TrafficLoc: Localizing Traffic Surveillance Cameras in 3D Scenes Supplementary Material

Yan Xia*[1,2]† Yunxiang Lu*[2] Rui Song[2,4] Oussema Dhaouadi[2,5] João F. Henriques[6] Daniel Cremers[2,3]
[1]School of Artificial Intelligence and Data Science, University of Science and Technology of China
[2]Technical University of Munich [3]Munich Center for Machine Learning (MCML)
[4] Fraunhofer IVI [5] DeepScenario [6] Visual Geometry Group, University of Oxford

A. Overview

In this supplementary material, we provide a detailed explanations of our TrafficLoc and the proposed *Carla Intersection* dataset. Additionally, we present extended experimental results on the *Carla Intersection* dataset and KITTI Odometry dataset [5], showcasing the robust localization capabilities of our TrafficLoc and offering further insights we gathered during the development.

We begin by presenting the localization performance of our TrafficLoc on real-world datasets in Sec. B, and then show more experimental results and comprehensive ablation studies and analysis in Sec. C. In Sec. D, we outline the data collection process and provide visualizations of our *Carla Intersection* dataset. Sec. E describes the detailed elements of the Fusion Transformer in the GFF module, followed by Sec. F with implementation details of our network architecture and training procedure. Finally, Sec. G offers additional visualizations of our localization results across different datasets.

B. Evaluation on the real-world datasets

Figure 1 presents the qualitative localization result of our TraffifLoc on a real-world intersection from the USTC dataset [14]. Since the ground-truth pose is unavailable, we validate localization accuracy by projecting the point cloud onto the image plane with the predicted transformation matrix $[R|t]$ and intrinsic parameters K. The close alignment between the projected point cloud and the input image demonstrates strong Sim2Real generalization capability of our approach.

We further evaluate TrafficLoc on the OpenTrafficCam dataset [16], where intersection point clouds are reconstructed using COLMAP [13] following the official pipeline. As shown in Fig. 2 (a), our TrafficLoc maintains reasonable localization performance. Since the OpenTrafficCam dataset does not publicly release the specific data used in their experiments, we reconstructed one real-world traffic intersection scene using their provided scene reconstruction pipeline and align the scene with the real-world scale from GPS coordinates. Due to the sparsity of the point clouds obtained from COLMAP [13] and the resulting modality gap compared to real-world LiDAR data, we

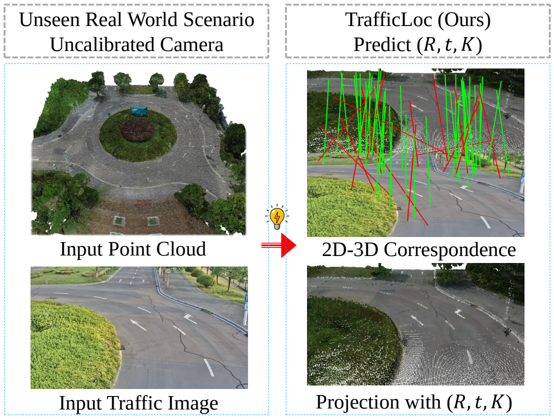


Figure 1. Localization performance of our TrafficLoc on the USTC intersection dataset [14]. Note that the model is trained on the synthetic *Carla Intersection* dataset.

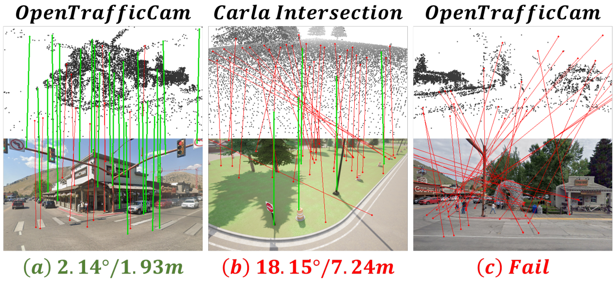


Figure 2. Additional localization results and failure case visualizations of our TrafficLoc on the Carla Intersection dataset and OpenTrafficCam dataset [16].

†Corresponding author, email address: yan.xia@ustc.edu.cn.
* Equal contribution.

	CM	FM	GAL	**Test**$_{T1-T7}$		**Test**$_{T1-T7hard}$		**Test**$_{T10}$	
				RRE(°)	RTE(m)	RRE(°)	RTE(m)	RRE(°)	RTE(m)
Baseline	NCL	/	/	1.53	0.82	3.79	1.98	7.35	7.47
	ICL	/	/	1.27	0.74	3.72	1.87	4.09	3.26
	ICL	/	✓	0.95	0.64	3.12	1.46	3.03	2.86
	ICL + DTA	/	/	1.01	0.62	3.23	1.65	2.99	2.80
	ICL + DTA	✓	/	0.84	0.62	3.17	1.42	2.98	2.83
Ours	ICL + DTA	✓	✓	**0.66**	**0.51**	**2.64**	**1.13**	**2.53**	**2.69**

Table 1. Ablation Study on loss function and model design. We report median RRE and median RTE results on all three test splits of *Carla Intersection* dataset. CM denotes Coarse Matching and FM denotes Fine Matching. "NCL" means using normal contrastive learning, while "ICL" means using the proposed Inter-intra Contrastive Learning. "DTA" indicates using the proposed Dense Training Alignment. "GAL" represents applying the proposed Geometry-guided Attention Loss.

further performed dense reconstruction to bridge this discrepancy. The camera poses optimized during the reconstruction process were used as the ground truth poses.

C. More Ablation Studies and Analysis

In this section, we show more experimental results and ablation studies to evaluate the effectiveness of different proposed components in our TrafficLoc.

Geometry-guided Attention Loss (GAL). As shown in Table 1, the model incorporating GAL consistently outperforms its counterpart without GAL across all three test splits of the *Carla Intersection* dataset. Notably, when evaluated on images with unseen pitch angles from the **Test**$_{T1-T7hard}$ split, TrafficLoc achieves remarkable **20.4%** improvement in RTE (see the last two rows), highlighting the robustness of GAL to viewpoint variations.

Inter-intra Contrastive Learning (ICL). The first and second rows of Table 1 present the ablation study results comparing ICL and normal contrastive learning (NCL). When using NCL in coarse matching, the model exhibits relatively high error across all three test splits, particularly on **Test**$_{T10}$ which features an unseen world style. Leveraging ICL significantly improves performance, achieving **44.3%** and **56.3%** gains in RRE and RTE on **Test**$_{T10}$, respectively. This enhancement brings the localization accuracy in unseen scenes from an unseen world style to a reasonable level, making reliable localization feasible.

Dense Training Alignment (DTA). The second and fourth rows of Table 1 present the ablation study results of DTA, which facilitates global image supervision by allowing gradients to back-propagate through all image patches via softargmax operation. With the proposed DTA, we observe an improvement of **26.9%** and **14.1%** in RRE and RTE, respectively, on **Test**$_{T10}$.

Fine Matching (*FM*). The fine matching module refines point-to-pixel correspondences within the point group–image patch pairs derived from the coarse matching results. As shown in the fourth and fifth rows of Table 1, the fine matching module further enhances the model's localization accuracy in seen world styles, achieving a **16.8%** improvement in RRE on the **Test**$_{T1-T7}$ split.

More analysis of GAL. The ablation results for the Geometry-guided Attention Loss (GAL) are summarized in Table 2. We conducted experiments on the *Carla Intersection* dataset with GAL using different threshold parameters and applying GAL across different layers of the Geometry-guided Feature Fusion (GFF) module.

When the lower and upper threshold are set to the same value (see the second and third row), the model performs worse than not applying GAL, which highlights the importance of defining a tolerant region that enables the network to flexibly learn attention relationships for intermediate cases between the lower and upper thresholds. With thresholds θ_{low}, θ_{up}, d_{low} and d_{up} set to 10°, 20°, 3m and 5m, our model consistently outperforms the baseline without GAL across all metrics. Moreover, we observed that applying GAL to either the first layer or all layers of the GFF module yields worse localization results compared to applying it only to the last layer. This is mostly because such configurations constrain the network's ability to capture global features during the early stages (or initial layers) of multimodal feature fusion.

More analysis of ICL, DTA and GAL. To further verify the effectiveness of our Inter-intra Contrastive Learning (ICL), Dense Training Alignment (DTA) and Geometry-guided Attention Loss (GAL) in enhancing previous Image-to-Point Cloud (I2P) registration methods, we integrated them into the state-of-the-art CoFiI2P network [8]. We conducted experiments on the KITTI Odometry dataset [5] to assess the improvements. The experimental results shown in Table 3 demonstrate that each module contributes positively to the improvement of localization accuracy. When all three modules are employed together, we achieve the best performance, with an improvement of **25.4%** in RRE and **27.6%** in RTE.

Feature extraction backbone. Table 4 illustrates the results under different image and point cloud feature extraction backbone. Our model performs best when using

	$\theta_{low}(°)$	$\theta_{up}(°)$	$d_{low}(m)$	$d_{up}(m)$	Layer	**Test**$_{T1-T7}$		**Test**$_{T1-T7hard}$		**Test**$_{T10}$	
						RRE(°)	RTE(m)	RRE(°)	RTE(m)	RRE(°)	RTE(m)
Baseline	/	/	/	/	/	0.84	0.62	3.17	1.42	2.98	2.83
	10	10	3	5	Last	1.24	0.80	3.49	1.53	6.07	7.45
	10	20	3	3	Last	1.27	0.83	3.05	1.46	3.55	2.95
	20	30	3	5	Last	0.91	0.59	2.71	1.27	3.05	2.78
	10	20	5	7	Last	0.85	0.55	2.63	1.15	3.08	2.75
	10	20	3	5	First	1.00	0.59	2.68	1.14	4.30	3.23
	10	20	3	5	All	1.02	0.62	3.01	1.19	3.45	3.33
Ours	10	20	3	5	Last	**0.66**	**0.51**	**2.64**	**1.13**	**2.53**	**2.69**

Table 2. Ablation Study on Geometry-guided Attention Loss (GAL). θ_{low} and θ_{up} denote the angular threshold for image-to-point cloud (**I2P**) attention, while d_{low} and d_{up} represent the distance threshold for point cloud-to-image (**P2I**) attention. "Layer" specifies the fusion layer within the Geometry-guided Feature Fusion (GFF) module where GAL is applied.

Base model	ICL	DTA	GAL	RRE(°)	RTE(m)
CoFiI2P	×	×	×	1.14	0.29
CoFiI2P	✓	×	×	1.11	0.26
CoFiI2P	×	✓	×	0.94	0.24
CoFiI2P	×	×	✓	1.01	0.27
CoFiI2P	✓	✓	×	0.94	0.22
CoFiI2P	✓	×	✓	1.04	0.25
CoFiI2P	×	✓	✓	**0.85**	0.22
CoFiI2P	✓	✓	✓	**0.85**	**0.21**

Table 3. Experimental results on KITTI Odometry dataset [5] based on current SOTA model CoFiI2P [8]. "ICL", "DTA" and "GAL" mean whether we add the proposed Inter-Intra Contrastive Loss, Dense Training Alignment and Geometry-guided Attention Loss into CoFiI2P, respectively. We report the mean RRE, mean RTE, and RR metrics for comparison.

	Img Encoder	PC Encoder	RRE(°)	RTE(m)
Baseline	ResNet [6]	PiMAE [2]	1.25	0.87
	ResNet [6]	PT [19]	0.85	0.58
	DUSt3R [17]	PiMAE [2]	1.03	0.75
	DUSt3R* [17]	PT [19]	0.77	0.59
Ours	DUSt3R [17]	PT [19]	**0.66**	**0.51**

Table 4. Ablation study on feature extraction backbone. We report median RRE and median RTE results on test split **Test**$_{T1-T7}$. DUSt3R* means using frozen DUSt3R backbone during training.

DUSt3R [17] and Point Transformer [19] as backbones, benefiting from DUSt3R's strong generalization ability. Even with a frozen DUSt3R, the model achieves comparable performance. In contrast, when using ResNet [6] or PiMAE [2], the model's performance declines due to the lack of attentive feature aggregation during the feature extraction stage. When utilizing PiMAE, we load the pretrained weights of its point encoder.

Localization with unknown intrinsic parameters. Ablation results of localization with predicted intrinsic parameters are shown in Table 5. In the absence of ground-truth intrinsic parameters during inference, we leverage DUSt3R [17] to predict the focal length of the images. The camera is assumed to follow a simple pinhole camera model, with the principle point fixed at the center of the image. When using predicted intrinsic parameters instead of ground-truth focal length, the localization accuracy shows a significant decline. However, enabling focal length refinement during EPnP-RANSAC [4, 10] yields notable improvement on **Test**$_{T1-T7}$, while maintaining similar performance on other two test splits. This suggests that refining predicted focal length during pose estimation is more effective when the correspondences are of higher quality.

Block number of Fusion transformer. Table 6 shows the experimental results of using different numbers of feature fusion layers N_C in Geometry-guided Feature Fusion (GFF) module. Our model achieves the best performance when utilizing a four-layer structure.

Input point cloud size. We conducted ablation studies to investigate the effect of input point cloud size on the representation learning process. The number of coarse point groups was fixed to $M = 512$, as these groups were generated using Farthest Point Sampling (FPS), ensuring uniform sampling across the point cloud. As shown in Table 7, the localization accuracy decreases with lower point cloud densities, as overly sparse point cloud lose local critical structural details. On the other hand, higher-density point clouds place a heavy computational burden. To balance computational efficiency and accuracy, we selected an input size of 20,480 points.

D. Carla Intersection Dataset

Our proposed *Carla Intersection* dataset consists of 75 intersections across 8 worlds ($Town01$ to $Town07$ and $Town10$) within the CARLA [3] simulation environment, encompassing both urban and rural landscapes. $Town01$ to $Town07$ include multiple intersections for training and testing, while $Town10$ contains only one intersection for testing. Specifically, we utilize the first Intersection scenario from each world (e.g. $Town01$ $Int1$, $Town02$ $Int1$, $\ldots$, $Town07$ $Int1$, $Town10$ $Int1$) for testing, with all re-

	GT Focal	Refine Focal	**Test**$_{T1-T7}$		**Test**$_{T1-T7hard}$		**Test**$_{T10}$	
			RRE(°)	RTE(m)	RRE(°)	RTE(m)	RRE(°)	RTE(m)
Ours	×	×	2.04	1.72	4.56	2.19	4.61	4.80
	×	✓	0.95	0.80	3.74	2.36	3.88	5.06
	✓	×	**0.66**	**0.51**	**2.64**	**1.13**	**2.53**	**2.69**

Table 5. Ablation study on localization with intrinsic parameters predicted by DUSt3R [17]. We report the median RRE and median RTE across all three test splits of the *Carla Intersection* dataset. "GT Focal" refers to using the ground-truth focal length during inference, and "Refine Focal" enables focal length optimization as part of the EPnP-RANSAC [4, 10] process.

	N_c	RRE(°)	RTE(m)
Baseline	2	0.96	0.55
	6	0.73	0.59
	8	0.88	0.58
Ours	4	**0.66**	**0.51**

Table 6. Ablation study on the number of feature fusion layers N_c in Geometry-guided Feature Fusion (GFF) module. We report median RRE and median RTE on test split **Test**$_{T1-T7}$ of *Carla Intersection* dataset.

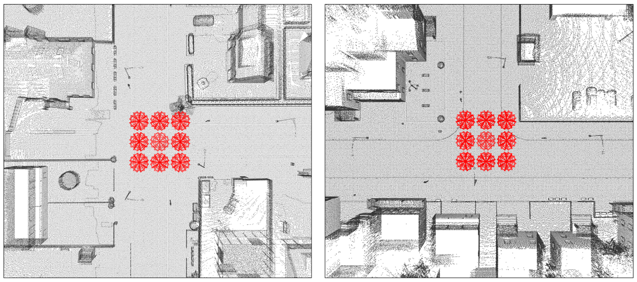
Point Number	RRE(°)	RTE(m)	FLOPs
5120	0.86	0.68	126.73G
10240	0.81	0.62	146.38G
20480	0.66	**0.51**	185.73G
40960	**0.59**	0.52	264.35G

Table 7. Ablation study on the input point cloud size. We report median RRE and median RTE on test split **Test**$_{T1-T7}$ of *Carla Intersection* dataset. The FLOPs is calculated during the inference process.

maining intersections reserved for training.

Images. For each intersection, we captured 768 training images and 288 testing images with known ground-truth 6-DoF pose at a resolution of 1920x1080 pixel and a horizontal field of view (FOV) of 90°, equals to a focal length of 960. To generate these images, we sampled camera positions in a grid-like pattern with different heights at the center of each intersection. For each position, we captured images at 8 yaw angles (spaced at 45° intervals) and 2 pitch angles. Figure 3 shows the sampled poses for example intersections.

Table 8 summarizes the image data collection details for our *Carla Intersection* dataset. All training images were captured with downward pitch angles of 15° and 30° at heights of 6m, 7m, and 8m. Testing images in the test splits **Test**$_{T1-T7}$ and **Test**$_{T10}$ share the same pitch angles as the training images, but were captured at heights of 6.5m and 7.5m. Additionally, for the test split **Test**$_{T1-T7hard}$, we captured 288 additional testing images for each intersection using the same positions as in **Test**$_{T1-T7}$, but with different pitch angles of 20° and 25°, at heights of 6.5m and 7.5m. These data capture settings closely reflect the real-world traffic surveillance camera installations following HIKVISION [7], ensuring typical positioning to provide optimal traffic views under varied monitoring conditions. The differences between three distinct test splits also allow us to evaluate the model's generalization ability across unseen intersections and unseen world styles. Note that all testing intersections were not seen during the training.


(a) Town01 Int1

(b) Town02 Int1

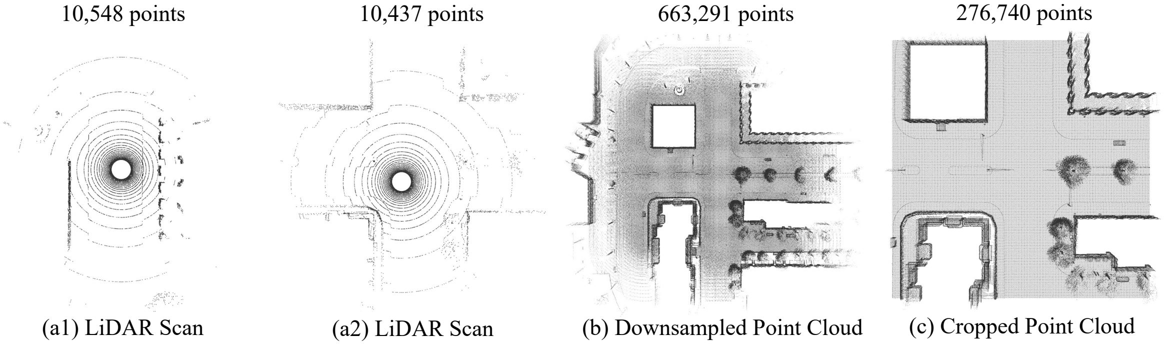
Figure 3. Sampled testing image poses of (a) Town01 Intersection1 and (b) Town02 Intersection1.

Point Clouds. To capture the point cloud of each intersection, we utilize a simulated LiDAR sensor in the CARLA [3] environment, which emulates a rotating LiDAR using ray-casting. The LiDAR operates at a rotation frequency of 10 frames per second (FPS), with a vertical field of view (FOV) ranging from 10° (upper) to -30° (lower). The sensor generates 224,000 points per second across all lasers. Other parameters of the simulated LiDAR follow the default configuration in CARLA Simulator. As shown in Figure 4, the LiDAR scans were captured in an on-board manner. Then, we accumulated all scans into a single point cloud and downsampled it with a resolution of 0.2m. Finally, the point cloud for each intersection was cropped to a region measuring 100m×100m×50m, focusing on the area of interest for our study.

During the data capturing process, we disabled dynamic weather variations and set the weather condition in CARLA simulation environment to the default weather parameters of world $Town10$. Some examples of our *Carla Intersection* dataset are shown in Figure 6. Our data collection codes and datasets will be publicly available upon acceptance.

	Training	**Test**$_{T1-T7}$	**Test**$_{T1-T7hard}$	**Test**$_{T10}$
worlds	$Town$01-07	$Town$01-07	$Town$01-07	$Town$10
# intersections	67	7	7	1
# images per scene	768	288	288	288
height (m)	6 / 7 / 8	6.5 / 7.5	6.5 / 7.5	6.5 / 7.5
pitch (°)	15 / 30	15 / 30	20 / 25	15 / 30
seen intersection	–	×	×	×
seen world	–	✓	✓	×

Table 8. Image data collection details of the proposed *Carla Intersection* dataset. "# intersections" means the number of intersection scenes in each split dataset and "# images per scene" means the number of images in each intersection scene. "Seen intersection" and "seen world" represent whether the testing intersections are seen and whether the testing intersections are from the seen world during the training process, respectively.


10,548 points

10,437 points

663,291 points

276,740 points

(a1) LiDAR Scan

(a2) LiDAR Scan

(b) Downsampled Point Cloud

(c) Cropped Point Cloud

Figure 4. Point cloud capturing example from $Town10$ $Int1$. (a1) and (a2) depict the LiDAR scan from a single frame. (b) shows the aggregated and downsampled point cloud. (c) presents the final cropped point cloud with dimensions of 100m×100m×100m.

E. Geometry-guided Feature Fusion

Our Geometry-guided Feature Fusion (GFF) module comprises of N_c transformer-based fusion blocks, each consisting of a self-attention layer followed by a cross-attention layer.

Given the image feature $\mathbf{F}_I$ and point cloud feature $\mathbf{F}_P$, both enriched with positional embeddings, the self-attention layer enhances features within each modality individually using standard multi-head scalar dot-product attention:

$$\dot{\mathbf{F}} = \mathbf{Q} + \mathrm{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}), \tag{1}$$

where $\mathbf{Q} = \mathbf{K} = \mathbf{V} = \mathbf{F} \in \mathbb{R}^{N_t \times C}$ denotes the *Query, Key* and *Value* matrices, and $\mathbf{F}$ represents either $\mathbf{F}_I$ or $\mathbf{F}_P$ depending on the modality. Within the MHA layer, the attention operation is conducted by projecting $\mathbf{Q}, \mathbf{K}$ and $\mathbf{V}$ using h heads:

$$\begin{aligned} \mathrm{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= [head_1, \dots, head_h]\mathbf{W}^O \\ head_i &= \mathrm{Attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \end{aligned} \tag{2}$$

where $\mathbf{W}_i^{Q,K,V,O}$ denote the learnable parameters of linear projection matrices and the $\mathrm{Attention}$ operation is defined as:

$$\mathrm{Attention}(q, k, v) = \mathrm{Softmax}(\frac{q \cdot k^\top}{\sqrt{d_k}})v, \tag{3}$$

where d_k is the dimension of latent feature.

The cross-attention layer fuses image and point cloud features by applying the attention mechanism across modalities, following the same formulation as Equation 1. However, the *Query, Key* and *Value* matrices differs based on the direction of attention. Specifically, for **I2P** (Image-to-Point Cloud) attention, we use $\mathbf{Q} = \mathbf{F}_I$ and $\mathbf{K} = \mathbf{V} = \mathbf{F}_P$, while for **P2I** (Point Cloud-to-Image) attention, we set $\mathbf{Q} = \mathbf{F}_P$ and $\mathbf{K} = \mathbf{V} = \mathbf{F}_I$.

Layer Normalization is applied to ensure stable training. For our GFF module, we set $N_c = 4$ and $h = 4$. Both the input channel C and the latent dimension d_k are set to 256.

F. Implementation Details

In *Carla Intersection* dataset, each intersection point cloud represents a region of 100m×100m×50m and contains over 200,000 points. Following [15], as a preprocessing step, we first divide each intersection point cloud into several 50m×50m×50m voxels with a stride of $25m$. For each voxel V_i, we assign an associated set of images $\{I_i\}$ based on the overlap ratio between the image frustum and the voxel. Specifically, a voxel V_i is associated with an image I_i if more than 30% projected points lie within the image plane. During each training epoch, we uniformly sample B images for each voxel from its associated image set, result-

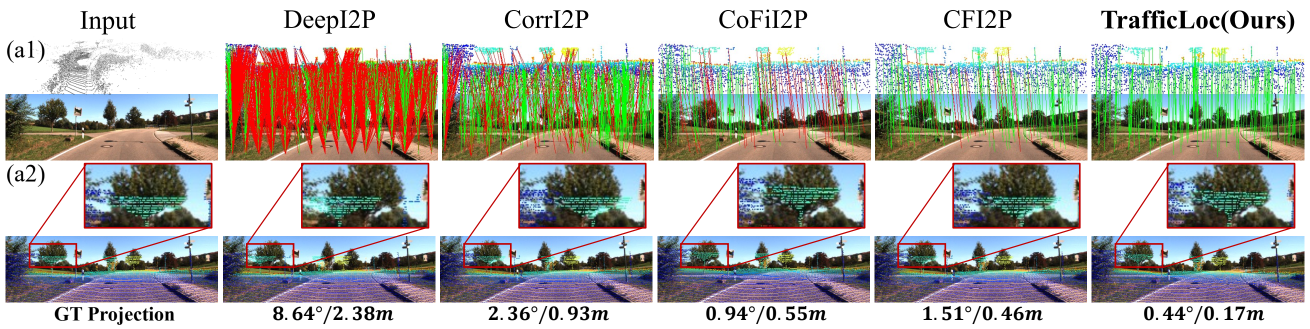


Figure 5. Qualitative results of our TrafficLoc and other baseline methods on the KITTI Odometry dataset [5]. (a1) shows predicted correspondences and (a2) visualizes the point cloud projected onto the image plane. The first column provides the input point cloud, the input image and the ground-truth projection for reference.

ing in $B \cdot N_v$ training image-point cloud pairs, where N_v denotes the total number of voxels.

The input images are resized to 288 × 512, and the input point cloud size is $N = 20480$ points. We utilize a pre-trained Vision Transformer from *DUSt3R_ViT Large* [17] to extract the image feature. For coarse matching, we use a resolution of 1/16 of the input resolution for image ($s = 16$) and set the number of point group $M = 512$, with a coarse feature channel size of $C = 256$. For fine matching, we adopt a resolution of $(H/2 \times W/2 \times C')$ for fine image feature and $(N \times C')$ for fine point feature, where H, W and N equal to input dimensions and the fine feature channel size is set to $C' = 64$. As part of data augmentation, we apply random center cropping to the input images before resizing operation to simulate images captured by different focal lengths. The input point cloud is first normalized into a unit cube, followed by random rotations around the z-axis (up to 360°) and random shifts along the xy-plane (up to 0.1m).

The whole network is trained for 25 epochs with a batch size of 8 using the Adam optimizer [9]. The initial learning rate is set to 0.0005 and is multiplied by 0.5 after every 5 epochs. For the joint loss function, we set $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1$. The safe radius r, positive margin m_p, negative margin m_p and scale factor γ in loss function are set to 1, 0.2, 1.8 and 10, respectively. For the Geometry-guided Attention Loss (GAL), the angular thresholds θ_{low} and θ_{up} are set to 10° and 20°, while the distance thresholds d_{low} and d_{up} are set to 3m and 5m, respectively. The training is conducted on a single NVIDIA RTX 6000 GPU and takes approximately 40 hours.

During inference, we utilize the super-point filter to select reliable in-frustum point groups from the fused coarse point features $\mathbf{F}_P^{coarse}$, using a confidence threshold of 0.9. In the **coarse matching stage**, we compute the coarse similarity map between each point group and the image. Following [18], a **window soft-argmax** operation is employed on similarity map to estimate the corresponding coarse pixel position. This involves first identifying the target center with an argmax operation, followed by a soft-argmax within a predefined window (window size set to 5). In the **fine matching stage**, with the predicted coarse pixel position, we first extract a fine local patch feature of size $w \times w$ from the fine image feature and select the fine point feature of each point group center, and then compute the fine similarity map between each point group center and the extracted local patch. Since the extracted local fine image patch has a relative small size ($w = 8$), a soft-argmax operation is applied over the **entire** fine similarity map to determine the final corresponding 2D pixel for each 3D point group center. Finally, we estimate the camera pose using EPnP-RANSAC [4, 10] based on the predicted 2D-3D correspondences. For cases where one single image is associated with multiple point clouds, an additional EPnP-RANSAC step is performed using all inliers from each image-point cloud pair to compute the final camera pose.

For experiments on the KITTI Odometry [5] and Nuscenes [1] datasets, we ensure a fair comparison by adopting the same procedures as in previous works [8, 11, 12] to generate image-point cloud pairs.

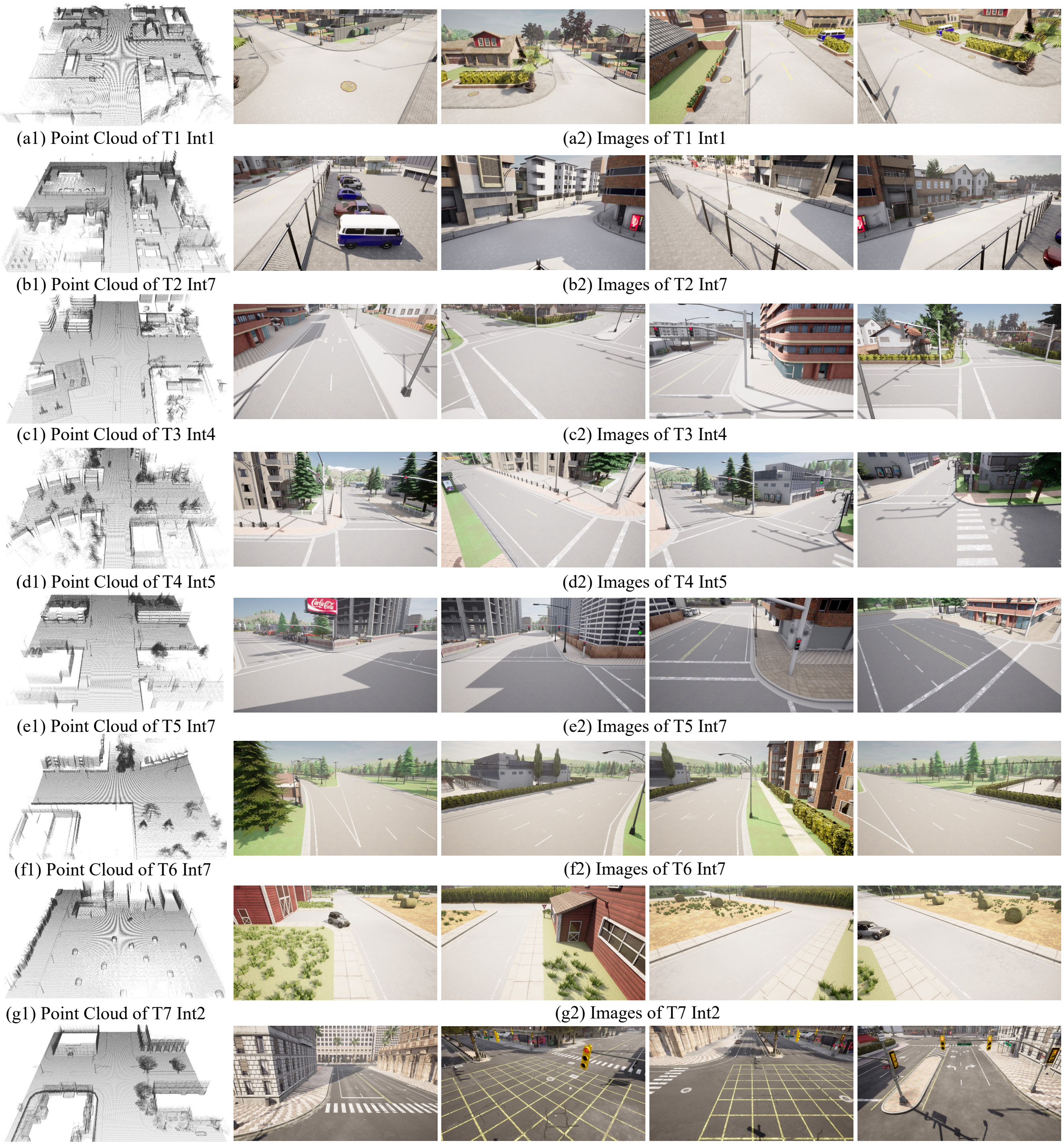
In the KITTI Odometry dataset [5], there are 11 sequences with ground-truth camera calibration parameters. Sequences 0-8 are used for training, while sequences 9-10 are reserved for testing. Each image-point cloud pair was selected from the same data frame, meaning the data was captured simultaneously using a 2D camera and a 3D LiDAR with fixed relative positions. During training, the image resolution was set to 160×512 pixels, and the number of points was fixed at 20480. The model was trained with a batch size of 8 until convergence. The initial learning rate is set to 0.001 and is multiplied by 0.5 after every 5 epochs.

For the NuScenes dataset [1], we utilized the official SDK to extract image-point cloud pairs, with the point clouds being accumulated from the nearby frames. The dataset includes 1000 scenes, of which 850 scenes were used for training and 150 for testing, following the official data split. The image resolution was set to 160×320 pixels, and the number of points was fixed at 20480.

G. More Visualization Results

In this section, we present additional examples of localization results. Figure 5 and Figure 7 compare the localization performance of TrafficLoc with other baseline methods on the KITTI Odometry dataset [5] and all three test splits of the *Carla Intersection* dataset, respectively. Our TrafficLoc predicts a higher number of correct point-to-pixel correspondences, and the point cloud projected with the predicted pose exhibits greater overlap with the image, demonstrating superior performance.

We also include two failure cases in Fig. 2 for limitation analysis. Fig. 2 (b) is from our synthetic *Carla Intersection* dataset. Although the model successfully matches features in distinctive regions (e.g., electric pole and traffic cones), the large **low-texture** grassy area in the image leads to inaccurate pose estimation. Note that this scene was unseen during the training. The second case comes from the OpenTrafficCam dataset [16] (Fig. 2 (c)). Due to the sparsity of the COLMAP-reconstructed point cloud in this area and the resulting **modality gap** compared to LiDAR scans, the model fails to extract expressive 3D features for reliable matching.



(a1) Point Cloud of T1 Int1

(a2) Images of T1 Int1

(b1) Point Cloud of T2 Int7

(b2) Images of T2 Int7

(c1) Point Cloud of T3 Int4

(c2) Images of T3 Int4

(d1) Point Cloud of T4 Int5

(d2) Images of T4 Int5

(e1) Point Cloud of T5 Int7

(e2) Images of T5 Int7

(f1) Point Cloud of T6 Int7

(f2) Images of T6 Int7

(g1) Point Cloud of T7 Int2

(g2) Images of T7 Int2

(h1) Point Cloud of T10 Int1

(h2) Images of T10 Int1

Figure 6. Example point clouds and images data of our *Carla Intersection* dataset. T1 means $Town01$ and Int1 means $Intersection1$. Since all instances of the $Intersection1$ scenario across different worlds are included in the test set, we focus on showcasing their testing images (e.g. T1 Int1 and T10 Int1). For other intersections, we present the training images instead.

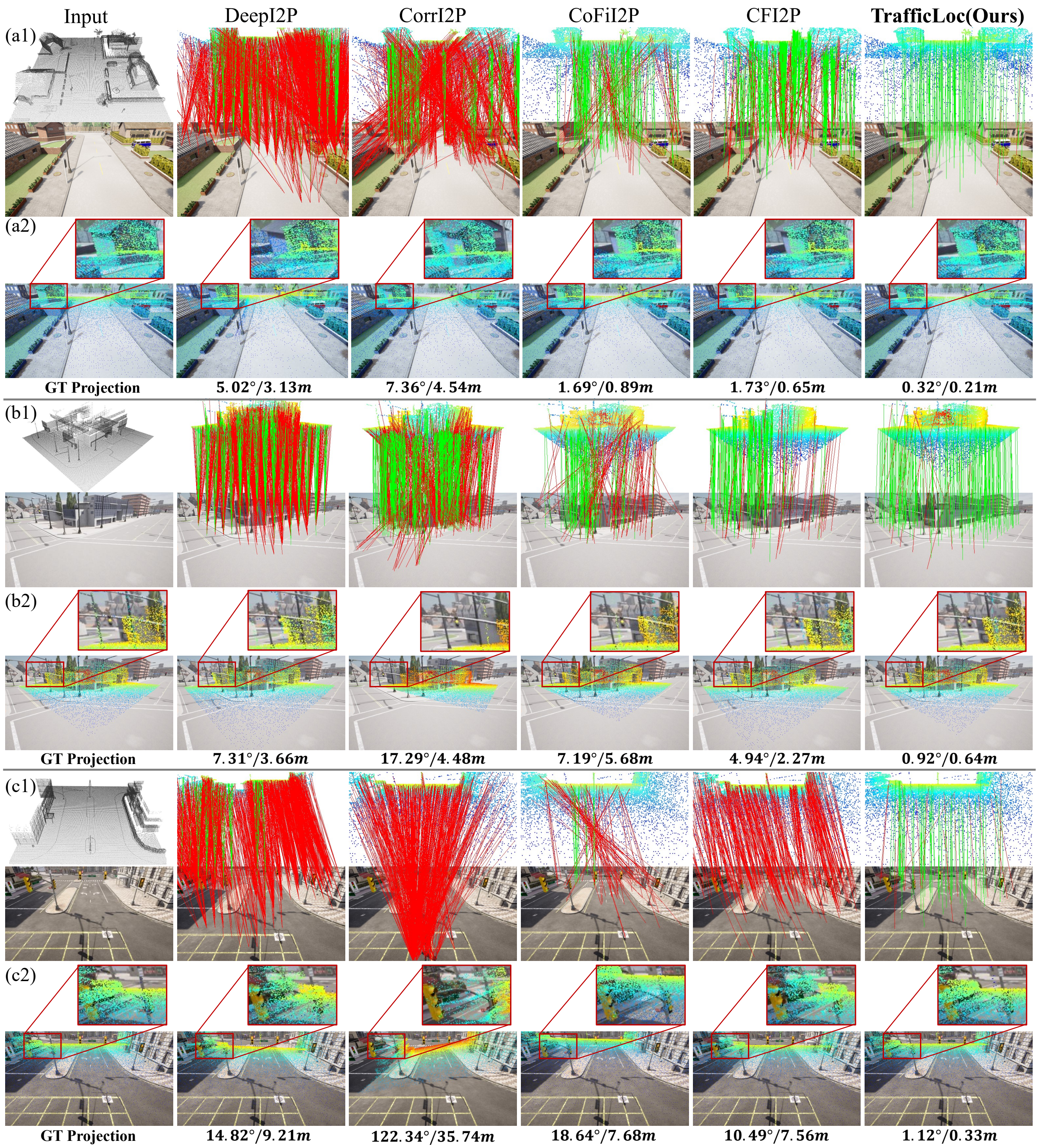


Figure 7. Qualitative results of our TrafficLoc and other baseline methods on the *Carla Intersection* dataset. The point cloud is projected onto a 2D view and displayed above the image, with point colors indicating distance. The proposed TrafficLoc achieves superior performance, with more correct (green) and fewer incorrect (red) point-to-pixel pairs. (a1) shows predicted correspondences on **Test**$_{T1-T7}$ and (a2) visualizes the point cloud projected onto the image plane. Similarly, (b1) and (b2) show results on **Test**$_{T1-T7hard}$, (c1) and (c2) show results on **Test**$_{T10}$. The first column provides the input point cloud, the input image and the ground-truth projection for reference.

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

[2] Anthony Chen, Kevin Zhang, Renrui Zhang, Zihan Wang, Yuheng Lu, Yandong Guo, and Shanghang Zhang. Pimae: Point cloud and image interactive masked autoencoders for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5291–5301, 2023. 3

[3] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio López, and Vladlen Koltun. Carla: An open urban driving simulator. *Conference on Robot Learning,Conference on Robot Learning*, 2017. 3, 4

[4] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 3, 4, 6

[5] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 2, 3, 6, 7

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[7] HIKVISION. Anpr product functions and problem troubleshooting. `https://www.securitywholesalers.com.au/files/ANPRINSTALLATION1.pdf`, 2016–2024. 4

[8] Shuhao Kang, Youqi Liao, Jianping Li, Fuxun Liang, Yuhao Li, Xianghong Zou, Fangning Li, Xieyuanli Chen, Zhen Dong, and Bisheng Yang. Cofii2p: Coarse-to-fine correspondences for image-to-point cloud registration. *arXiv preprint arXiv:2309.14660*, 2023. 2, 3, 6

[9] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 6

[10] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision (IJCV)*, 81(2):155–166, 2009. 3, 4, 6

[11] Jiaxin Li and Gim Hee Lee. DeepI2P: Image-to-Point Cloud Registration via Deep Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15960–15969, 2021. 6

[12] Siyu Ren, Yiming Zeng, Junhui Hou, and Xiaodong Chen. Corri2p: Deep image-to-point cloud registration via dense correspondence. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(3):1198–1208, 2022. 6

[13] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[14] Yu Sheng, Lu Zhang, Xingchen Li, Yifan Duan, Yanyong Zhang, Yu Zhang, and Jianmin Ji. Rendering-enhanced automatic image-to-point cloud registration for roadside scenes. *arXiv preprint arXiv:2404.05164*, 2024. 1

[15] Shitao Tang, Sicong Tang, Andrea Tagliasacchi, Ping Tan, and Yasutaka Furukawa. Neumap: Neural coordinate mapping by auto-transdecoder for camera localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 929–939, 2023. 5

[16] Khiem Vuong, Robert Tamburo, and Srinivasa G. Narasimhan. Toward planet-wide traffic camera calibration. In *IEEE Winter Conference on Applications of Computer Vision*, 2024. 1, 7

[17] 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. 3, 4, 6

[18] Junyi Zhang, Charles Herrmann, Junhwa Hur, Eric Chen, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. Telling left from right: Identifying geometry-aware semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3076–3085, 2024. 6

[19] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16259–16268, 2021. 3