

MVGGT: Multimodal Visual Geometry Grounded Transformer for Multiview 3D Referring Expression Segmentation

Supplementary Material

1. Implementation Details

Network Architecture. Our framework adopts a dual-branch design. The geometric branch utilizes the pre-trained Pi3-Large backbone [6], while the linguistic branch employs a pre-trained language encoder. To preserve pre-trained knowledge and minimize computational costs, the majority of the network parameters (1343M) are kept frozen. The core trainable component, MVGGT, comprises approximately 385M learnable parameters.

Training Protocols. The entire framework is optimized end-to-end using the AdamW [3] optimizer with a weight decay of 0.05. We set the initial learning rate to 1×10^{-4} for the trainable MVGGT module. The training spans 30 epochs (with 1,000 iterations per epoch) and a batch size of 16. The Foreground Gradient Dilution (FGD) issue is addressed via the PVSO loss with λ_p fixed to 1.0.

Input Settings. We adopt distinct input resolutions to balance efficiency and precision. During training, input RGB images are resized to 224×224 pixels. During inference (testing), we increase the resolution to 518×336 pixels to capture finer geometric details from the sparse views.

Hardware Environment. All experiments are conducted on a workstation equipped with a single NVIDIA RTX 4090 GPU (24GB VRAM).

Analysis of Late Fusion Strategy. Regarding the fusion architecture, we identify Late Fusion as the optimal stage for injecting linguistic cues, which we implement by attaching the multimodal branch exclusively to the final 12 layers of the backbone. We attribute the superiority of this design to the hierarchical nature of feature evolution within Transformers. Shallow layers in the visual encoder primarily capture low-level geometric primitives, such as edges and textures, which possess a significant semantic gap relative to abstract linguistic concepts; consequently, injecting language at this early stage introduces semantic interference, disrupting the bottom-up construction of scene structure. In contrast, deep layers encode high-level semantic abstractions that are naturally aligned with textual representations. Therefore, by restricting fusion to the later stages, MVGGT ensures that cross-modal interaction occurs within a shared semantic space, allowing language to effectively modulate object-level visual features without compromising the underlying geometric scaffold.

2. Additional Baseline Comparisons

To evaluate the competitiveness of MVGGT against the state-of-the-art in 3D referring expression segmentation, we extend our comparison to include two powerful methods: LESS [2] and MDIN [7].

Implementation of Baselines. Since LESS [2] and MDIN [7] depend on dense, high-quality point clouds which are unavailable in our setting, we adapt them into the Two-stage pipeline.

1. *Reconstruction Stage:* We use the Pi3 [6] backbone to reconstruct a sparse point cloud from the input multi-view RGB images. Note that these point clouds inevitably contain noise, outliers, and missing structures due to the sparsity of input views.
2. *Segmentation Stage:* The reconstructed point clouds are then fed into LESS [2] and MDIN [7] (retrained on MVRefer) to predict the target mask.

Table I. Comparison with additional baselines on MVRefer.

Method	Easy		Hard		Overall	
	Global	View	Global	View	Global	View
2D-Lift	25.4	24.1	6.4	15.0	17.8	20.4
Two-stage (LESS) [2]	25.8	28.2	8.1	8.6	18.5	20.3
Two-stage (MDIN) [7]	4.1	41.6	2.7	49.7	3.6	44.6
MVGGT (Ours)	50.1	70.6	24.4	67.3	39.9	69.3

Results and Analysis. Table I contrasts MVGGT with representative baselines adapted to the two-stage setting. When relying on independent 2D predictions (2D-Lift) or sparse geometric reconstructions (LESS [2] and MDIN [7]), performance remains limited, reflecting the difficulty of grounding language without high-fidelity 3D signals. Specifically, the strongest baseline (LESS [2]) achieves only 18.5 overall mIoU_{global}, while MDIN [7] drops significantly to 3.6, indicating that standard 3DRES models struggle to extract valid features from noisy, sparse-view point clouds. We attribute the poor global performance of MDIN to its reliance on superpoint extraction. While effective for dense scans, this geometric pre-processing proves fragile under sparse reconstruction noise, leading to feature representation collapse. Consequently, the model frequently predicts empty masks. Although this behavior maintains a baseline mIoU_{view} score by correctly classifying target-absent views, the model fails to segment the object when it is actually present, resulting in minimal global accuracy. In contrast, MVGGT yields a clear improvement,

raising overall $mIoU_{global}$ to 39.9 and $mIoU_{view}$ to 69.3. This stability is most pronounced on hard scenes, where performance jumps from 8.1 (LESS [2]) to 24.4, showing that integrating language into geometric reasoning efficiently guides sparse-view aggregation even when structural evidence is minimal. This substantial gap reinforces our premise that effective MV-3DRES relies on an architecture specifically designed to be resilient to sparse, uneven supervision rather than one dependent on perfect geometry.

3. Full Pipeline Efficiency Analysis

Real-world deployment on embodied agents (e.g., robots, AR devices) requires considering the entire latency budget, from sensor data capture to the final decision. To this end, we conduct a holistic efficiency comparison between MVGGT, the Traditional 3D RES pipeline, and Two-stage baselines, covering the full data lifecycle.

Evaluation Setup. We perform evaluations on a single NVIDIA RTX 4090 GPU with a batch size of 1. The reported latency is the average time over the ScanRefer validation set. We break down the timeline into three critical stages:

- **Stage 1: Data Acquisition (Acq.).** For Traditional 3D RES, this involves time-consuming dense scanning (estimated as ~ 300 s); for Baselines and MVGGT, it is standardized to ~ 2.0 s for sparse capture.
- **Stage 2: 3D Reconstruction (Recon.).** This covers explicit geometric computation (e.g., offline reconstruction or Pi3 [6] inference).
- **Stage 3: Segmentation (Seg.).** The inference time of the segmentation model.

The **Total Latency** is defined as the cumulative time of these three stages.

Table II. Full pipeline efficiency comparison. We evaluate latency across Data Acquisition (Acq.), 3D Reconstruction (Recon.), and Segmentation (Seg.). We explicitly sum Stage 2 and 3 as **Processing (Proc.)** time to highlight the algorithmic efficiency.

Method	Acq. (Stage 1)	Recon. (Stage 2)	Seg. (Stage 3)	Proc. (Stage 2+3)	Total	Remark
Traditional 3D RES	~ 300 s	$1\sim 2$ h	~ 0.3 s	> 1 h	> 1 h	High Latency
2D-Lift	~ 2.0 s	0.3 s	0.5 s	0.8 s	~ 2.8 s	No Generalization
Two-stage (LESS) [2]	~ 2.0 s	0.3 s	0.3 s	0.6 s	~ 2.6 s	
Two-stage (MDIN) [7]	~ 2.0 s	0.3 s	0.4 s	0.7 s	~ 2.7 s	
MVGGT (Ours)	~ 2.0 s	0.3s(End to End)		0.3 s	~ 2.3 s	Robust Perception

Analysis of Full Pipeline Trade-offs. Table II reveals critical bottlenecks in existing approaches. The Traditional 3D RES pipeline involves prohibitive costs for scanning (~ 300 s) and offline reconstruction ($1\sim 2$ h), resulting in high latency unsuitable for online interaction. While Two-stage Baselines (e.g., 2D-Lift, LESS [2]) successfully reduce acquisition time via sparse capture, they still incur a cumulative Processing (Proc.) overhead ranging from 0.6 s to 0.8 s due to the sequential execution of reconstruction and seg-

Table III. Memory Usage Comparison

Method	MVGGT (Ours)	Two-stage	2D-Lift
Memory (GB)	8.39	11.14	12.26

mentation. Crucially, despite this acceptable latency, they suffer from poor generalization due to the noise in sparse reconstructions. In contrast, MVGGT unifies these steps into a single differentiable pass, reducing the processing time to just 0.3 s. This represents a significant $2\times$ to $3\times$ speedup in algorithmic efficiency over the baselines, achieving a total system latency of ~ 2.3 s. This efficiency enables robust perception in near real-time, meeting the strict speed and accuracy requirements of dynamic embodied agents.

Inference Memory Usage. Beyond temporal efficiency, MVGGT also optimizes computational resource utilization. As shown in Table III, MVGGT consumes only 8.39 GB of GPU memory, representing a significant 25%–32% reduction in memory footprint compared to the other two methods. By unifying geometric reconstruction and linguistic grounding into a single pass, MVGGT avoids the redundant feature caching required by sequential pipelines, ensuring that robust 3D perception remains feasible.

4. Extended Capability Study on ReferIt3D

To further assess the robustness and generalization capability of MVGGT, particularly in fine-grained discrimination tasks, we extend our evaluation to the ReferIt3D benchmarks: Sr3D (Spatial Reference) and Nr3D (Natural Reference). Unlike ScanRefer which contains unique objects, ReferIt3D focuses on distinguishing a target from multiple distractors of the same class (e.g., “the chair on the left” vs. “the chair on the right”), placing significantly higher demands on spatial reasoning. We conduct this evaluation under the challenging sparse-view protocol and benchmark MVGGT against the 2D-Lift baseline to demonstrate its superior spatial awareness and view-consistency.

Table IV. Performance comparison on ReferIt3D benchmarks. We report Global $mIoU$ and View-dependent $mIoU$. The “Easy” and “Hard” subsets follow the MV-3DRES definition.

Method	Easy _{MV-3DRES}		Hard _{MV-3DRES}		View-Dep		View-Indep		Overall	
	Global	View	Global	View	Global	View	Global	View	Global	View
Nr3D (Natural Reference)										
2D-Lift	22.1	25.4	6.8	21.5	12.0	22.0	14.9	24.1	13.9	23.4
MVGGT (Ours)	36.6	64.4	19.0	65.7	26.9	64.7	28.4	65.3	27.9	65.1
Sr3D (Spatial Reference)										
2D-Lift	21.4	24.0	5.9	19.8	8.1	18.1	13.2	21.9	13.0	21.7
MVGGT (Ours)	31.2	39.9	15.5	30.0	22.2	34.2	24.3	35.5	24.2	35.5

Results and Analysis. Table IV presents a comprehensive comparison where MVGGT consistently outperforms

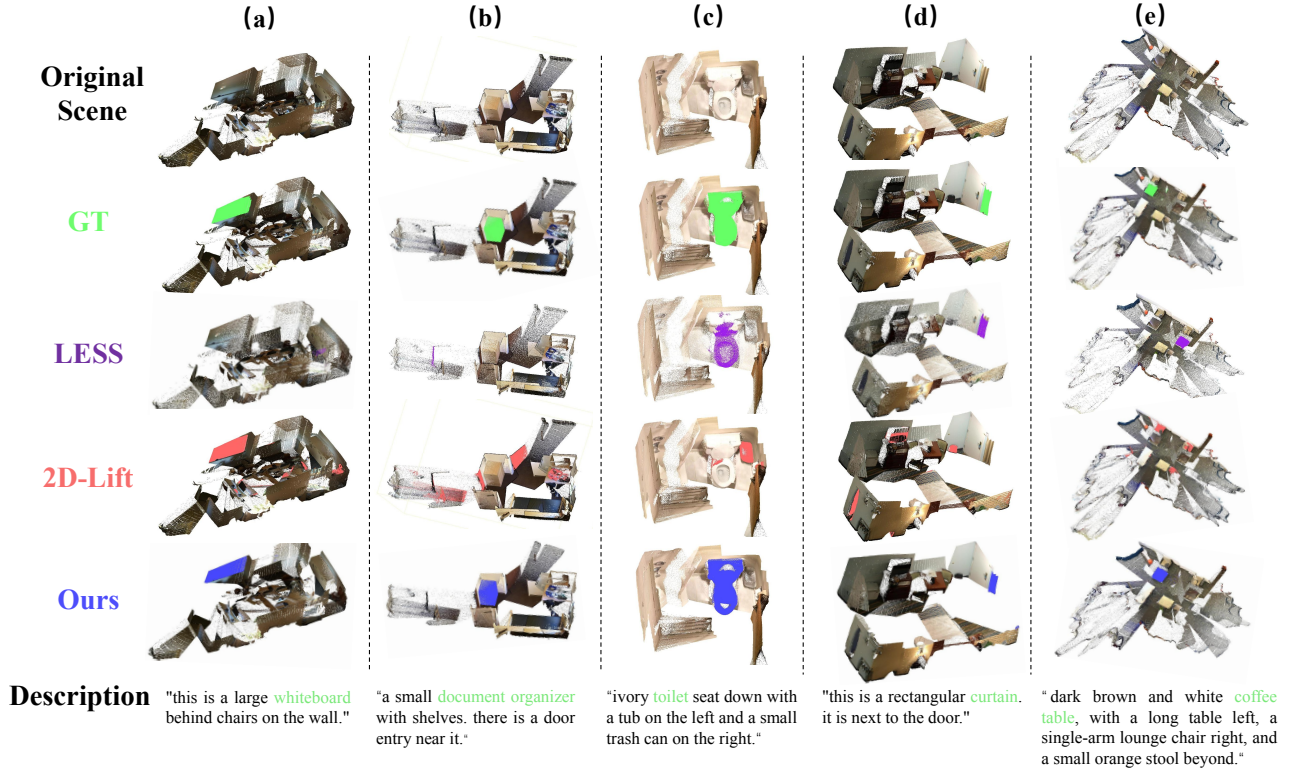


Figure I. **Qualitative comparison on the MVRefer benchmark (Part 1).** We visualize additional failure cases of baselines and the success of MVGGT.

the 2D-Lift baseline across all metrics. Specifically, on the Sr3D benchmark which relies heavily on spatial predicates, 2D-Lift achieves only 13.0 overall $mIoU_{global}$. Since it operates on isolated images, it lacks the global geometric context required to resolve spatial ambiguities that vary across viewpoints. In contrast, MVGGT nearly doubles the performance to 24.2, confirming that our geometric branch effectively encodes relative spatial positions even from sparse inputs. This robustness is most pronounced in the $Hard_{MV-3DRES}$ split, where objects have minimal geometric evidence. While 2D-Lift collapses to a single-digit accuracy of 5.9 on Sr3D, indicating a failure to localize targets without salient visual cues, MVGGT maintains a robust performance of 15.5. This stability stems from our PVS0 strategy that stabilizes optimization against sparse gradients, coupled with the multimodal fusion mechanism that allows linguistic cues to bridge the gap caused by missing geometry. Furthermore, on Nr3D, MVGGT achieves a remarkable improvement in $mIoU_{view}$, jumping from 23.4 to 65.1. This demonstrates that unlike 2D methods which fail to maintain consistency across the sequence, MVGGT’s end-to-end aggregation ensures accurate semantic understanding across diverse viewing angles.

5. Additional Qualitative Results

To provide a more intuitive understanding of the performance gap between MVGGT and existing approaches, we present extensive qualitative comparisons in Figure I and Figure II. As observed across these examples, the 2D-Lift baseline suffers severely from spatial inconsistency. Since it predicts masks on isolated 2D views without global 3D constraints, the back-projected 3D masks often appear fragmented, consisting of disjointed patches floating in space. Similarly, the LESS [2] method exhibits extreme sensitivity to reconstruction quality. In our sparse setting where point clouds contain holes and floating artifacts, LESS [2] often fails to recover complete object extents or mistakes background noise for the target. In stark contrast, MVGGT demonstrates superior robustness, producing stable and unified segmentations that align accurately with the ground truth.

In Figure I, example (a) demonstrates that while LESS [2] fails to identify the object entirely and 2D-Lift generates incoherent patches, MVGGT leverages linguistic context to correctly localize the flat whiteboard. In example (b), where both baselines fail to segment the document organizer, MVGGT successfully isolates the target; this preci-

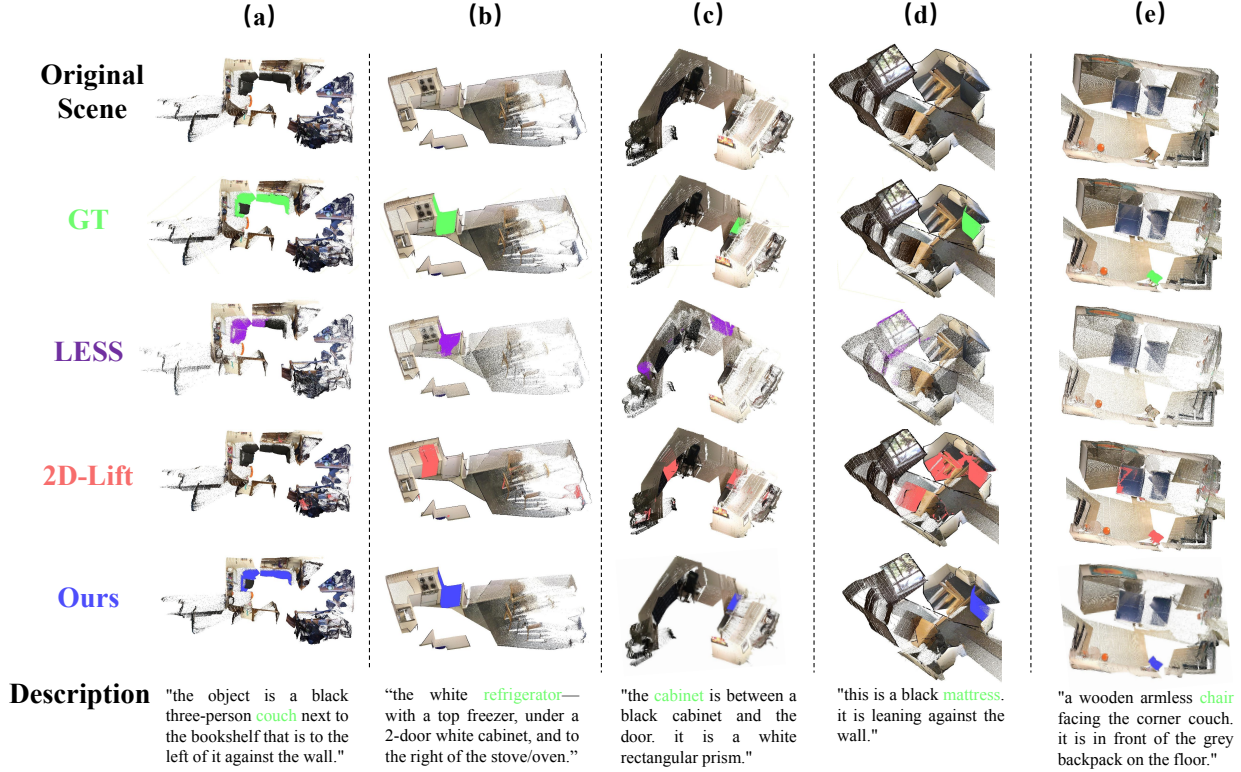


Figure II. **Qualitative comparison on the MVRefer benchmark (Part 2).** More examples demonstrating the robustness of our method under sparse views.

sion is explicitly supported by PVSO’s balanced per-view supervision, which prevents the weak foreground signal from being overwhelmed. Example (c) contains severe reconstruction noise around the toilet; LESS [2] yields only a broken mask, and 2D-Lift, overwhelmed by artifacts, erroneously predicts background objects to the right. In contrast, MVGGT recovers the complete shape, enabled by the dual-branch design that refines geometry with textual cues. Similarly, in example (d), while the LESS [2] mask remains fragmented and 2D-Lift produces scattered predictions, MVGGT successfully distinguishes the curtain from the adjacent door frame. Finally, in example (e), LESS [2] misidentifies the object and 2D-Lift produces disjointed fragments, whereas MVGGT achieves a complete segmentation of the coffee table.

Moving to Figure II, for the large couch in example (a), LESS [2] generates a fragmented mask, while 2D-Lift fails to detect the object entirely; in contrast, MVGGT recovers the full volume through global cross-view reasoning. In example (b), LESS [2] similarly yields a broken mask, whereas 2D-Lift incorrectly predicts the oven to the left of the target. Regarding example (c), LESS [2] fails to produce a valid prediction, and while 2D-Lift locates the object, it

erroneously includes significant background noise. However, MVGGT achieves precise segmentation. A similar trend is observed for the mattress leaning against the wall in example (d), where LESS [2] erroneously misidentifies the background as the target due to geometric ambiguity, whereas MVGGT accurately captures its planar structure. Finally, in example (e), despite the low point density and partial visibility, MVGGT successfully identifies the small chair, demonstrating robust cross-view consistency.

6. Robustness to View Sparsity

Scaling with View Numbers. We performed a scaling ablation on view numbers to evaluate the impact of input sparsity on MVGGT. As shown in Table V, the mIoU_{global} improves consistently from 34.3% ($N = 2$) to 41.9% ($N = 16$). Notably, MVGGT remains robust even at $N = 4$, demonstrating strong resilience to extreme sparsity. This ability to maintain stable grounding with limited visual evidence confirms the efficacy of our dual-branch architecture in real-world sensing conditions.

Table V. Ablation on the number of input views (N).

Number of Views	2	4	8 (Default)	16
mIoU _{global}	34.3	34.9	39.9	41.9

7. Evaluation on Various Backbones

To evaluate the backbone dependence of our framework, we replace the default Pi3 backbone with alternative reconstruction models, including VGGT [4], DUST3R [5], and MAST3R [1]. Although performance dropped a bit with these backbones, our method still worked well and trained successfully across all of them. These solid experiments support our claims that the framework is robust and flexible, demonstrating that MVGGT can effectively integrate linguistic cues into geometric scaffolds without relying on a specific backbone.

Table VI. Ablation Study on Different Backbone Architectures.

Backbone	Pi3	VGGT	DUST3R	MAST3R
mIoU _{global}	39.9	36.8	35.3	34.4

8. Evaluation of Loss Strategies

Comparison against Imbalance-aware Losses. To evaluate our optimization strategy, we compare PVSO with representative imbalance-aware losses. As shown in Table VII, these results confirm that leveraging dense 2D priors is significantly more effective than simple re-weighting for resolving the sparse 3D Foreground Gradient Dilution (FGD) problem.

Table VII. Comparison of Loss Functions.

Loss	PVSO	Focal Loss	WCE	Lovasz
mIoU _{global}	39.9	35.5	34.1	32.0

PVSO Hyperparameter Ablation. We also investigate the impact of the balancing weight λ_p between dense 2D supervision and sparse 3D signals. We set $\lambda_p = 1.0$ as the default to balance gradient magnitudes. As summarized in Table VIII, while the performance remains relatively stable, $\lambda_p = 1.0$ yields the best result (39.9% mIoU_{global}), validating our choice for stable end-to-end training.

Table VIII. Ablation on PVSO balancing weight λ_p .

Weight λ_p	0.1	0.5	1.0 (Default)	2.0	5.0
mIoU _{global} (%)	37.8	39.7	39.9	38.9	37.2

9. Robustness to Lighting Extremes

MVGGT is robust to lighting variations in practice. We evaluate the framework under challenging lighting conditions, including both strong-light and low-light scenarios. As illustrated in Fig. III, even when image quality or shape details degrade due to these lighting extremes, our multi-modal reasoning remains effective by compensating for visual artifacts. This synergy between geometric scaffolding and linguistic cues ensures stable and accurate predictions where purely visual or geometric methods might drift.

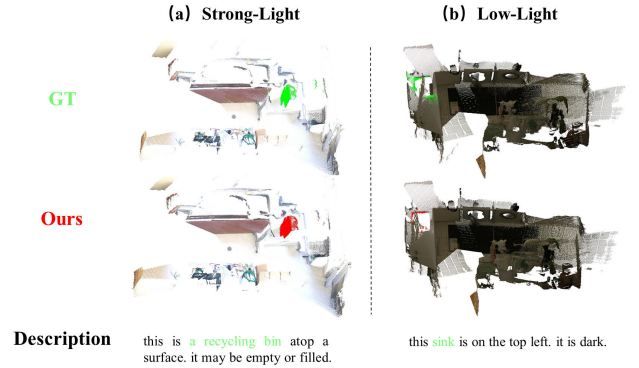


Figure III. Visualizations under Lighting Extremes.

10. Failure Case Analysis

Despite its strong performance, we identify three typical failure modes of MVGGT that suggest directions for future work: (a) *Semantic Ambiguity*—difficulty distinguishing identical instances based on subtle text cues; (b) *Motion Artifacts*—quick camera movement causes blur, making 3D reconstruction unreliable; (c) *Target Absence*—hallucinating targets when objects are not present. We acknowledge that handling motion artifacts and target-absent scenarios is a critical capability for practical deployment, and we mark these as primary directions for future research.

References

- [1] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r, 2024. 5
- [2] Xuexun Liu, Xiaoxu Xu, Jinlong Li, Qiudan Zhang, Xu Wang, Nicu Sebe, and Lin Ma. Less: Label-efficient and single-stage referring 3d segmentation. *arXiv:2410.13294*, 2024. 1, 2, 3, 4
- [3] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1
- [4] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 5

- [5] 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. [5](#)
- [6] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. Pi3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025. [1](#), [2](#)
- [7] Changli Wu, Yihang Liu, Jiayi Ji, Yiwei Ma, Haowei Wang, Gen Luo, Henghui Ding, Xiaoshuai Sun, and Rongrong Ji. 3d-gres: Generalized 3d referring expression segmentation. In *ACM MM*, 2024. [1](#), [2](#)