Geometry-Preserving in 3D Gaussian Splatting for LiDAR-Camera Extrinsic Calibration: Supplementary Material

A Implementation Details

A.1 Hyperparameter Details

This section provides the hyperparameter settings used in GeoP-Calib. Unless otherwise noted, all experiments in Sec. 5 are conducted using the hyperparameter settings listed in Table S1.

Table S1: Hyperparameter settings used in our experiments.

Parameter	Value	Description
iterations	10000	Maximum number of optimizer steps.
voxel_res	0.1	Voxel size used for initialization.
optimizer_type	sparse_adam	Gaussian parameter optimizer type.
lambda_dssim	0.2	Mixing ratio between L1 and SSIM in the photometric loss.
position_lr_init	1.60E-04	Initial learning rate for Gaussian center positions (xyz).
feature_lr	2.50E-03	Learning rate for feature-related parameters.
opacity_lr	2.50E-02	Learning rate for opacity parameters.
scaling_lr	5.00E-03	Learning rate for Gaussian scaling parameters.
rotation_lr	1.00E-03	Learning rate for Gaussian rotation parameters.
pose_rotation_lr	3.00E-03	Learning rate for extrinsic rotation parameters.
pose_translation_lr	3.00E-02	Learning rate for extrinsic translation parameters.
Pose optimizer step interval	level $\leq$ 2: 10, level 3: 30	Frequency of extrinsic parameter updates.
Multi-scale resolution stages	[4., 2., 1., 1.]	Order of camera scale / image pyramid levels used during training.
Stage transition iterations	[0, 2000, 4000, 8000]	Iterations at which the optimization stage is changed.
Pose LR scale per stage	[1., 0.5, 0.5, 0.1]	Stage-wise scaling factor for the extrinsic optimizer learning rate.

Parameter	Value	Description
match window	2	Range of reference frames used for the matching loss.
min_depth	0.1	Lower bound of the valid depth range for supervision.
max_depth	50	Upper bound of the valid depth range for supervision.
lambda_depth	10	Weight for sparse depth anchoring loss
lambda_dense	10	Weight for dense depth anchoring loss
lambda_match	1	Weight for temporal matching loss
lambda_scale	0.03	Weight for scale regularization
dense_stage	1 (2000 iterations)	Stage at which the dense depth loss is activated.
dense_ratio	0.05	Relative margin applied to the rendered depth for dense depth supervision.
dense_sharp	100	Sigmoid sharpness parameter for dense depth supervision.

B Experiment Details

B.1 Dataset Details

We evaluate our method on the KITTI-360 [3] and KITTI odometry [1] datasets (hereafter, KITTI), following the evaluation protocols of HiGS-Calib [4] and RobustCalib [6]. For KITTI-360, we use the same five sequences as HiGS-Calib: *Straight Line* (897), *Small ZigZag* (2907), *Small Rotation* (86), *Large ZigZag* (10948), and *Large Rotation* (2743). For KITTI, we randomly select five sequences from the set used in RobustCalib: 5-300, 6-100, 7-250, 9-540, and 10-300. We refer to these sequences as seq1–5 in the order listed above.

All methods are trained using 30 frames from the beginning of each sequence. For KITTI-360, we follow HiGS-Calib and report the average performance over cam0 and cam1. For KITTI, we follow RobustCalib and use only cam2 for evaluation.

B.2 Baseline Implementation Details

This section describes the implementation details of the baselines used in Sec. 5. We use the official implementations released by the authors for all baselines in our experiments. For GST [2], we make a minimal modification to enable the use of ground-truth LiDAR poses for a fair comparison.

B.3 Evaluation Protocol

This section describes the evaluation protocol used in Sec. 5.3. Unless otherwise specified, we initialize the extrinsic parameters of all methods except GST with $\mathbf{R} = \mathbf{I}$ and

Table S2: HiGS-Calib executed with seed 0 on KITTI-360.

	seq 1	seq 2	seq 3	seq 4	seq 5	Avg.
E_r (°)	0.096	0.189	0.100	0.114	0.140	0.128
E_t (m)	0.035	0.086	0.115	0.083	0.099	0.083

Table S3: Hardware configurations used in our experiments. CLAIM [5], RobustCalib, HiGS-Calib, and GeoP-Calib were evaluated in the same cluster environment, whereas GST was evaluated in a separate Docker-based environment.

Item	**Shared Environment**	**GST Environment**
OS	Ubuntu 20.04.6 LTS	Ubuntu 22.04.5 LTS (Docker)
Kernel	5.4.0-216-generic	5.15.0-139-generic
CPU	Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz	AMD Ryzen 5 3600 6-Core Processor
Allocated CPU	6 cores (64 logical CPUs available on the node)	12 logical CPUs
RAM	29 GiB allocated (440 GiB available on the node)	46 GiB
GPU	NVIDIA RTX A5000	NVIDIA GeForce RTX 4070 Ti
GPU Memory	24564 MiB	12282 MiB
Driver Version	535.247.01	570.181
CUDA Version (driver)	12.2	12.8

$\mathbf{t} = \mathbf{0}$. For GST, due to the sparsity of LiDAR points, SuperGlue fails to operate reliably. Therefore, we instead perturb the initial extrinsic parameters with rotation and translation noise of $|\mathbf{R}| = 1.0°$ and $|\mathbf{t}| = 0.2$, respectively. These values are determined based on the accuracy of the edge-based method used as a baseline in GST.

For CLAIM, considering that it is based on grid search, we set sufficiently wide search ranges so that the ground-truth extrinsic parameters are included in the search space for each dataset: ($|\mathbf{R}| = 20.0°, |\mathbf{t}| = 1.0$) for KITTI-360 and ($|\mathbf{R}| = 10.0°, |\mathbf{t}| = 0.2$) for KITTI. All evaluations are conducted using the same five fixed random seeds (178132, 423337, 922852, 787576, 660993) across all methods and datasets. The final reported extrinsic parameters are taken from the last optimization step, i.e., the values obtained at the end of the program. To clarify the discrepancy between our reported HiGS-Calib results and those in its original paper [4], we note that the official code fixes the random seed to 0, whereas we report the mean and standard deviation over the five seeds above for reliability. Table S2 reports HiGS-Calib run with its default seed-0 setting, which closely matches the originally reported numbers.

Table S4: Software environment used for HiGS-Calib and GeoP-Calib.

Item	HiGS-Calib / GeoP-Calib
OS	Ubuntu 20.04.6 LTS
Python	3.12.12
PyTorch	2.9.0+cu126
CUDA	12.6 (PyTorch), 12.2 (driver)
cuDNN	91002

Table S5: Detailed component analysis of GeoP-Calib on KITTI-360. We report sequence-wise calibration errors for each component configuration. Values in parentheses denote the standard deviation. Note that VSM is a sub-component of DDA and is not defined without it.

GD	DDA	VSM	seq1		seq2		seq3		seq4		seq5		Avg.	
			$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)
			0.120 (0.016)	0.037 (0.006)	0.156 (0.017)	0.084 (0.003)	0.078 (0.020)	0.089 (0.033)	0.117 (0.035)	0.077 (0.002)	0.137 (0.013)	0.098 (0.002)	0.122 (0.031)	0.077 (0.022)
	✓	✓	0.097 (0.043)	0.038 (0.005)	0.148 (0.022)	0.062 (0.006)	0.069 (0.015)	0.066 (0.019)	0.102 (0.037)	0.074 (0.001)	0.166 (0.011)	0.098 (0.002)	0.117 (0.044)	0.068 (0.021)
✓			0.140 (0.017)	0.033 (0.006)	0.172 (0.025)	0.081 (0.004)	0.062 (0.033)	0.055 (0.008)	0.136 (0.037)	0.073 (0.002)	0.124 (0.025)	0.093 (0.002)	0.127 (0.042)	0.067 (0.022)
✓	✓		0.132 (0.026)	0.029 (0.006)	0.156 (0.023)	0.076 (0.002)	0.056 (0.047)	0.052 (0.003)	0.129 (0.030)	0.072 (0.001)	0.147 (0.030)	0.101 (0.002)	0.124 (0.045)	0.066 (0.024)
✓	✓	✓	0.124 (0.046)	0.031 (0.006)	0.145 (0.020)	0.059 (0.007)	0.074 (0.030)	0.059 (0.002)	0.110 (0.042)	0.071 (0.001)	0.153 (0.030)	0.093 (0.003)	0.121 (0.041)	0.063 (0.021)

B.4 Hardware and Software Details

This section describes the hardware and software settings used in the experiments presented in Sec. 5.3. We evaluate all methods except GST on the same hardware setup, which is summarized in Table S3. For software, we follow the environment settings provided by the official implementations for all methods. The software environment used for HiGS-Calib and our method is summarized in Table S4.

B.5 Detailed Results for Ablation Studies

Table S5 presents the detailed per-sequence results corresponding to Table 2 in the main paper.

B.6 Additional Details for Geometric Accuracy Analysis

This section provides additional details on the geometric accuracy analysis in Sec. 5.5. After training, we use the optimized Gaussians to evaluate the Mean Absolute Error (MAE) between the rasterized sparse LiDAR depths and the rendered depths at the corresponding pixel locations. Consistent with the depth anchoring setup, the evaluation is performed using the 30 training views and only for points within a distance of 50 m. The sequence-wise geometric accuracy results are presented in Table S6. Table S6 shows that GeoP-Calib achieves overall lower geometric error than the baseline,

Table S6: Detailed geometric accuracy analysis on KITTI-360. We report E_{depth}, defined as the mean absolute error (MAE) between rasterized sparse LiDAR depths and rendered depths at the corresponding pixel locations. Values in parentheses denote the standard deviation.

Method	seq1	seq2	seq3	seq4	seq5	Avg.
	E_{depth}(m)	E_{depth}(m)	E_{depth}(m)	E_{depth}(m)	E_{depth}(m)	E_{depth}(m)
BASE	0.1866 (0.0103)	0.2733 (0.0156)	0.3008 (0.0763)	0.3665 (0.0203)	0.3454 (0.0162)	0.2945 (0.0722)
GeoP-Calib	0.1372 (0.0024)	0.2751 (0.0024)	0.2133 (0.0020)	0.3011 (0.0054)	0.2967 (0.0025)	0.2447 (0.0623)

Table S7: Total runtime comparison on KITTI-360. All methods are evaluated under the same hardware setup with one run per sequence. Values in parentheses denote the standard deviation across sequences.

Method	**CLAIM**	**RobustCalib**	**HiGS-Calib**	**GeoP-Calib**
Runtime (s)	1708 (5)	1185 (122)	854 (24)	922 (21)

while also exhibiting reduced standard deviation across sequences. This suggests that GeoP-Calib stabilizes the geometric structure of the Gaussians during optimization.

C Additional Experiments

C.1 Runtime Analysis

We measure the total runtime of CLAIM, RobustCalib, HiGS-Calib, and GeoP-Calib on all KITTI-360 sequences using one run per sequence. Due to resource constraints, this runtime analysis is conducted in a separate environment from the one used for the main quantitative experiments, using an AMD Ryzen 5 3600 CPU, an NVIDIA RTX 4070 Ti 12GB GPU, 48GB RAM, and Ubuntu 20.04. The results are summarized in Table S7. GeoP-Calib is 46.0% faster than CLAIM and 22.1% faster than RobustCalib on average. Compared with HiGS-Calib, GeoP-Calib requires 68 additional seconds on average, corresponding to an 8.0% increase in runtime.

C.2 Hyperparameter Analysis for VSM

We conduct a sensitivity analysis on the two VSM hyperparameters defined in Eq. 11, namely the depth-proportional tolerance margin (τ) and the sharpness parameter (β). All experiments are conducted on seq2 and seq5 of the KITTI-360 dataset using only cam0 and three random seeds: 178132, 423337, and 922852.

Table S8 presents the sensitivity results for the VSM hyperparameters. For both sequences, varying β leads to similar performance and does not exhibit a clear trend, suggesting that our method is relatively insensitive to the sharpness parameter. In contrast, the effect of τ is more sequence-dependent. While seq2 does not show a clear

Table S8: Sensitivity analysis of the VSM hyperparameters on KITTI-360. Values in parentheses denote the standard deviation.

Seq.	τ	$\beta = 25$		$\beta = 50$		$\beta = 100$		$\beta = 200$		$\beta = 400$	
		$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)
seq2	0.025	0.1278	0.0517	0.1301	0.0517	0.1444	0.0532	0.1429	0.0524	0.1281	0.0521
		(0.0123)	(0.0011)	(0.0062)	(0.0019)	(0.0056)	(0.0008)	(0.0167)	(0.0014)	(0.0205)	(0.0027)
	0.05	0.1394	0.0525	0.1433	0.0528	0.1396	0.0531	0.1413	0.0520	0.1451	0.0525
		(0.0145)	(0.0009)	(0.0105)	(0.0017)	(0.0119)	(0.0015)	(0.0128)	(0.0014)	(0.0114)	(0.0014)
	0.1	0.1330	0.0537	0.1311	0.0525	0.1340	0.0537	0.1449	0.0530	0.1290	0.0539
		(0.0220)	(0.0020)	(0.0084)	(0.0011)	(0.0078)	(0.0006)	(0.0135)	(0.0008)	(0.0094)	(0.0009)
seq5	0.025	0.1620	0.0944	0.1585	0.0939	0.1581	0.0939	0.1738	0.0950	0.1697	0.0951
		(0.0126)	(0.0015)	(0.0168)	(0.0008)	(0.0084)	(0.0018)	(0.0095)	(0.0012)	(0.0150)	(0.0020)
	0.05	0.1723	0.0960	0.1732	0.0955	0.1783	0.0953	0.1800	0.0956	0.1801	0.0957
		(0.0091)	(0.0014)	(0.0202)	(0.0014)	(0.0088)	(0.0013)	(0.0271)	(0.0006)	(0.0323)	(0.0016)
	0.1	0.1846	0.0976	0.1907	0.0965	0.1811	0.0975	0.1847	0.0967	0.1879	0.0969
		(0.0117)	(0.0012)	(0.0158)	(0.0005)	(0.0113)	(0.0003)	(0.0218)	(0.0012)	(0.0168)	(0.0015)

Table S9: Optimization schedule analysis for DDA. The table compares calibration errors when $\mathcal{L}_{depth}^{dense}$ is activated at different stages. Values in parentheses denote the standard deviation.

Seq.	Stage 0 (Start)		Stage 1 (Default)		Stage 2 (Late)	
	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)
seq2	0.139	0.051	0.142	0.053	0.145	0.055
	(0.008)	(0.002)	(0.009)	(0.001)	(0.015)	(0.003)
seq5	0.177	0.095	0.171	0.096	0.178	0.097
	(0.025)	(0.001)	(0.018)	(0.001)	(0.007)	(0.002)

trend, seq5 tends to achieve lower rotation and translation errors as the tolerance margin decreases. This indicates that a smaller margin can lead to improved accuracy in certain scenarios Overall, the performance remains reasonably stable across different parameter choices, indicating that GeoP-Calib is not overly sensitive to the VSM hyperparameters.

C.3 Optimization Schedule Analysis for Dense Depth Anchoring

We conduct this experiment under the same setting as in Sec. C.2, while varying the stage at which DDA is activated. As shown in Table S9, none of the tested schedules leads to a severe performance drop. However, for seq5, activating DDA from the beginning without a sparse depth anchoring warm-up slightly increases the rotation error and its standard deviation, indicating less stable behavior. Therefore, we choose Stage 1 as the default setting to achieve a better balance between performance and stability.

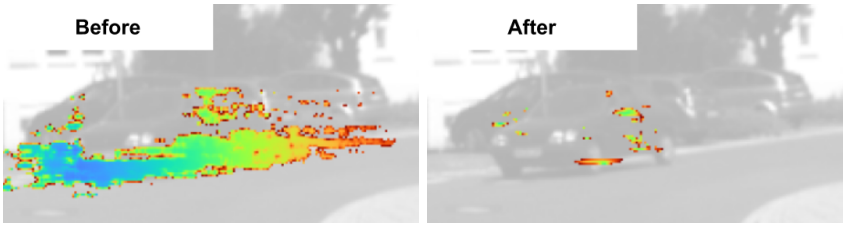
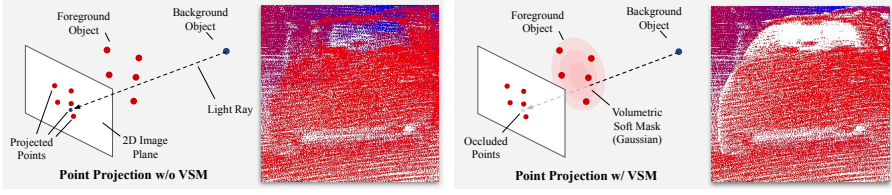


Fig. S1: Dynamic object Gaussians in KITTI-360 Seq. 4 before (left) and after (right) optimization. Only Gaussians within bounding boxes of dynamic vehicles are visualized.



(a) **(b)**

Fig. S2: Illustration of the VSM. (a) w/o VSM. (b) w/ VSM.

C.4 Dynamic Object Sensitivity Analysis

Since DDA accumulates multi-view LiDAR observations, scenes containing dynamic objects can in principle produce ghost trails in the reconstructed geometry. However, such transient artifacts conflict with the multi-view photometric and reprojection consistency enforced during optimization: they cannot be jointly explained across views and therefore converge to low opacity, effectively fading out as optimization proceeds. Figure S1 visualizes the Gaussians within the bounding boxes of dynamic vehicles in KITTI-360 Seq. 4 before and after optimization, where the dynamic-object Gaussians are largely suppressed. The few residual artifacts that remain do not introduce significant conflict on the observed calibration views and thus do not dominate the extrinsic update. Consistently, on KITTI-360 Seq. 4, which contains a moving vehicle, our method achieves calibration accuracy comparable to the dataset average (Table S5). Nevertheless, this robustness is not guaranteed in scenes with a large number of dynamic objects, which we leave as a direction for future work.

D Additional Visualizations

D.1 Illustration of Volumetric Soft Mask

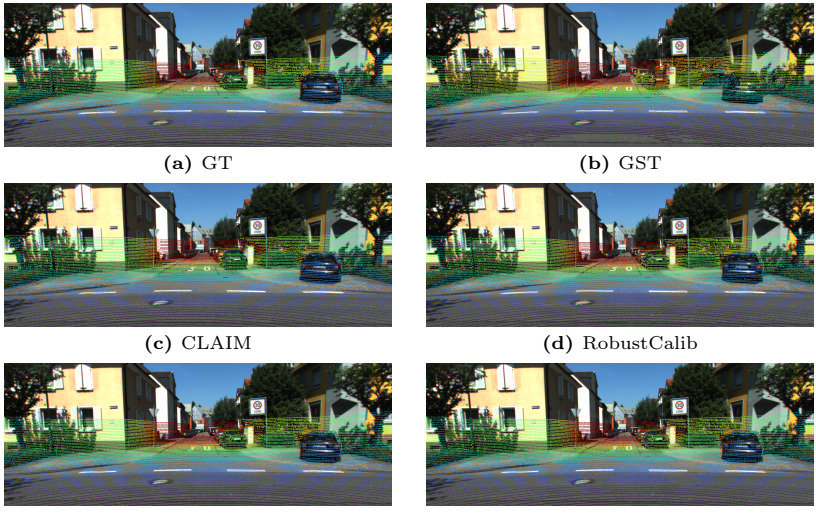
Figure S2 illustrates the effect of the VSM. Without VSM, LiDAR points projected behind foreground Gaussians may incorrectly contribute to supervision. In contrast, VSM suppresses points that are likely to be occluded by foreground geometry, leading to more geometrically consistent supervision.

Table S10: Worst-performing seed selected for LiDAR–camera overlay visualization on KITTI and KITTI-360. For each sequence, we report the seed that yields the largest rotation error, along with its corresponding rotation and translation errors. For KITTI-360, only Camera 0 is used for visualization.

	Seq.	GST [2]		CLAIM [5]		RobustCalib [6]		HiGS-Calib [4]		GeoP-Calib	
		Seed		Seed		Seed		Seed		Seed	
		$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)	$E_r(^\circ)$	E_t(m)
KITTI-360	seq1	922852		922852		787576		787576		423337	
		0.4488	0.3106	1.7035	0.5498	0.2753	0.0538	0.1371	0.0295	0.1918	0.0239
	seq2	922852		660993		178132		922852		423337	
		7.2006	0.6713	1.1604	0.1155	0.2866	0.1189	0.1607	0.0830	0.1480	0.0545
	seq3	423337		660993		178132		423337		660993	
		1.0948	0.6003	0.3686	0.0783	0.4429	0.0666	0.0626	0.1664	0.1084	0.0608
	seq4	423337		423337		660993		922852		787576	
		1.3228	0.6287	1.7777	0.6873	0.3725	0.1334	0.6104	0.4347	0.1631	0.0715
	seq5	787576		423337		660993		787576		178132	
		1.5919	0.2873	0.3287	0.0681	0.3441	0.0633	0.1532	0.0982	0.2017	0.0945
KITTI	seq1	423337		922852		423337		423337		922852	
		4.7169	0.4123	0.4024	0.0605	0.3579	0.0552	0.1251	0.0453	0.1153	0.0361
	seq2	423337		922852		660993		660993		660993	
		1.8504	0.3368	0.5033	0.1360	1.5552	0.0781	0.2037	0.0551	0.1725	0.0534
	seq3	660993		178132		178132		178132		423337	
		2.4040	0.1894	0.3144	0.0833	0.2782	0.1649	0.2609	0.0470	0.2439	0.0229
	seq4	178132		660993		787576		922852		660993	
		0.7905	0.2347	0.4872	0.1195	0.2689	0.0943	0.2812	0.0617	0.2784	0.0504
	seq5	660993		660993		787576		423337		922852	
		4.4159	0.5527	0.5141	0.0787	0.2842	0.1395	0.2131	0.0683	0.2256	0.0592

D.2 LiDAR-Camera Overlay Visualization

For each sequence in the KITTI and KITTI-360 datasets, we overlay LiDAR points on the RGB image using the extrinsic parameters from the worst-performing seed, identified by the rotation error. The selected seed and its corresponding rotation and translation errors are listed in Table S10. Figures S3–S7 present the visualizations for the KITTI-360 dataset, while Figures S8–S12 correspond to the KITTI dataset.

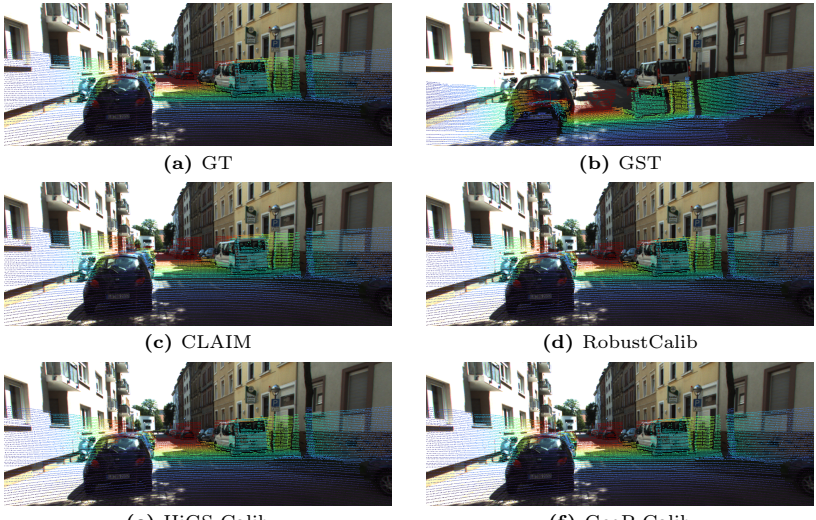


(a) GT **(b)** GST

(c) CLAIM **(d)** RobustCalib

(e) HiGS-Calib **(f)** GeoP-Calib

Fig. S3: LiDAR-camera overlay visualizations for KITTI-360 sequence 1.

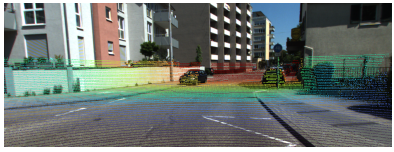


(a) GT **(b)** GST

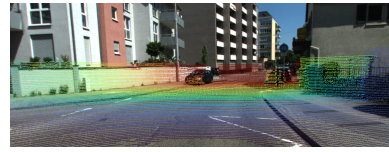
(c) CLAIM **(d)** RobustCalib

(e) HiGS-Calib **(f)** GeoP-Calib

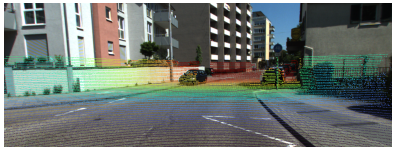
Fig. S4: LiDAR-camera overlay visualizations for KITTI-360 sequence 2.



(a) GT

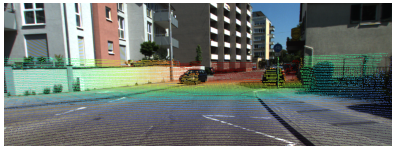


(b) GST

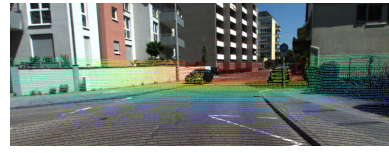


(c) CLAIM

(d) RobustCalib



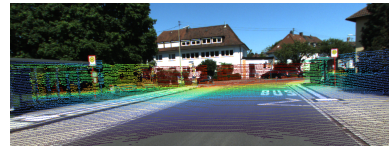
(e) HiGS-Calib



(f) GeoP-Calib

Fig. S5: LiDAR-camera overlay visualizations for KITTI-360 sequence 3.

(a) GT



(b) GST

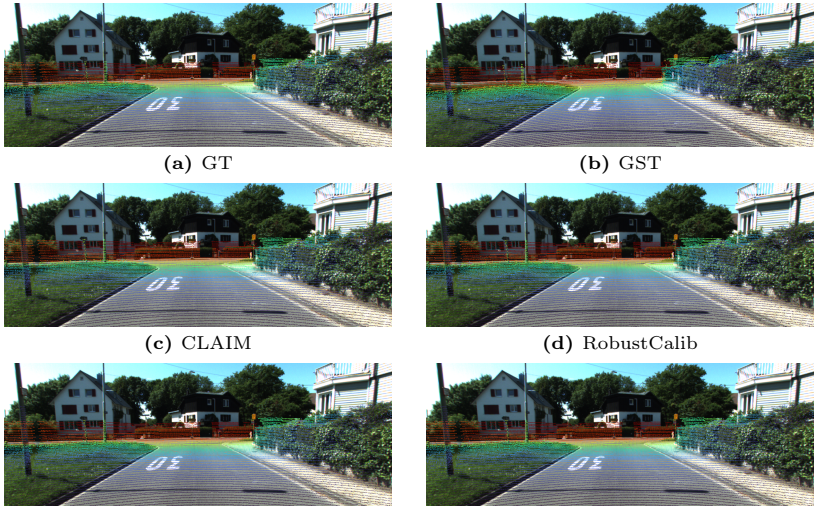
(c) CLAIM

(d) RobustCalib

(e) HiGS-Calib

(f) GeoP-Calib

Fig. S6: LiDAR-camera overlay visualizations for KITTI-360 sequence 4.

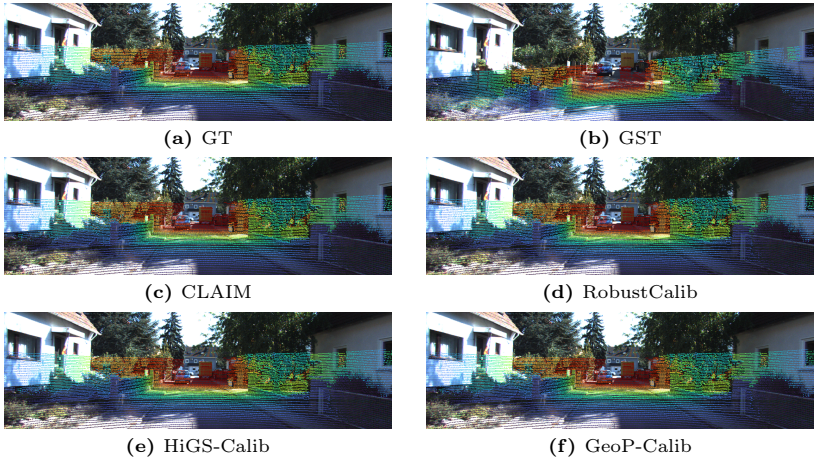


(a) GT **(b)** GST

(c) CLAIM **(d)** RobustCalib

(e) HiGS-Calib **(f)** GeoP-Calib

Fig. S7: LiDAR-camera overlay visualizations for KITTI-360 sequence 5.

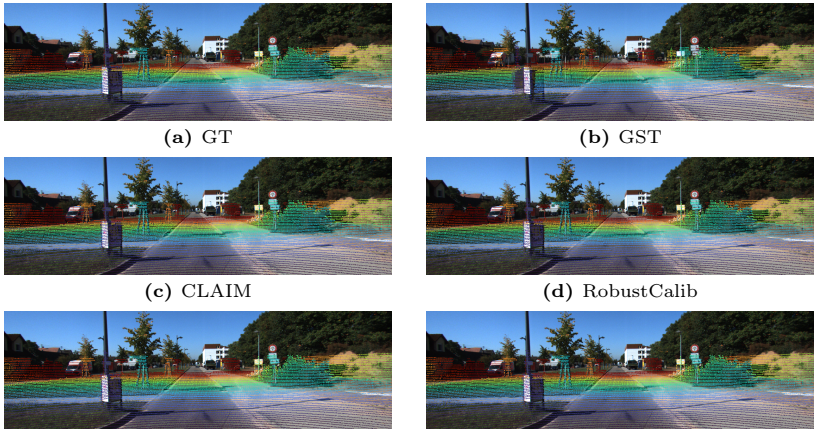


(a) GT **(b)** GST

(c) CLAIM **(d)** RobustCalib

(e) HiGS-Calib **(f)** GeoP-Calib

Fig. S8: LiDAR-camera overlay visualizations for KITTI sequence 1.

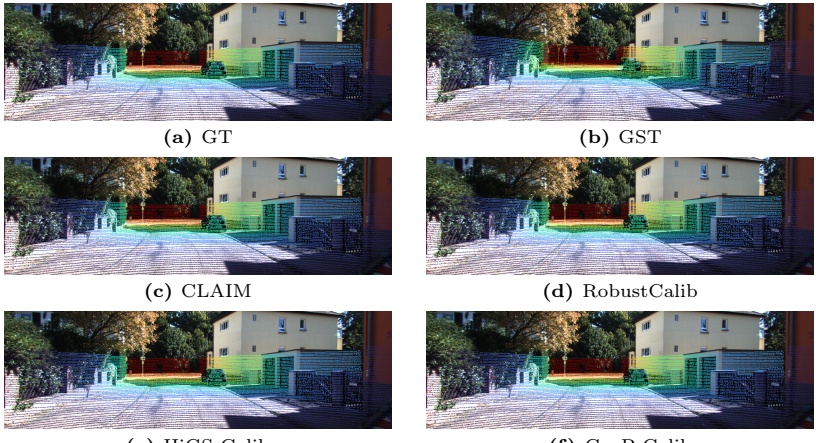


(a) GT **(b)** GST

(c) CLAIM **(d)** RobustCalib

(e) HiGS-Calib **(f)** GeoP-Calib

Fig. S9: LiDAR-camera overlay visualizations for KITTI sequence 2.

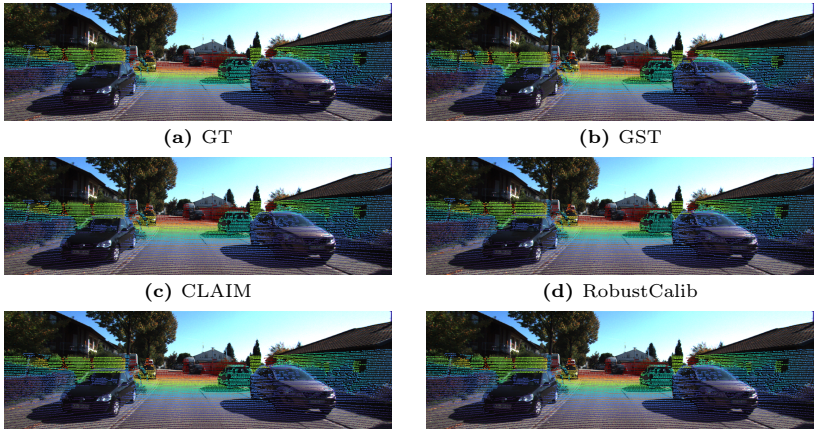


(a) GT **(b)** GST

(c) CLAIM **(d)** RobustCalib

(e) HiGS-Calib **(f)** GeoP-Calib

Fig. S10: LiDAR-camera overlay visualizations for KITTI sequence 3.

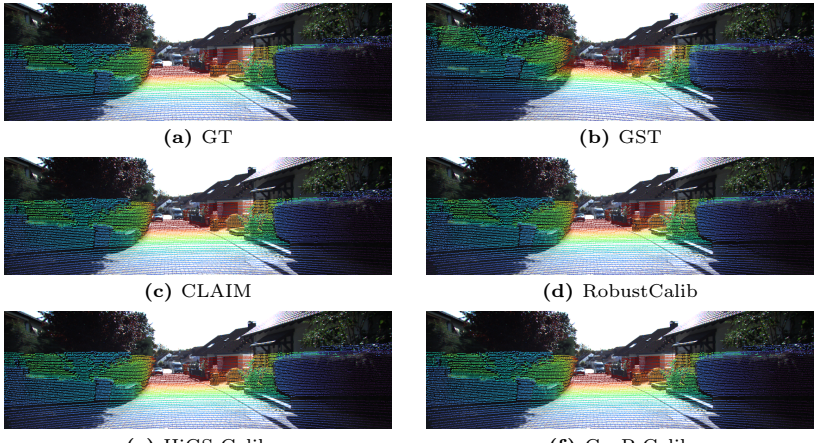


(a) GT **(b)** GST

(c) CLAIM **(d)** RobustCalib

(e) HiGS-Calib **(f)** GeoP-Calib

Fig. S11: LiDAR-camera overlay visualizations for KITTI sequence 4.



(a) GT **(b)** GST

(c) CLAIM **(d)** RobustCalib

(e) HiGS-Calib **(f)** GeoP-Calib

Fig. S12: LiDAR-camera overlay visualizations for KITTI sequence 5.

References

1. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012) 2
2. Koide, K., Oishi, S., Yokozuka, M., Banno, A.: General, single-shot, target-less, and automatic lidar-camera extrinsic calibration toolbox. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 11301–11307. IEEE (2023) 2, 8
3. Liao, Y., Xie, J., Geiger, A.: KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. arXiv preprint arXiv:2109.13410 (2021) 2
4. Zhang, T., Zhang, L., Wang, H.: Higs-calib: A hierarchical 3d gaussian splatting based targetless local-consistent lidar-camera calibration method. IEEE Transactions on Circuits and Systems for Video Technology (2025) 2, 3, 8
5. Zhang, Z., Liu, Y., Zhang, M., Tan, F., Ding, Y.: Claim: Camera-lidar alignment with intensity and monodepth. In: 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 17921–17926. IEEE (2025) 3, 8
6. Zhou, S., Xie, S., Ishikawa, R., Oishi, T.: Robust lidar-camera calibration with 2d gaussian splatting. IEEE Robotics and Automation Letters (2025) 2, 8