], retaining spatial QA samples with valid visual inputs and rendering visual markers using the released pipeline. We exclude navigation samples because of their variable-length image sequences.

LLaVA-Video-178K. We use the LLaVA-Hound split as general video instruction data, sampling frames at 2 FPS and keeping 4–8 frames per video.

297K Spatial Reasoning Setting. Following VG LLM [42], we use 234K SPAR-7M samples and 63K LLaVA-Hound samples, corresponding to 3% and 25% of the two source datasets. The same composition and preprocessing are used for VG LLM and LieVLM.

C.3 Large Scale Spatial Reasoning Setting

Training Mixture. Following the public data sources and annotations released with CA-MLLM, we construct a 3.05M example generalist mixture covering video instruction following, spatial question answering, 3D grounding and captioning, 3D detection, and object centric spatial reasoning. We

[2] `https://github.com/fudan-zvg/spar`

retain our own architecture and training pipeline and do not use the camera ray embeddings, explicit camera intrinsics, camera specific augmentation, or UniDepth distillation introduced by CA-MLLM. Table 10 summarizes the complete composition.

Table 10: Composition of the 3.05M generalist training mixture.

Data Component	Public Sources	Sampling	Examples
General video instruction	LLaVA Hound split of LLaVA Video 178K	Full released split	0.26M
Large scale spatial QA	SPAR subsets from ScanNet, ScanNet++, Structured3D, and RxR	10% of each training subset	1.63M
3D grounding and captioning	ScanRefer and Scan2Cap	Official training splits	0.07M
Other spatially grounded tasks	ScanNet, 3RScan, Matterport3D, ARKitScenes, Objectron, and SUN RGB-D	Released annotations	1.09M
Total	–	–	**3.05M**

Matched Training and Evaluation. The Qwen2.5 variants start from the same Qwen2.5-VL-7B and VGGT-1B checkpoints, while the Qwen3 variants start from the same Qwen3-VL-8B-Instruct and VGGT-1B checkpoints. Within each pair, VG LLM and LieVLM use identical training examples, training settings, and evaluation implementations. The only architectural difference is additive geometry fusion or Lie routing. The evaluated checkpoints receive no benchmark specific fine tuning.

Data Exposure. The complete pretraining and post training data of Qwen2.5-VL are not publicly auditable, so prior exposure to ScanNet or ScanNet++ cannot be excluded. Such exposure may affect the absolute scores of both variants. Within each matched comparison, VG LLM and LieVLM use the same base checkpoint and instruction tuning data, so this uncertainty does not create a differential data exposure advantage for LieVLM. CV-Bench-3D and BLINK-Spatial further provide evaluation outside the original ScanNet centered benchmark suite.

C.4 Evaluation Details

Following VG LLM [42], we evaluate LieVLM with LMMs-Eval[3] and use customized task interfaces when necessary. During inference, all responses are generated with greedy decoding.

3D Dense Captioning. For Scan2Cap, we follow the common two-stage evaluation protocol [44, 43]. Object proposals are first obtained from the Mask3D detections provided by LEO [23]. We transform each proposal center into the reference-frame coordinate system using the corresponding extrinsic matrices [36], and then prompt the model to describe the object conditioned on this coordinate. Generated captions are evaluated with CIDEr, BLEU-4, ROUGE, and METEOR using the COCO caption evaluation toolkit[4].

3D Visual Grounding. For ScanRefer, the model predicts both the frame where the referred object is visible and the corresponding oriented 3D bounding box in that frame's camera coordinates. The predicted box is transformed into the world coordinate system before evaluation. Since the original ground-truth boxes are axis-aligned, we append zero rotation angles to obtain oriented boxes for IoU computation. We then report Acc@0.25 and Acc@0.5. For the viewpoint trend in Figure 1(b), we retain the corresponding raw IoU before thresholding to obtain a continuous per-sample measure. For proposal refinement, we compare the predicted box with Mask3D proposals from LEO [23] and select the proposal with the highest IoU.

3D Video Object Detection. For each predicted box, we perform category-wise greedy matching against unused ground-truth boxes. A prediction is counted as a true positive if its best unmatched ground-truth box of the same category exceeds the IoU threshold of 0.25. Matched ground-truth boxes are removed from later matching. Precision, recall, and F1 are then computed for each category and averaged over the 20 common classes.

Spatial Reasoning. For VSI-Bench [38], we use the default LMMs-Eval configuration. Each video is uniformly sampled into 32 frames. Numerical-answer tasks are evaluated with Mean Relative Accuracy, while multiple-choice tasks are evaluated with standard accuracy.

[3] `https://github.com/EvolvingLMMs-Lab/lmms-eval`

[4] `https://github.com/tylin/coco-caption`

D Additional Experimental Results

D.1 Controlled Viewpoint Responses

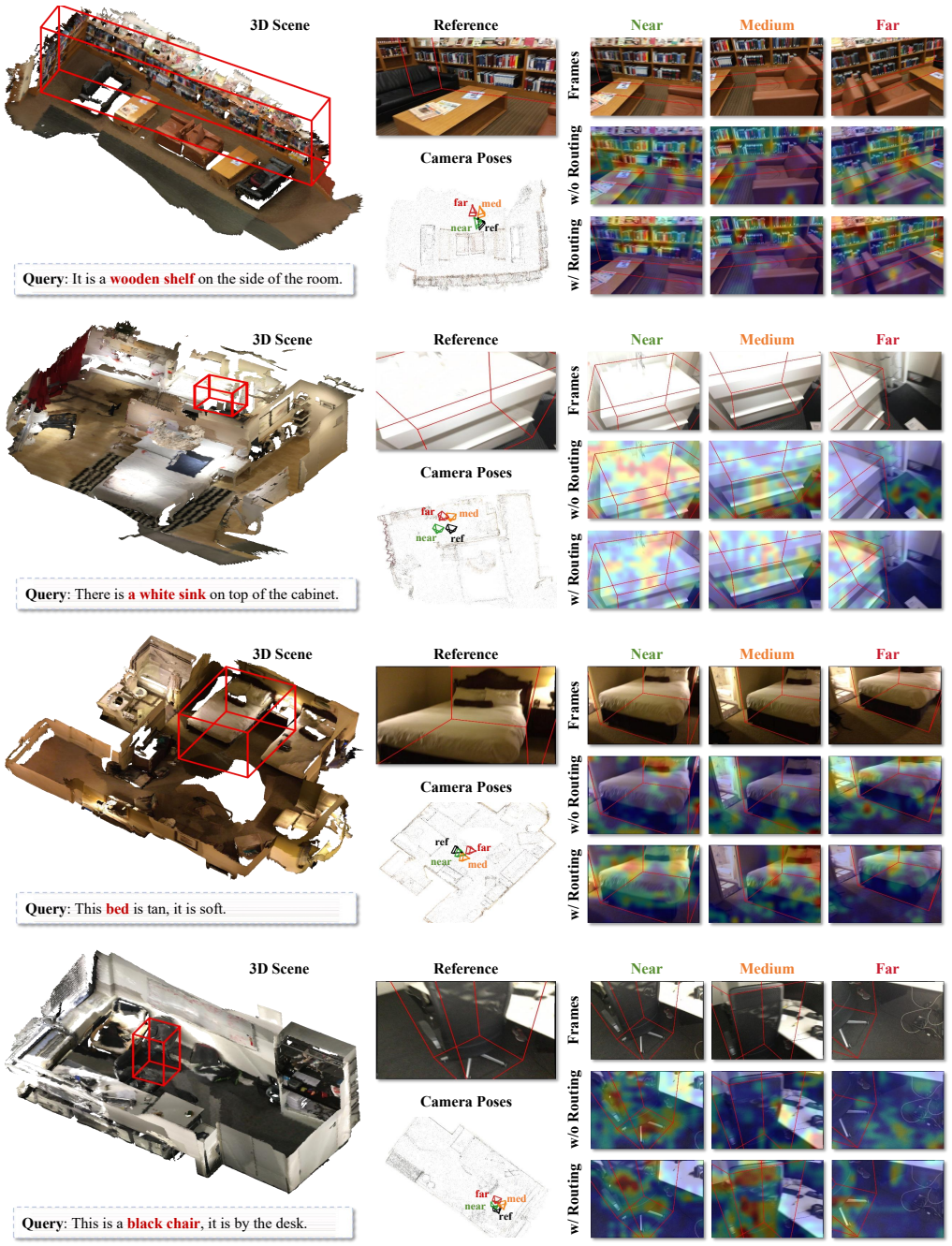


Figure 6: Additional controlled viewpoint response examples extending Fig. 1(a). query-relevant activations are shown across reference, near, medium, and far views without and with Lie routing. Warmer regions indicate stronger support for the target phrase. Red wireframes denote projected ground truth boxes used only for visualization and PBO@0.2.

Figure 6 extends the controlled analysis in Fig. 1a with four additional scene target query examples. Across the wooden shelf, white sink, tan bed, and black chair cases, responses without routing vary substantially across viewpoints and often become diffuse or shift away from the target. Lie

Table 11: Target overlap of query-relevant activations under controlled viewpoint variation. PBO@0.2 reports the percentage of top response locations inside the projected target box.

Scene	Target	Routing	Near	Medium	Far
scene0011_00	Long table	Without Routing	44.6	2.7	49.5
		Lie Routing	**81.6**	**49.4**	**62.8**
scene0064_00	Wooden shelf	Without Routing	82.6	64.7	34.0
		Lie Routing	**86.2**	**74.0**	**86.7**
scene0246_00	White sink	Without Routing	73.5	40.6	15.6
		Lie Routing	**95.1**	**71.5**	**80.7**
scene0645_00	Tan bed	Without Routing	48.8	11.5	30.6
		Lie Routing	**83.2**	**28.2**	**70.6**
scene0700_00	Black chair	Without Routing	48.3	31.9	8.9
		Lie Routing	**76.8**	**77.8**	**53.8**
Mean		Without Routing	59.6	30.3	27.7
		Lie Routing	**84.6**	**60.2**	**70.9**

routing consistently preserves more localized target responses across the near, medium, and far views, supporting the same cross view behavior observed in the main teaser.

Activation Localization. Following the controlled setting in Fig. 1(a), each example fixes the scene, target, query, checkpoint, and reference view, chosen as the first frame containing the target, and varies only the viewpoint along the same camera trajectory. Near, medium, and far views are selected by increasing reference relative rotation distance $d_{\mathrm{rot}}(r, v)$. Let $H_v(p)$ denote the final decoder feature at visual token p, and let $\mathcal{Q}_{\mathrm{t}}$ contain the target phrase tokens. The query-relevant activation is

$$A_v(p) = |\mathcal{Q}_{\mathrm{t}}|^{-1} \sum_{c \in \mathcal{Q}_{\mathrm{t}}} \left[W_c^{\top} H_v(p) \right]_+ , \tag{47}$$

where W_c is the vocabulary projection for token c, and $[x]_+ = \max(x, 0)$. Let C_v denote the top 20 percent activation locations and let B_v denote the projected target region. We measure target localization using

$$\mathrm{PBO@0.2} = 100 \frac{|C_v \cap B_v|}{|C_v|}. \tag{48}$$

PBO quantifies target localization within each view rather than monotonic degradation with viewpoint distance. The projected target boxes are used only for visualization and evaluation, and are not provided to the model. Lie routing improves PBO@0.2 in all 15 evaluated views and increases the overall mean from 39.2 to 71.9. The improvements remain consistent across the near, medium, and far viewpoint groups.

D.2 3D Video Object Detection on EmbodiedScan

Category-Level Results. Tables 12 and 13 provide category-level F1 scores for 3D video object detection. LieVLM improves over Qwen2.5-VL on nearly all categories, confirming the benefit of geometry-aware modeling for cross-frame 3D localization. Compared with VG LLM, LieVLM also shows clear gains on many spatially prominent categories, including table, couch, bed, bathtub, toilet, refrigerator, and computer in the 4-frame setting, as well as chair, table, desk, bed, bathtub, toilet, stove, and computer in the 6-frame setting. These improvements suggest that Lie-Group Routing helps preserve object orientation and spatial consistency when detections are aggregated into the first-frame coordinate system.

We also observe several remaining challenges. For categories with limited visual evidence or ambiguous geometry, such as ottoman, microwave, and monitor, LieVLM does not always improve over VG LLM. In particular, ottoman remains difficult in both frame settings, likely due to sparse instances and weak appearance cues. These results indicate that operatorized geometry substantially improves common object localization, while rare or small objects still require stronger supervision or more reliable visual evidence.

Table 12: The detailed results on 3D video object detection in 4-frame setting. We report the F1 score for all categories.

Object	Qwen2.5-VL	VG LLM	LieVLM (Ours)
chair	41.2	54.0	53.5
pillow	13.2	22.9	25.7
cabinet	11.6	17.1	21.3
table	36.5	46.5	56.8
lamp	9.0	18.9	23.3
couch	41.1	47.0	55.1
desk	34.5	41.0	45.4
stand	33.7	46.1	50.2
bed	68.2	74.1	82.9
backpack	33.3	41.4	49.5
bathtub	36.6	42.1	75.9
ottoman	0.0	0.0	0.0
dresser	38.2	49.2	47.5
bin	30.2	39.8	43.0
toilet	68.7	82.5	86.2
refrigerator	38.5	53.6	60.0
stove	61.2	68.6	68.5
microwave	21.3	33.3	24.1
monitor	21.3	34.0	29.7
computer	12.0	11.4	16.5

Table 13: The detailed results on 3D video object detection in 6-frame setting. We report the F1 score for all categories.

Object	Qwen2.5-VL	VG LLM	LieVLM (Ours)
chair	36.1	48.7	50.0
pillow	12.4	18.9	22.8
cabinet	10.6	17.9	19.6
table	32.7	44.8	54.2
lamp	8.0	15.9	18.7
couch	40.7	46.4	51.7
desk	34.4	32.3	45.7
stand	31.3	47.3	43.1
bed	64.6	75.8	83.7
backpack	28.6	40.4	46.3
bathtub	38.4	40.4	79.9
ottoman	2.7	5.3	0.1
dresser	36.4	44.5	46.6
bin	25.0	38.5	35.5
toilet	68.6	83.2	86.5
refrigerator	33.2	62.0	59.6
stove	55.9	69.5	77.5
microwave	7.8	25.8	19.6
monitor	20.3	31.7	25.6
computer	5.2	11.6	15.3

Multi-Seed Reliability. We further evaluate the stability of the matched VG LLM and LieVLM comparison over three random seeds. Table 14 provides detailed three-seed detection statistics consistent with the F1 results in Table 7. LieVLM improves all six aggregate precision, recall, and F1 metrics in every matched run, with 4-frame and 6-frame F1 gains of 4.58 ± 0.11 and 4.05 ± 0.13, respectively.

Table 14: Multi-seed reliability on EmbodiedScan. Results are mean ± std over three matched random seeds.

Setting	Metric	VG LLM	LieVLM	Paired Gain	Positive Seeds
4-Frame	Prec.@0.25	43.19 ± 0.25	48.26 ± 0.14	$+5.07 \pm 0.11$	3/3
	Recall@0.25	39.44 ± 0.23	43.78 ± 0.12	$+4.34 \pm 0.11$	3/3
	F1@0.25	41.18 ± 0.24	45.76 ± 0.13	$+4.58 \pm 0.11$	3/3
6-Frame	Prec.@0.25	43.35 ± 0.14	47.58 ± 0.28	$+4.23 \pm 0.14$	3/3
	Recall@0.25	38.57 ± 0.12	41.60 ± 0.24	$+3.03 \pm 0.12$	3/3
	F1@0.25	40.05 ± 0.13	44.10 ± 0.26	$+4.05 \pm 0.13$	3/3

Table 15: Detailed VSI-Bench comparison across model and scaling settings. Green values denote changes over the corresponding controlled VG LLM baseline.

Method	**Avg.**	**Numerical Answer**				**Multiple Choice Answer**			
		Obj. Count	**Abs. Dist.**	**Obj. Size**	**Room Size**	**Rel. Dist.**	**Rel. Dir.**	**Route Plan**	**Appr. Order**
Prior MLLMs and Spatial Models									
GPT-4o	34.0	46.2	5.3	43.8	38.2	37.0	41.3	31.5	28.5
Gemini-1.5-Pro	45.4	56.2	30.9	64.1	43.6	51.3	46.3	36.0	34.6
Gemini-2.5-Pro	52.7	48.2	35.6	**71.3**	51.7	58.9	42.4	46.8	66.7
InternVL2-8B	34.6	23.1	28.7	48.2	39.8	36.7	30.7	29.9	39.6
VILA-1.5-40B	31.2	22.4	24.8	48.7	22.7	40.5	25.7	31.5	32.9
LongVA-7B	29.2	38.0	16.6	38.9	22.2	33.1	43.3	25.4	15.7
LLaVA-NeXT-Video-72B	40.9	48.9	22.8	57.4	35.3	42.4	36.7	35.0	48.6
SAT-LLaVA-Video-7B	-	-	-	-	47.3	41.1	37.1	36.1	40.4
RynnEC-7B	45.8	58.5	25.4	54.9	42.7	44.2	-	38.7	30.5
CA-MLLM-4B	46.8	71.3	39.3	50.4	65.9	50.7	45.5	31.4	20.2
Qwen3-VL-8B	57.9	-	-	-	-	-	-	-	-
SenseNova-SI	**64.8**	-	-	-	-	-	-	-	-
Controlled Qwen2.5 Setting with 297K Training									
VG LLM-8B	50.6	67.9	37.7	58.6	62.0	46.6	40.7	32.4	59.2
LieVLM-8B (Ours)	52.4 +1.7	69.6 +1.7	40.5 +2.8	58.7 +0.1	64.7 +2.7	49.1 +2.5	42.9 +2.2	33.5 +1.1	59.8 +0.6
Controlled Qwen3 Setting with 3.05M Training									
VG LLM-Qwen3	60.6	74.4	50.9	64.3	70.3	57.8	52.2	45.0	69.9
LieVLM-Qwen3 (Ours)	62.8 +2.2	**75.4** +1.0	**52.7** +1.8	64.2 -0.1	**71.5** +1.2	**60.6** +2.8	**55.5** +3.3	**48.8** +3.8	**73.7** +3.8

D.3 Detailed VSI-Bench Results

Full Category Comparison. Table 15 complements the compact scaling comparison in Table 6 with all eight VSI-Bench categories and a broader set of prior MLLMs and spatial models. The Qwen3 matched comparison improves seven of eight categories, while the original Qwen2.5 setting provides complementary three seed evidence. Category results are unavailable for the two external Qwen3 scale references, for which only the reported benchmark averages are shown.

Multi Seed Reliability. The original Qwen2.5 setting is additionally evaluated over three matched random seeds. LieVLM improves all eight categories in every run, increasing the average from 50.62 ± 0.18 to 52.35 ± 0.11. Table 16 reports the complete uncertainty, paired gains, and positive seed counts.

D.4 Robustness to Sparse Views and Geometry Perturbations

Sparse Input Views. We further evaluate whether the advantage of Lie routing persists when fewer views are available. Frames are uniformly subsampled from the same temporal interval while preserving the scene, query, temporal span, and endpoint views, and VG LLM and LieVLM receive identical frame subsets at every density. As shown in Table 17, LieVLM remains stronger across all evaluated input densities. The performance gap is preserved as the observations become sparser, with larger gains under the lowest density on Scan2Cap and EmbodiedScan.

Geometry Perturbations. We separately measure the sensitivity of LieVLM to errors in the predicted geometry by perturbing F^{3d} while keeping the visual tokens F^{2d} fixed. Additive feature noise with $\sigma = 0.10$ and token dropout with $p = 30\%$ are applied to the trained checkpoints

Table 16: Multi seed reliability on VSI-Bench. Results are mean ± standard deviation over three matched seeds, with paired gains computed seed wise.

Metric	VG LLM-8B	LieVLM-8B	Paired Gain	Positive Seeds
Obj. Count	67.88±0.26	**69.58±0.14**	**+1.70±0.12**	3/3
Abs. Dist.	37.68±0.24	**40.50±0.11**	**+2.82±0.13**	3/3
Obj. Size	58.60±0.06	**58.70±0.09**	**+0.10±0.03**	3/3
Room Size	61.98±0.21	**64.69±0.13**	**+2.71±0.09**	3/3
Rel. Dist.	46.57±0.22	**49.11±0.10**	**+2.54±0.12**	3/3
Rel. Dir.	40.69±0.22	**42.90±0.15**	**+2.20±0.08**	3/3
Route Plan	32.39±0.20	**33.50±0.08**	**+1.11±0.12**	3/3
Appr. Order	59.20±0.07	**59.80±0.12**	**+0.60±0.05**	3/3
Average	50.62±0.18	**52.35±0.11**	**+1.72±0.07**	3/3

Table 17: Performance under decreasing input view density. Relative density is measured with respect to the task specific full input setting. Both models receive identical frame subsets sampled from the same temporal interval. The full-density results are from the runs used for the sparse-view study, whereas Table 7 reports three-seed averages.

Task	Frames	Relative Density	VG LLM	LieVLM
Scan2Cap C@0.5	32	100%	80.06	82.70
	16	50%	74.31	77.08
	8	25%	66.42	70.71
ScanRefer Acc@0.5	32	100%	50.87	57.21
	16	50%	48.34	54.85
	8	25%	42.66	49.36
EmbodiedScan 6F F1@0.25	6	100%	40.78	44.41
	4	66.7%	38.05	42.41
	3	50%	34.47	39.32

without corruption specific retraining, and each stochastic setting is averaged over three independent realizations. As shown in Table 18, LieVLM retains at least 95.5% of clean performance under both perturbations, indicating tolerance to moderate errors in the geometry representation.

Table 18: Inference time robustness to geometry perturbations. Values in brackets report performance retained relative to the clean setting. The clean results are from the runs used for the perturbation study, whereas the multi-seed detection results are reported in Table 14.

Condition	Severity	ScanRefer Acc@0.5	EmbodiedScan 4F F1@0.25	EmbodiedScan 6F F1@0.25
Clean	–	57.20 [100%]	45.90 [100%]	44.40 [100%]
Geometry Feature Noise	$\sigma = 0.10$	55.87 [97.7%]	44.93 [97.9%]	43.31 [97.5%]
Geometry Token Dropout	$p = 30\%$	54.76 [95.7%]	43.84 [95.5%]	42.69 [96.1%]

E More Visualization

We provide additional qualitative results for 3D visual grounding in Fig. 7. The examples cover diverse object categories and spatial expressions, including cabinets, armchairs, pictures, trash cans, chairs, doors, carts, windows, and tables. Across these cases, LieVLM is able to associate the language query with the correct temporal evidence and localize the referred object in 3D space. The projected boxes generally align well with the target regions across different views, showing that the model can handle object attributes, relative positions, and view-dependent appearances. We also observe that challenging cases mainly arise from depth ambiguity, partial visibility, and thin or distant objects, where the 2D projection can still look plausible while the 3D box orientation or depth is less accurate.

Fig. 8 presents additional results for 3D video object detection. Compared with VG LLM, LieVLM produces more stable 3D boxes across frames and better preserves the spatial orientation of objects.

In several examples, VG LLM detects the major objects but predicts boxes with incorrect poses or misses objects under viewpoint changes. In contrast, LieVLM recovers more complete object sets and yields boxes that are better aligned with the visible geometry. These qualitative comparisons further support the role of Lie-Group Routing in reducing view-induced semantic drift and improving cross-frame 3D localization.

F Broader Impact

LieVLM aims to improve the view robustness of 3D multimodal reasoning from RGB videos. This capability can benefit applications that require reliable spatial understanding across changing viewpoints, such as embodied navigation, assistive robotics, indoor mapping, and human-robot interaction. By reducing the dependence on explicit 3D sensors or reconstructed scene inputs, the framework may also make 3D scene understanding more accessible in resource-constrained settings. At the same time, LieVLM inherits several risks from its underlying MLLM backbone and geometry encoder. The model may still hallucinate objects or spatial relations, especially under poor visibility, severe occlusion, motion blur, or inaccurate geometry estimation. Since its predictions can influence downstream spatial decisions, the model should not be used as the sole source of truth in safety-critical systems without verification, calibration, and human oversight. The base language and vision models may also carry dataset biases, and improved 3D reasoning could be misused for intrusive surveillance or unauthorized spatial analysis. We therefore encourage responsible deployment, careful evaluation across environments and populations, and transparency about failure cases when releasing 3D MLLM systems.

G Limitations

LieVLM uses the same geometry representation for direct 3D context and latent operator prediction, so reconstruction or alignment errors can affect both paths. The routing gate and Padé normalization constrain the influence of moderate perturbations, but severe geometry degradation can still reduce the reliability of routed corrections. The current formulation uses local $\mathrm{SO}(3)$ actions within learned three dimensional channel groups. The rotation constraint applies to these local action matrices. The learned projections and routing gates do not impose equivariance or composition consistency with respect to physical camera transformations on the complete model. These actions provide rotation oriented structure but do not explicitly represent translation, non rigid motion, or long range interactions, and the latent groups should not be interpreted as physical coordinates. The routing heads and output projections add computation relative to additive fusion. Current evaluations focus mainly on static indoor scenes, leaving dynamic objects, outdoor environments, and long horizon trajectories for future study.

H Usage of Large Language Models

In this work, Large Language Models (LLMs) are only used for polishing the writing. They are not involved in research ideation, experimental design, data analysis, or the formulation of conclusions. All substantive intellectual contributions are made by the authors. The authors have checked the content throughout the paper after polishing, and take full responsibility for the current form.

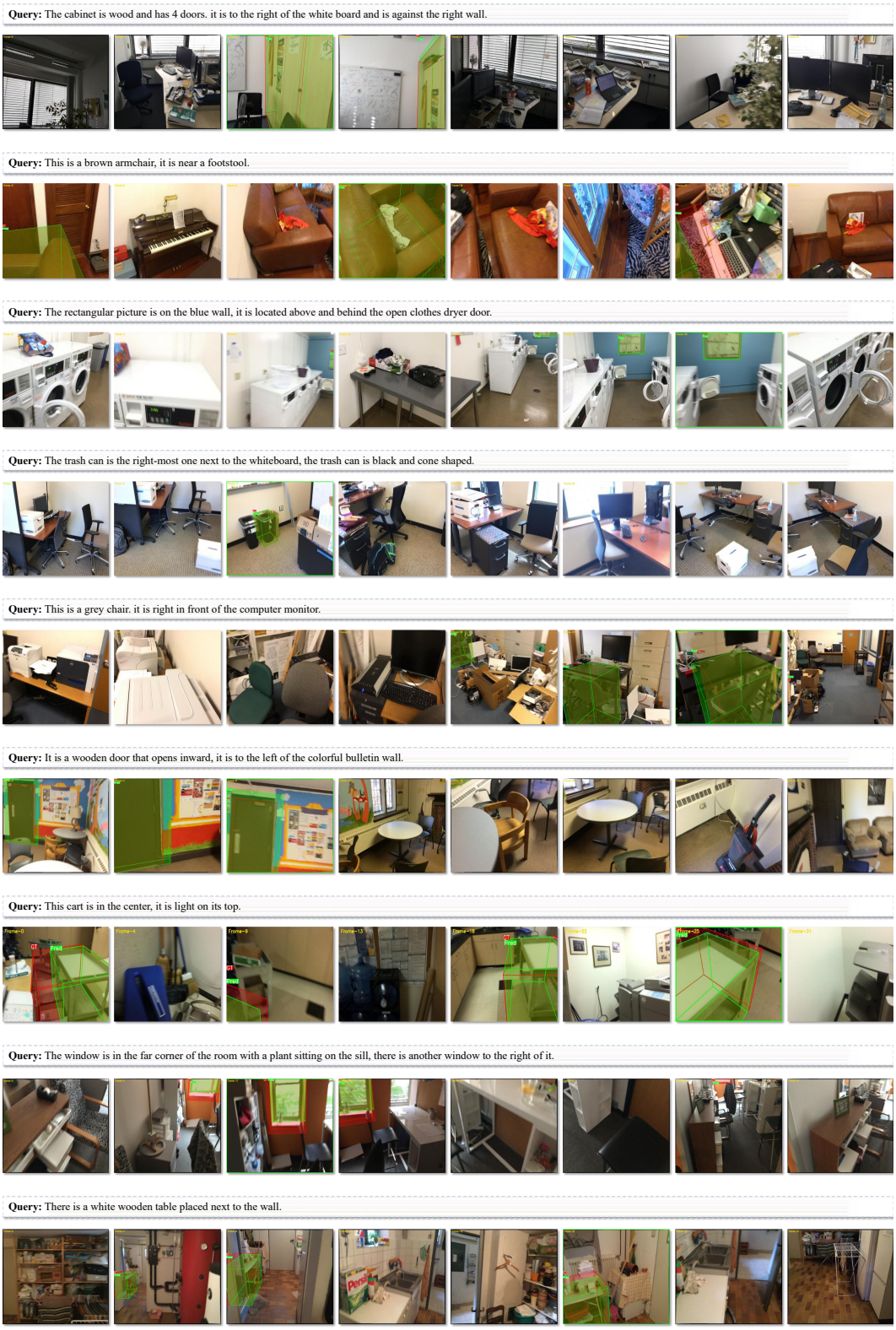


Figure 7: Additional visualization results for 3D visual grounding.

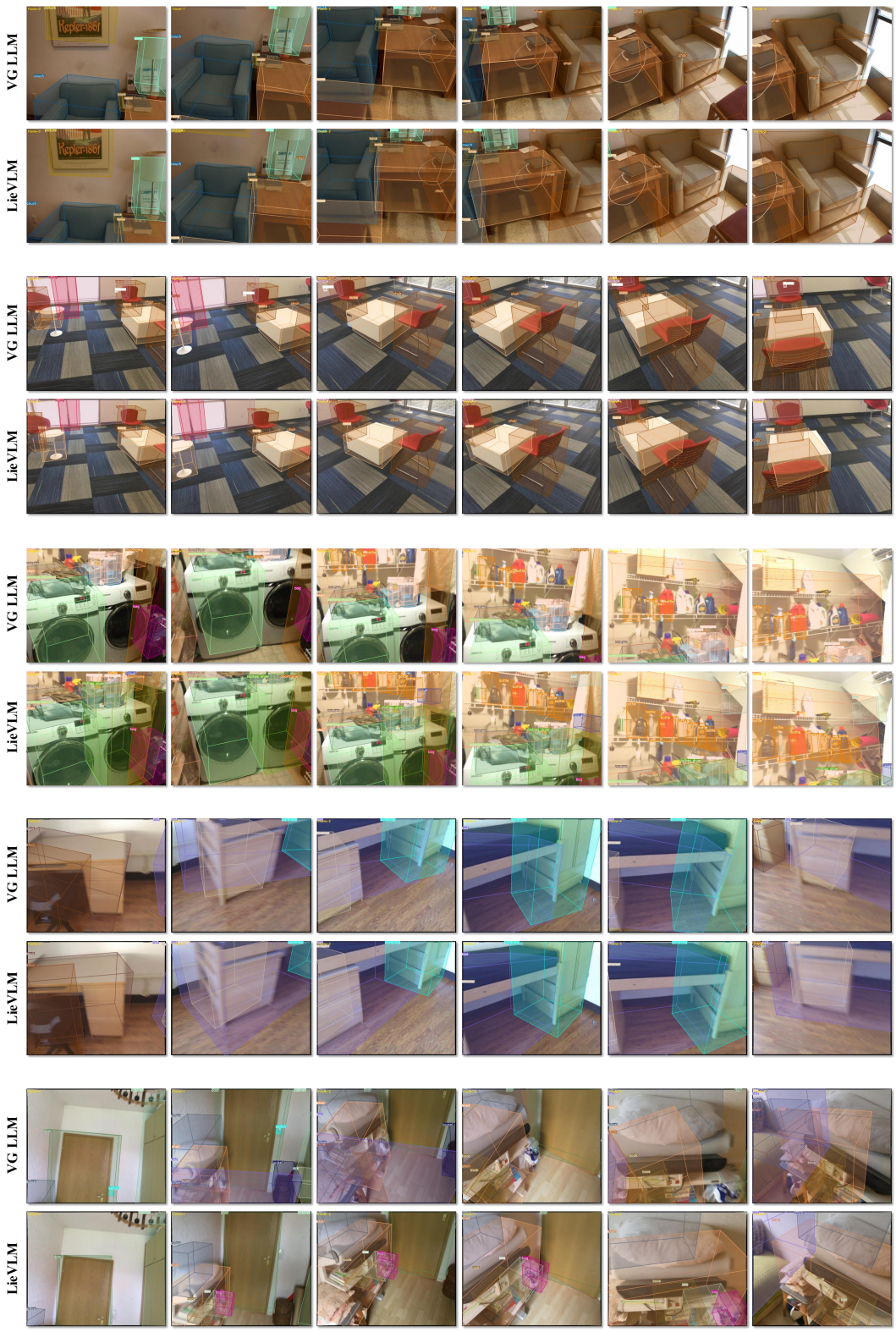


Figure 8: Additional visualization results for 3D video object detection.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction covers the theoretical and technical contribution, the developed method and the experimental contribution.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: At the end of the conclusion section, a limitation discussion is provided.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: Yes, the assumptions are given in the form of Lema in the Prelimaries section. A complete proof is given in the Supplementary material.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Yes, the baseline model, technical details, hyper-parameter and configuration are detailed in the submission for reproducibility.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: The datasets this paper uses are publicly available, and the source code is promised to be public once published.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The implementation details are given at the beginning of the experimental section.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Following prior works in this field, the evaluation protocols on the corresponding datasets do NOT require a report of the error bar.

   Guidelines:

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

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The hardware, especially the GPU requirement, is listed in the supplementary materials.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This paper focuses on a fundamental task of machine learning and conducts experiments on publicly available datasets.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The societal impact of this work has been discussed at the end of the conclusion section. We do not envision negative societal impact could be brought by this work.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This work focuses on a fundamental problem in machine learning and conducts experiments on standard datasets. We do not envision such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Yes, all the assets have been properly cited, with a license to use for academia and no commercial purpose.

    Guidelines:

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

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

13. **New assets**

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

    Answer: [N/A]

    Justification: This paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development in this research of this paper does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

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