SwiftVGGT: Scalable Visual Geometry Grounded Transformer for Large-Scale Scenes

Supplementary Material

A. How Do VGGT's DINO Enable VPR?

Although VGGT [15] was never designed for Visual Place Recognition (VPR), we find that its DINO [10]-based encoder naturally contains several properties that make it unexpectedly effective for place-level matching once appropriately normalized. In this section, we analyze why this phenomenon occurs and explain how our lightweight post-processing turns VGGT's reconstruction-centric tokens into competitive loop descriptors.

VGGT's DINO Tokens. The VGGT encoder is initialized from DINO, a self-supervised vision transformer known to produce highly semantic, spatially consistent patch tokens. These tokens capture mid-level structures such as building facades, vegetation patterns, and road geometry–exactly the type of cues that are robust to illumination changes and viewpoint variations in VPR. Whereas traditional VPR models [1, 3, 7, 17] explicitly learn such invariances through contrastive training, VGGT inherits them "for free" from DINO. This provides a strong initialization for transforming the patch tokens into global descriptors.

VGGT's Frame/Global Attention. Unlike standalone DINO, VGGT is trained for multi-view 3D reasoning using both frame attention and global attention. These mechanisms encourage *(1) view-to-view feature consistency*, *(2) stable representations for the same region observed under different poses*, *(3) suppression of transient or dynamic details*. This implicit geometric alignment effect makes the encoder features more stable across large camera motions, which is valuable for long-range place recognition.

Normalization. Despite these strengths, raw VGGT tokens cannot be used directly for VPR because: (1) they contain strong low-frequency scene priors (dataset bias); (2) Dominant feature directions lead to hubness, where unrelated frames appear spuriously similar; (3) VGGT is optimized for 3D reconstruction, not discrimination across places. Hence, naive averaging of DINO tokens performs poorly as shown in Tab. 2. A transformation is required to emphasize discriminative directions while suppressing dataset-shared ones.

Our transformation relies only on classical, training-free normalization techniques. First, signed power normalization step reduces the dominance of large-magnitude components and amplifies weaker ones. Although reminiscent of RootSIFT [2], classical VPR methods rarely apply it on transformer global descriptors. In our case, it significantly stabilizes the descriptor distributions across scenes. Second, while PCA whitening is common in retrieval, it is almost always applied together with a learned VPR backbone, not directly on features from a 3D foundation model. Our key observation is that *"VGGT's DINO patch tokens become VPR-discriminative once their shared low-frequency directions are removed."*

To the best of our knowledge, no prior work has demonstrated that DINO tokens from a 3D reconstruction model can function as reliable VPR descriptors without any fine-tuning. This training-free pipeline is crucial for SwiftVGGT, as it avoids the heavy cost of an additional VPR encoder while maintaining loop-closure accuracy comparable to fully trained models.

Novelty. The novelty of our Sec. 3.3 is not merely applying normalization tricks, but lies in the following:

- Discovering that VGGT's 3D transformer already embeds stable place-level signals, despite no VPR supervision.
- Showing a minimal, training-free normalization pipeline unlocks these signals.
- Replacing a dedicated VPR model entirely, reducing computation and enabling the large speedup reported in the experiments.

B. Why We Use PIN-SLAM as Pseudo GT

A key challenge in evaluating camera tracking accuracy on the KITTI Odometry dataset [6] is that ground-truth poses are available only for sequences 00–10, while sequences 11–21 do not provide any pose annotations. To enable quantitative evaluation on these sequences, we adopt the poses produced by PIN-SLAM [11] as pseudo ground truth.

This choice is justified by the remarkable robustness of PIN-SLAM, which leverages LiDAR measurements to estimate trajectories. Unlike purely vision-based SLAM [8, 14] systems that may drift significantly in long or texture-sparse environments, PIN-SLAM benefits from accurate geometric constraints derived from high-resolution LiDAR scans. As shown in Table 1, its trajectory error remains extremely small across all kilometer-scale KITTI scenes, consistently approaching near-zero ATE values even in challenging driving scenarios. This performance is well aligned with prior observations that LiDAR-based SLAM methods can maintain long-term consistency without suffering from cumulative drift.

Because PIN-SLAM provides stable and precise poses over large-scale sequences where no official ground truth exists, its trajectories serve as a reliable surrogate for evalu-

Table 1. Comparison of ATE RMSE performance between the LiDAR-based method (PIN-SLAM) and RGB-based methods.

Methods	Modality	00†	01	02†	03	04	05†	06†	07†	08†	09†	10	Average
DPV-SLAM [8]	RGB	112.80	11.50	123.53	2.50	0.81	57.8	54.86	18.77	110.49	76.66	13.65	53.03
DROID-SLAM [14]	RGB	92.10	344.6	107.61	2.38	1.00	118.5	62.47	21.78	161.60	72.32	118.70	100.28
VGGT-Long [4]	RGB	9.87	111.06	37.56	4.89	3.75	9.09	7.47	4.02	62.86	47.48	25.49	29.41
SwiftVGGT (**Ours**)	RGB	8.17	102.53	36.49	8.12	4.88	11.94	8.88	5.01	64.68	44.13	26.18	29.18
PIN-SLAM [11]	LiDAR	**0.91**	**3.26**	**2.15**	**0.43**	**0.81**	**0.25**	**0.16**	**0.25**	**0.18**	**1.20**	**0.56**	**1.07**

ating loop detection and camera tracking accuracy. Consequently, using PIN-SLAM as pseudo ground truth allows us to conduct fair and meaningful comparisons across all KITTI scenes, including those without provided annotations, while avoiding distortions that would arise from qualitative-only assessments.

C. Depth Normalization to the Ref. Intrinsic

Our reliability-guided point sampling Sec. 3.2 relies on the observation that, after rescaling all depth maps to a common reference intrinsic, overlapping frames from neighboring temporal chunks produce almost identical depth values in static regions. Here we briefly derive the depth normalization in Eq. (1) of the main paper.

Consider the standard pinhole camera model with focal lengths f_x and f_y. Let D_{src} denote the depth map estimated by VGGT under the source intrinsic $(f_{x,\text{src}}, f_{y,\text{src}})$, and let D_{ref} be the depth that would be obtained under the reference intrinsic $(f_{x,\text{ref}}, f_{y,\text{ref}})$ while keeping the underlying 3D geometry fixed. Ignoring lens distortion and assuming a locally planar pixel grid, the depths are related by a simple scale factor:

$$D_{\text{ref}} \approx \frac{1}{2}\left(\frac{f_{x,\text{ref}}}{f_{x,\text{src}}} + \frac{f_{y,\text{ref}}}{f_{y,\text{src}}}\right) D_{\text{src}}. \tag{1}$$

We adopt this symmetric scaling to account for potential anisotropy in f_x and f_y. Empirically, this normalization makes the overlapping regions between neighboring temporal chunks highly consistent in depth, which is crucial for our subsequent reliability-guided sampling strategy.

D. Reliability-Guided Point Sampling

After normalizing depths to the reference intrinsic, most pixels in the overlapping regions of neighboring temporal chunks already exhibit small depth discrepancies, while large deviations are concentrated near object boundaries and occlusion edges. Our reliability-guided point sampling exploits this structure by retaining only pixels whose depth difference and confidence satisfy

$$\begin{aligned} &\left|D_{\text{ref},t} - D_{\text{ref},t+1}\right| < \lambda_D, \\ &\gamma_t > \lambda_\gamma \mu_{\gamma_t}, \\ &\gamma_{t+1} > \lambda_\gamma \mu_{\gamma_{t+1}}. \end{aligned} \tag{2}$$

In practice, our sampling strategy does not aggressively sparsify the overlap region. Instead, it selects pixels whose depths, after being normalized to the reference intrinsic, are mutually consistent between neighboring chunks and have sufficiently high confidence according to VGGT's depth-confidence head. This effectively removes depth-discontinuous boundaries and low-confidence regions, while retaining the geometrically stable portions of the overlap. As a result, the correspondences fed to the Umeyama solver are both reliable and well aligned, which explains the substantial runtime reduction reported in Tab. 5 of the main paper and the improved ATE compared to using all points.

E. Additional Ablation Studies.

We conduct further ablation experiments to supplement the analyses presented in the main paper.

Loop detection. Tab. 2 reports an ablation study on the loop detection pipeline using the 12 KITTI sequences containing loops. Removing loop detection entirely eliminates the need for loop-centric chunk inference, reducing runtime by approximately 16 seconds, but results in a substantial degradation of ATE (over 16 m). A naive baseline that directly averages VGGT's DINO patch tokens followed by ℓ_2 normalization achieves better accuracy than the no-loop variant, but still performs worse than our full pipeline due to incorrect loop associations. Adding signed power normalization improves performance by producing more balanced and discriminative descriptors. Finally, incorporating PCA whitening yields the best overall ATE, confirming that both power normalization and whitening are necessary to transform VGGT patch tokens into reliable loop descriptors without requiring a dedicated VPR encoder.

Reliability-guided point sampling. Tab. 3 presents a more detailed ablation of our reliability-guided point sampling strategy. Using all overlap pixels leads to suboptimal alignment accuracy and slightly higher runtime due to the large number of point correspondences. Applying only the depth difference threshold already produces the best accuracy, demonstrating that depth alignment alone is a strong reliability indicator. We additionally retain depth confi-

Table 2. Additional ablation study on loop detection methods.

Method	Average ATE (m)	Elapsed Time (s)
w/o Loop Detection	44.01	**126.35**
w/ Token Avg. + ℓ_2 Norm.	34.39	142.98
w/ Signed Power	27.77	143.13
Full (Ours)	**27.33**	142.67

Table 3. Additional ablation study on loop detection methods.

Method	Average ATE (m)
w/ IRLS	29.23
w/ Whole Points	32.95
w/ Depth Diff. Only	**28.42**
w/ Depth Conf. Only	30.23
Full (Ours)	29.18

dence filtering because, despite a marginal decrease in average ATE, it yields more stable performance across all scenes and generalizes better to the Waymo Open [13] and Virtual KITTI datasets [5]. This trade-off makes it the preferred configuration in our final model.

F. Additional Results

Virtual KITTI dataset. We evaluate our method on the Virtual KITTI dataset [5], and the quantitative results are summarized in Tab. 5. Although DROID-SLAM [14] achieves the best average ATE, it completely fails to track camera poses in the `0006_rain` sequence. CUT3R [16] and FastVGGT [12] show poor tracking results, exceeding 10 m ATE in most sequences, and FastVGGT additionally runs out of GPU memory on `0020`. In contrast, both VGGT-Long [4] and our method maintain consistently strong performance across all scenes. Notably, SwiftVGGT achieves slightly better accuracy while being significantly faster, demonstrating its robustness even in short, loop-free sequences.

Waymo Open dataset. Following the evaluation protocol described in the main paper, we report 3D reconstruction metrics in Tab. 4. Since the ground-truth point clouds are generated directly from Waymo [13] LiDAR measurements and camera parameters, their density and coverage differ from the reconstructed outputs. As a result, completeness scores are generally low across all methods. MASt3R-SLAM and CUT3R yield relatively large errors, whereas FastVGGT, VGGT-Long, and SwiftVGGT achieve comparably strong results. Considering the point cloud metrics in this table together with the camera tracking performance and runtime comparisons in Tab. 3, SwiftVGGT offers the most favorable balance of accuracy and efficiency.

KITTI Scene 11–20. Using pseudo ground-truth trajectories generated by PIN-SLAM [11], we evaluate scenes 11–20 of the KITTI dataset [6], covering both loop and non-loop sequences. The results are presented in Tab. 6. Regardless of whether loops are present, SwiftVGGT consistently outperforms prior SLAM-based and VGGT-based methods. These results highlight that our loop detection mechanism remains effective when loops exist, but does not degrade performance when they do not, further demonstrating the general applicability of our approach.

Table 4. Performance comparison on the Waymo Open dataset [13]. Acc. and Comp. denote accuracy and completeness, respectively; CD stands for Chamfer Distance.

Methods	163453191			183829460			315615587			346181117			371159869		
	Acc.	Comp.	CD	Acc.	Comp.	CD	Acc.	Comp.	CD	Acc.	Comp.	CD	Acc.	Comp.	CD
MASt3R-SLAM [9]	0.475	35.440	17.960	0.560	17.790	9.180	1.003	12.210	6.608	1.554	9.980	5.767	0.810	32.001	16.405
CUT3R [16]	3.015	8.071	5.543	0.485	11.700	6.093	1.667	8.556	5.117	3.216	11.450	7.333	2.801	13.062	7.932
FastVGGT [12]	0.349	3.157	1.753	0.305	8.058	4.182	0.738	4.845	2.792	0.882	4.601	2.741	0.937	8.170	4.554
VGGT-Long [4]	0.401	4.319	2.360	0.200	9.489	4.844	0.222	5.409	2.816	0.326	4.932	2.629	1.113	7.880	4.497
Ours	0.492	4.171	2.331	0.595	9.150	4.872	0.329	5.181	2.755	0.451	5.552	3.001	1.234	7.215	4.224

Methods	405841035			460417311			520018670			610454533			Average		
	Acc.	Comp.	CD	Acc.	Comp.	CD	Acc.	Comp.	CD	Acc.	Comp.	CD	Acc.	Comp.	CD
MASt3R-SLAM [9]	0.284	11.630	5.959	1.109	5.080	3.095	1.015	12.810	6.913	0.840	5.889	3.365	0.850	15.870	8.361
CUT3R [16]	2.008	7.947	4.977	1.335	6.501	3.918	1.567	9.337	5.452	1.003	35.220	18.110	1.900	12.428	7.164
FastVGGT [12]	0.243	6.668	3.455	0.707	2.827	1.767	0.624	9.748	5.186	0.774	5.204	2.989	0.618	5.920	3.269
VGGT-Long [4]	0.185	7.434	3.810	0.197	3.604	1.900	0.251	9.558	4.904	0.487	5.287	2.887	0.376	6.435	3.405
Ours	0.151	7.570	3.861	0.547	3.172	1.859	0.333	9.488	4.910	0.440	5.537	2.989	0.508	6.337	3.422

Table 5. Performance comparison on the Virtual KITTI dataset [5]. The orange and yellow cells respectively indicate the highest and second-highest value.

Methods	Scene 0001						Scene 0002						Scene 0006					
	Clone	Fog	Morning	Overcast	Rain	Sunset	Clone	Fog	Morning	Overcast	Rain	Sunset	Clone	Fog	Morning	Overcast	Rain	Sunset
DROID-SLAM [14]	1.027	1.868	0.989	1.015	0.776	1.145	0.098	0.040	0.049	0.048	0.036	0.113	0.063	0.024	0.030	0.051	TL	0.0197
CUT3R [16]	43.304	62.191	50.608	38.744	51.548	43.785	23.771	9.948	28.415	24.644	7.963	25.973	0.836	0.408	0.599	0.720	1.059	1.013
FastVGGT [12]	57.097	58.584	59.552	56.839	58.383	57.999	2.958	2.997	2.946	2.965	2.974	2.955	3.731	3.812	3.731	3.742	3.792	3.732
VGGT-Long [4]	0.838	1.288	0.791	0.755	1.474	0.894	0.702	0.674	0.734	0.662	0.673	0.656	0.435	0.549	0.430	0.439	0.534	0.444
Ours	0.888	1.170	1.668	0.797	1.431	1.792	0.725	0.637	0.747	0.630	0.689	0.687	0.398	0.433	0.396	0.398	0.461	0.409

Methods	Scene 0018						Scene 0020						Average					
	Clone	Fog	Morning	Overcast	Rain	Sunset	Clone	Fog	Morning	Overcast	Rain	Sunset	0001	0002	0006	0018	0020	Average
DROID-SLAM [14]	2.478	2.032	1.894	2.332	2.549	1.943	3.592	5.079	3.733	3.852	3.780	4.907	1.137	0.064	0.038	2.205	4.157	1.571
CUT3R [16]	19.440	8.628	6.720	20.212	16.777	31.119	129.498	76.962	117.948	114.512	66.700	116.529	48.363	20.119	0.772	17.149	103.692	38.019
FastVGGT [12]	4.781	4.795	4.662	5.449	4.570	4.707	OOM	OOM	OOM	OOM	OOM	OOM	58.076	2.966	3.762	4.827	OOM	17.406
VGGT-Long [4]	1.778	0.796	1.324	1.425	1.900	1.998	13.094	14.762	6.649	3.460	5.191	3.749	1.006	0.684	0.472	1.537	7.817	2.303
Ours	1.747	0.787	1.516	1.653	1.572	1.780	9.599	14.087	3.141	5.969	5.031	6.339	1.291	0.686	0.416	1.509	7.361	2.253

Table 6. Performance comparison on the KITTI Odometry dataset [6] Scene 11–20. The † denotes sequences containing loops, and *Dense*$^{\diamond}$ indicates semi-dense 3D reconstruction.

Methods	Calib.	Recon.	11	12	$13^{\dagger}$	14	$15^{\dagger}$	$16^{\dagger}$	17	$18^{\dagger}$	$19^{\dagger}$	20	Average
DPV-SLAM [8]	✓	*Sparse*	26.00	57.06	53.73	9.72	51.20	34.15	57.29	17.96	229.36	8.73	54.52
DROID-SLAM [14]	✓	*Dense*$^{\diamond}$	17.18	9.61	48.40	8.93	39.97	30.41	0.86	11.58	182.82	1.04	35.08
VGGT-Long [4]	✗	*Dense*	38.45	25.93	7.76	12.98	6.33	18.71	22.22	6.84	118.10	10.13	26.75
Ours	✗	*Dense*	38.77	36.42	6.72	8.38	6.30	16.27	17.59	7.60	111.73	16.73	26.65

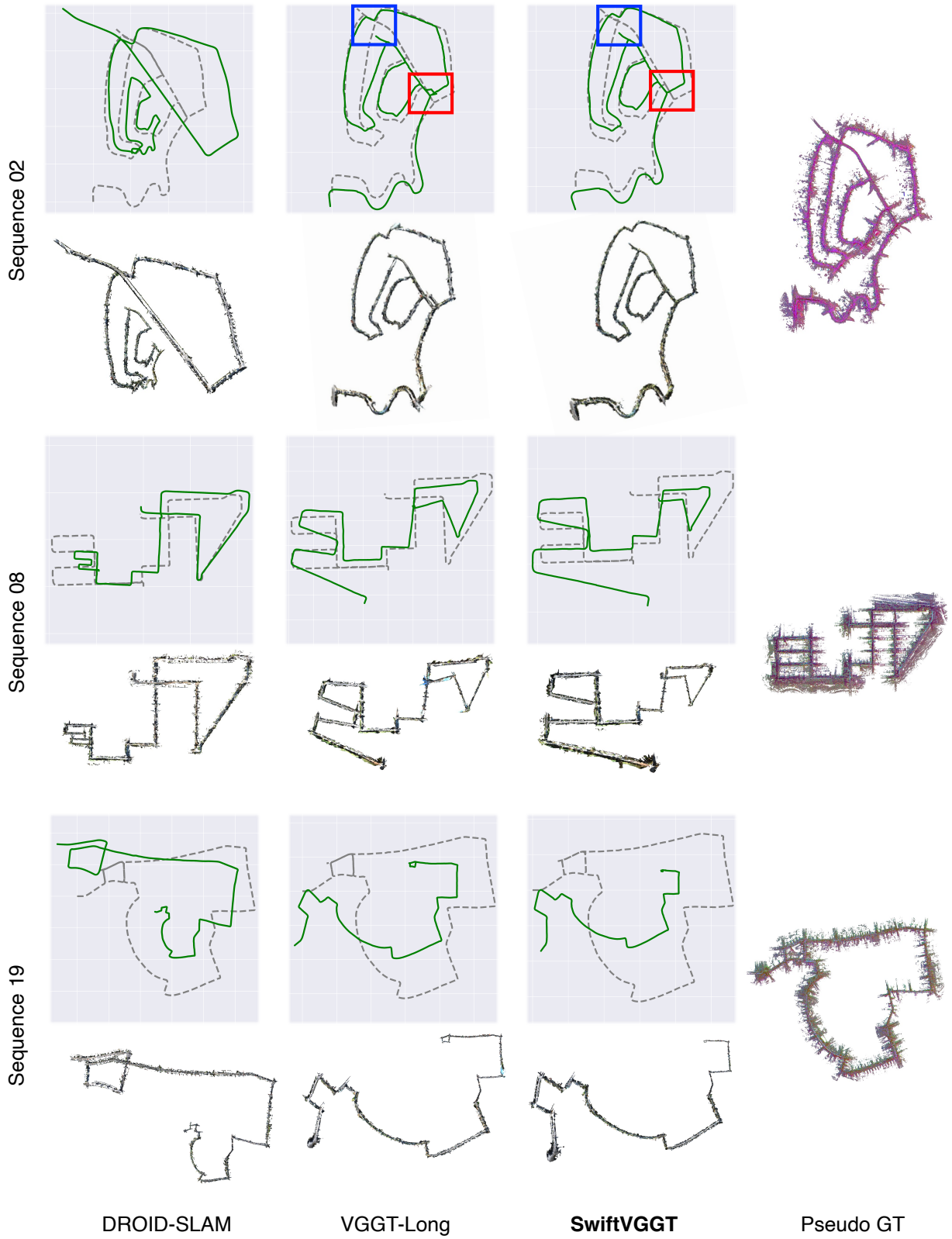

Figure 1. Failure cases of KITTI scenes. The pseudo ground-truth point cloud is obtained by combining the LiDAR point cloud with camera poses either provided by the dataset or estimated by PIN-SLAM [11]. The **gray** dashed line represents the ground-truth trajectory, while the **green** solid line denotes the estimated trajectory.

G. Failure Case Analysis

In this section, we analyze failure cases observed in several KITTI [6] sequences. As illustrated in Fig. 1, SwiftVGGT, DROID-SLAM [14], and VGGT-Long [4] all exhibit degraded performance on sequences 02, 08, and 19, both quantitatively and qualitatively. A common characteristic of these sequences is that loop closure is not successfully detected. For instance, in KITTI 02, DROID-SLAM completely diverges, while VGGT-Long incorrectly detects non-existent loops (**red** box in Fig. 1) and simultaneously misses true loops (**blue** box). In contrast, our method avoids hallucinated loops but still fails to identify the missing true loops. Similarly, in sequences 08 and 19, all three methods consistently fail to detect valid loop closures. Interestingly, as shown in Fig. 2, relaxing the loop detection threshold enables SwiftVGGT to correctly detect the loop in sequence 19, resulting in noticeably improved reconstruction quality. However, this adjustment does not reliably solve all failure cases across all sequences. Nonetheless, these results suggest that VGGT's DINO transformer features can outperform classical VPR encoders when appropriately processed, even without training or fine-tuning. A promising direction for future work is to incorporate feature-level or correspondence-level loop detection strategies (*e.g.*, point tracking or feature matching) to further improve robustness under challenging loop conditions.

Finally, beyond loop detection failures, some sequences still exhibit relatively high ATE (greater than 10m even when loops are correctly found). As discussed in Sec. 5 of the main paper, this limitation largely stems from the absence of bundle adjustment, meaning accumulated drift remains uncorrected. Integrating a lightweight or learned BA module into SwiftVGGT may address this remaining challenge and enable more precise large-scale reconstruction.

H. Point Cloud Visualization

We provide qualitative dense reconstruction results on the KITTI [6], Waymo Open [13], and Virtual KITTI [5] datasets in Figures Fig. 3, Fig. 4, and Fig. 5. The reconstructed point clouds are visualized with a depth-based color map, where the color corresponds to the z-axis value of each point. These results demonstrate that, in addition to achieving fast inference and state-of-the-art tracking accuracy, SwiftVGGT is capable of producing high-quality dense 3D reconstructions across diverse large-scale driving scenarios.

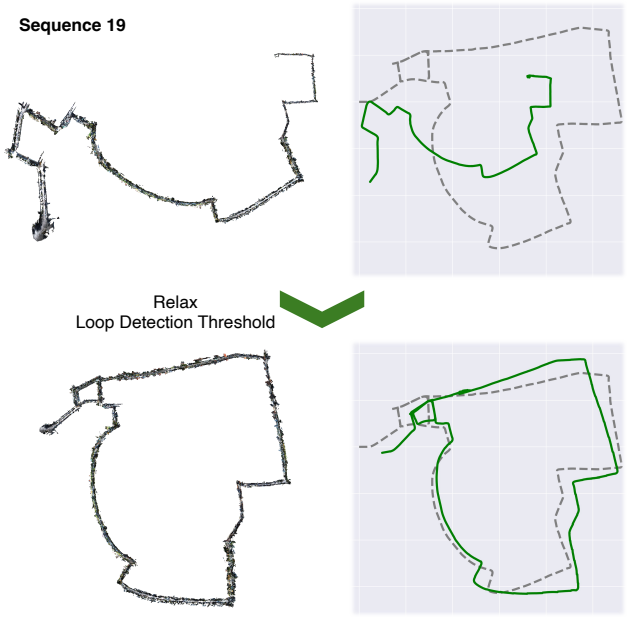

Figure 2. Relaxing loop detection threshold on KITTI sequence 19.

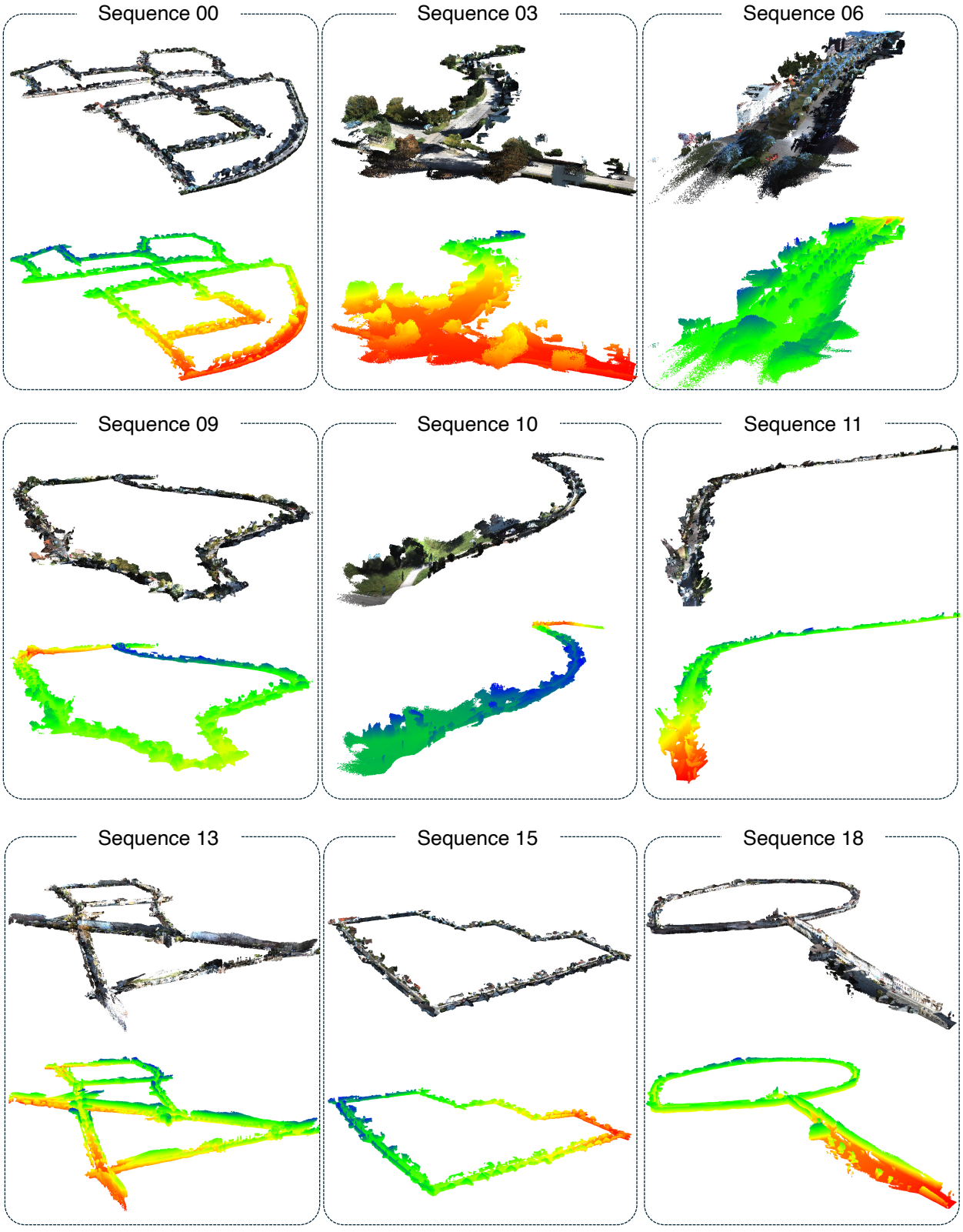

Figure 3. Point cloud visualization of KITTI dataset [6].

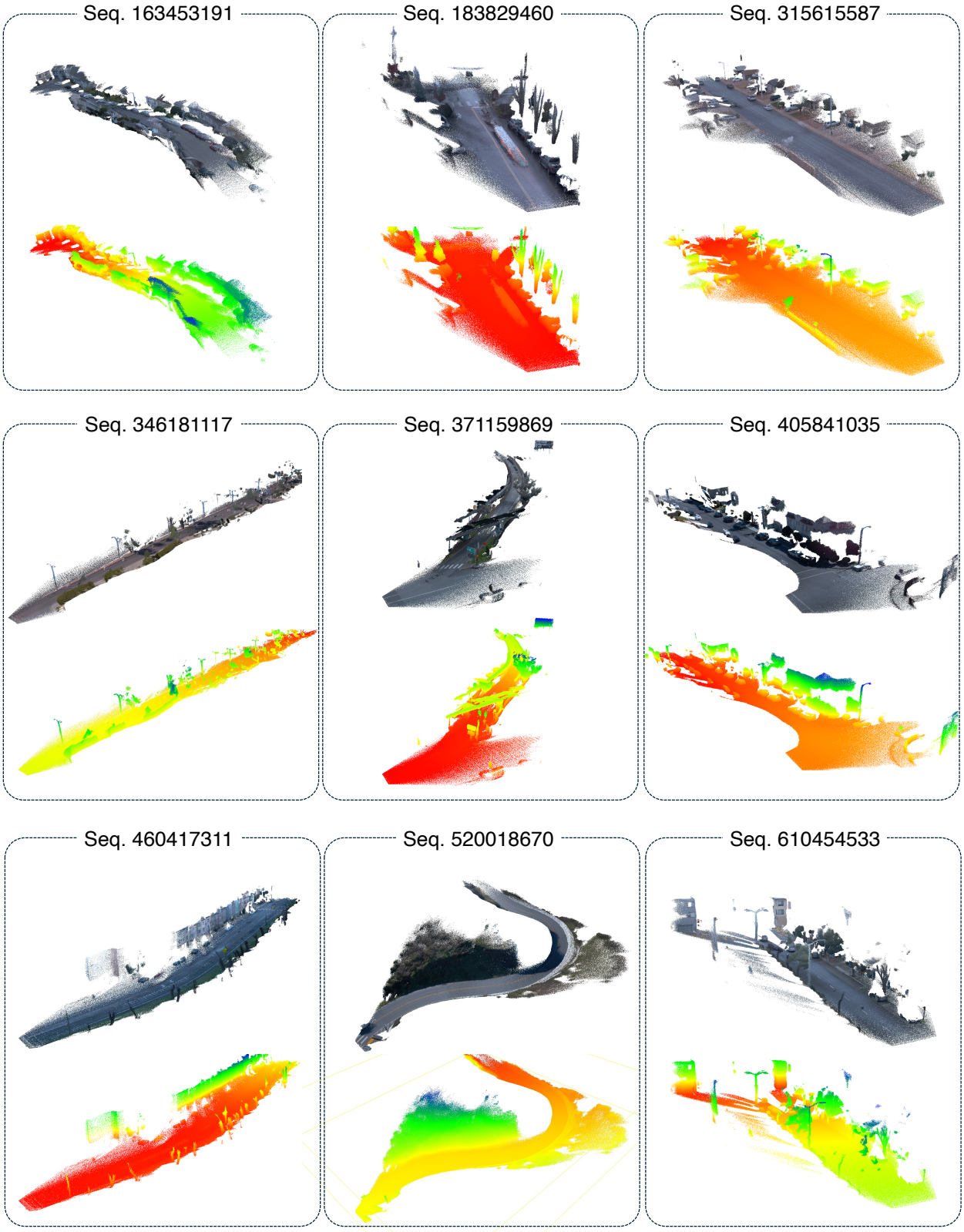

Figure 4. Point cloud visualization of Waymo Open dataset [13].

Figure 5. Point cloud visualization of Virtual KITTI dataset [5].

References

[1] Amar Ali-Bey, Brahim Chaib-Draa, and Philippe Giguere. Mixvpr: Feature mixing for visual place recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2998–3007, 2023. 1

[2] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, pages 2911–2918, 2012. 1

[3] Gabriele Berton, Carlo Masone, and Barbara Caputo. Rethinking visual geo-localization for large-scale applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4878–4888, 2022. 1

[4] Kai Deng, Zexin Ti, Jiawei Xu, Jian Yang, and Jin Xie. Vggt-long: Chunk it, loop it, align it–pushing vggt's limits on kilometer-scale long rgb sequences. *arXiv preprint arXiv:2507.16443*, 2025. 2, 3, 4, 6

[5] A Gaidon, Q Wang, Y Cabon, and E Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016. 3, 4, 6, 9

[6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 1, 3, 4, 6, 7

[7] Sergio Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 17658–17668, 2024. 1

[8] Lahav Lipson, Zachary Teed, and Jia Deng. Deep patch visual slam. In *European Conference on Computer Vision*, pages 424–440. Springer, 2025. 1, 2, 4

[9] Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16695–16705, 2025. 4

[10] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 1

[11] Yue Pan, Xingguang Zhong, Louis Wiesmann, Thorbjörn Posewsky, Jens Behley, and Cyrill Stachniss. Pin-slam: Lidar slam using a point-based implicit neural representation for achieving global map consistency. *IEEE Transactions on Robotics*, 40:4045–4064, 2024. 1, 2, 3, 5

[12] You Shen, Zhipeng Zhang, Yansong Qu, and Liujuan Cao. Fastvggt: Training-free acceleration of visual geometry transformer. *arXiv preprint arXiv:2509.02560*, 2025. 3, 4

[13] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 3, 4, 6, 8

[14] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021. 1, 2, 3, 4, 6

[15] 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. 1

[16] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. *arXiv preprint arXiv:2501.12387*, 2025. 3, 4

[17] Ruotong Wang, Yanqing Shen, Weiliang Zuo, Sanping Zhou, and Nanning Zheng. Transvpr: Transformer-based place recognition with multi-level attention aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13648–13657, 2022. 1