A Details on Dataset Collection

The released datasets consist of two distinct parts, UAVD4L-LoD and Swiss-EPFL, providing LoD3.0 and LoD2.0 models, respectively. The UAVD4L-LoD dataset, which spans an area of 2.5 square kilometers, is derived from a semi-automatic process that generates a 3D LoD map based on the mesh model of the UAVD4L [72] dataset. This dataset includes a diverse array of architectural structures, including skyscrapers, villas, apartment complexes, educational institutions, and rural dwellings. The query images for this dataset were captured using two UAVs equipped with real sensor data: a DJI M300 [3] drone with an H20T [2] camera and a DJI Mavic3 Pro [5] drone. The Swiss-EPFL dataset, which covers an expansive area of 8.18 square kilometers, sources its LoD2.0 models from data made publicly accessible by the Swiss federal authorities [9–11]. This dataset features a variety of architectural styles, such as libraries, residential apartments, and stadiums. The query images for this dataset were acquired through the CrossLoc [74] projects, using a DJI Phantom 4 RTK [6] drone. Figure 2 presents the 3D LoD maps and query images from these two datasets.

A.1 3D LoD Map Collection

The 3D LoD map for the UAVD4L-LoD dataset is generated semi-automatically with the assistance of the DP Modeler tool[4]. The process begins with the automatic generation of building blocks, characterized by their footprints and heights. Manual refinement is then applied to the architectural details of each building, raising them to the LoD3.0 level. The LoD accuracy of the UAVD4L-LoD dataset is consistent with the mesh model derived from UAVD4L.

For the Swiss-EPFL dataset, LoD2.0 models are downloaded from the SwissTOPO website [9–11]. To synchronize the coordinate systems between the map data and the drone-captured data from the CrossLoc dataset (which covers the same area with ground truth pose annotation), we converted the Swiss LoD map data in LV95 and LN02 coordinate systems to the ECEF coordinate system.

A.2 Query Image Collection

The query images of the UAVD4L-LoD dataset are divided into two categories: *in-Traj.* and *out-of-Traj.*, representing trajectory-based and free-flight scenarios, respectively. The *in-Traj.* images, totaling $1,604$, were captured using a DJI M300 drone equipped with an H20T camera, focusing primarily on residential buildings, villas, and educational institutions. In contrast, the *out-Traj.* images, totaling $2,192$, were captured using a DJI Mavic3 Pro drone, covering a variety of architectural structures such as skyscrapers and rural dwellings. Both the *in-Traj.* and *out-of-Traj.* datasets include real sensor priors. Table 5 outlines the specific differences between the *in-Traj.* and *out-Traj.* sequences.

Table 5: **Key distinctions between the *in-Traj.* and *out-of-Traj.* sequences**.

Name	Capture device	Capture pitch angle	Capture height	Capture route
in-Traj.	DJI M300+H20t	0° or 45°	120m	Zig-zag flight on a selected region
out-of-Traj.	DJI Mavic3 Pro	$30° \sim 60°$	$90\text{m} \sim 150\text{m}$	Manually controlled flight on the map

The real query images in the Swiss-EPFL dataset come from the CrossLoc [74] dataset. However, because the real-time kinematics (RTK) data from the DJI Phantom4 were used directly as ground truth (GT) poses, some GT poses show significant mislabeling. To resolve this issue, we projected the wireframes of LoD maps onto query images to identify and remove incorrectly labeled poses. The final query dataset comprises $2,254$ images.

A.3 Query GT Generation

For the UAVD4L-LoD dataset, we employ a semi-automatic annotation approach to generate pseudo-GT poses $\{\overline{\boldsymbol{\xi}}_i\}$ for the query images $\{\mathbf{I}_i^q\}$. The process is based on the SfM results and textured mesh model of the UAVD4L [72]. First, we perform SfM separately on the query images $\{\mathbf{I}_i^q\}$ and the reference images $\{\mathbf{I}_i^r\}$ from the UAVD4L to obtain SfM results $\mathcal{C}_q$ and $\mathcal{C}_r$. Next, we manually

in-Traj. *out-of-Traj.*

Figure 6: **Flight trajectories of query images in the UAVD4L-LoD dataset**. We present the flight trajectories of the registered $in\text{-}Traj.$ and $out\text{-}of\text{-}Traj.$ query images. The $in\text{-}Traj.$ images follow a predetermined flight path, primarily covering the left half of the map. In contrast, the $out\text{-}of\text{-}Traj.$ images navigate arbitrarily without a fixed route, randomly covering the entire map.

select points with distinctive visual features (e.g., building corners) as tie points to align $\mathcal{C}_q$ with $\mathcal{C}_r$. To further enhance the accuracy of the pseudo-GT, we utilize the render-and-compare pipeline [76] to refine the final poses $\{\overline{\boldsymbol{\xi}}_i\}$.

Additionally, we analyze the discrepancies between the pose prior and the GT pose, decoupling the poses into 3D translation in WGS84 space and Euler angles in terms of 'yaw-pitch-roll'. It is observed that the translation errors for x and y are within ± 10, z errors are within ± 30, yaw errors are within ± 7.5, and pitch and roll errors are approximately 1 degree.

B Details on UAVD4L-LoD Dataset

B.1 Pseudo-GT Generation

In the UAVD4L-LoD dataset, we employed a semi-automatic annotation technique to generate pseudo-GT poses $\{\overline{\xi}_i\}$ for the query images $\{\mathbf{I}_i\}$. Initially, we performed SfM separately on the query images $\{\mathbf{I}_i\}$ and the reference images $\{\mathbf{I}_j^r\}$ from UAVD4L, yielding corresponding SfM results $\mathcal{C}_q$ and $\mathcal{C}_r$. Subsequently, based on the capture region of the $\{\mathbf{I}_i\}$, we manually identified e distinctive tie points, such as the corner of the building, to align $\mathcal{C}_q$ with $\mathcal{C}_r$, resulting in $\mathcal{C}_f$. We then refined the pose accuracy of $\mathcal{C}_f$ using Bundle Adjustment. The accuracy of the GT poses was evaluated through the median reprojection error, which was 0.43 pixels for all connected points and 1.19 pixels for the tie points. Finally, we employed a render-and-compare [76] pipeline for the final refinement of the GT poses. In this manner, with the annotation of tens of $e = 20$ manual tie points, we were able to obtain pseudo-GT poses $\{\overline{\xi}_i\}$ for a total of $3,796$ query images $\{\mathbf{I}_i\}$. Figure 6 shows the flight trajectories of the $in\text{-}Traj.$ and $out\text{-}of\text{-}Traj.$

B.2 Sensor Pose Accuracy

In the UAVD4L-LoD dataset, we conduct a comprehensive data analysis to validate the precision of the sensor pose. This is accomplished by employing absolute error bar charts, as illustrated in Figure 7. Additionally, we assess the accuracy by projecting wireframe points onto the image plane using both sensor and GT poses. Results of these projections are depicted in Figure 8.

C Details on Swiss-EPFL Dataset

C.1 Data Cleaning

In the Swiss-EPFL dataset, the GT poses $\{\overline{\xi}_i\}$ for the query images $\{\mathbf{I}_i\}$ are sourced from the CrossLoc project [74]. This project directly acquires RTK data from the DJI Phantom 4 for GT annotation. Considering that the RTK device may introduce some noise, we identified and excluded query images with incorrect labeling. This was accomplished by projecting the wireframe onto the

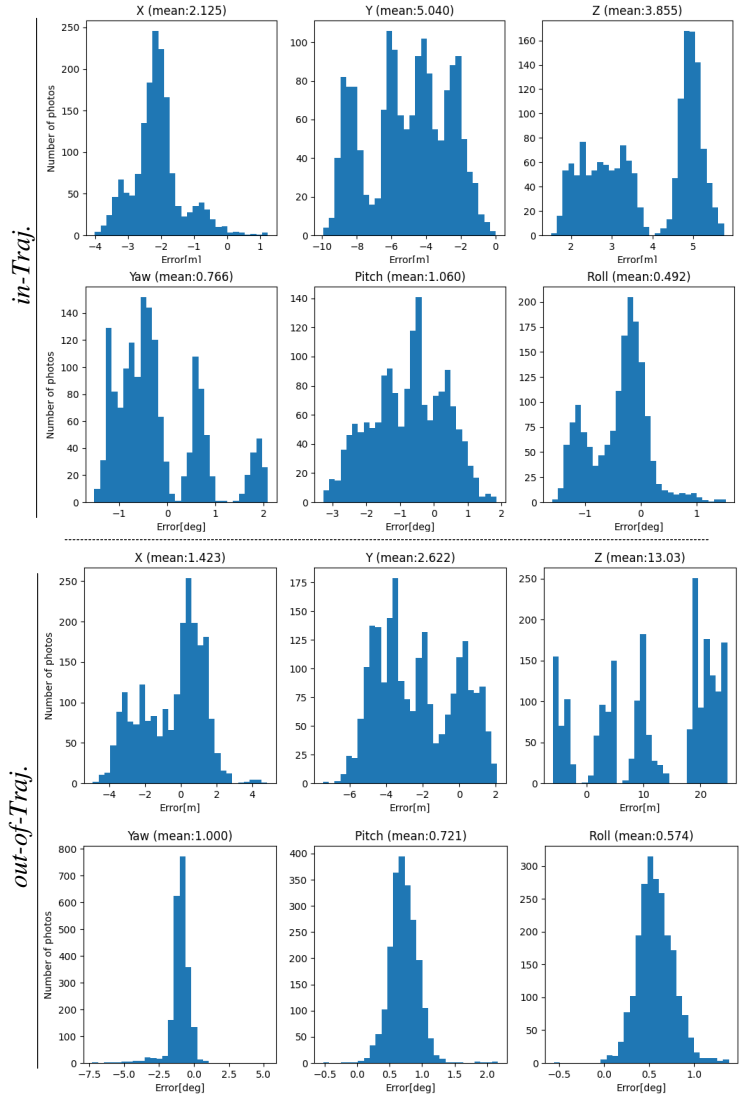

Figure 7: **The errors between priors and GT poses**. We visualize the absolute pose errors between the sensor and GT pose in 6-DoF. The errors in the X-Y-Z-yaw dimensions show indicate substantial discrepancies. Specifically, the errors in the X-Y range from -10 to 10 meters, the Z ranges from -30 to 30 meters, and the yaw fluctuates within the range of -7.5 to 7.5 degrees.

image plane and manually discarding the items exhibiting noticeable misalignment. The process is visualized in Figure 9.

C.2 Sensor Poses Generation

Since the CrossLoc [74] project does not provide GPS or other sensor data, we randomly generate sensor poses $\boldsymbol{\xi}_p$ by emulating the pose errors derived from the UAVD4L-LoD dataset. Specifically, X-Y for position range between $[-10, 10]$ meters, Z ranges between $[-30, 30]$ meters, yaw for rotation ranges in $[-7.5, 7.5]$ degrees, and $pitch$-$roll$ range between $[-1, 1]$ degrees. We present the discrepancy between the generated sensor poses and GT poses in a bar chart, as depicted in Figure 11.

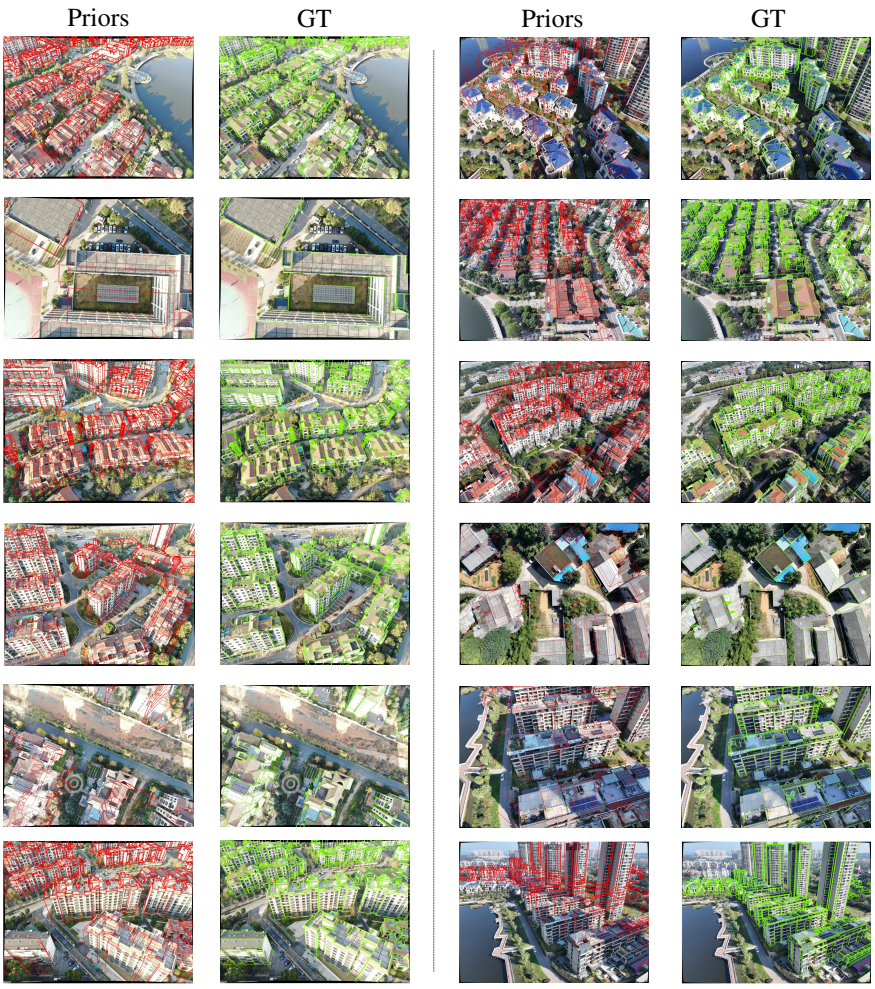

Figure 8: **3D wireframe projection over UAVD4L-Lod dataset.** We visualize the projected wireframe on query images based on sensor and GT poses to demonstrate their accuracy.

D Details on Method

D.1 Architecture of Multi-scale Feature Extractor

In this section, we provide a detailed description of the architecture of the multi-scale feature extractor in Table 6.

D.2 Jacobian Computation

The objective function for pose refinement is:

$$E(\boldsymbol{\xi}^*) = -\sum_i ||f_i||^2 = -\sum_i ||\mathbf{F}_{rf}[\Pi(\mathbf{R}^* \cdot \mathbf{P}_i + \mathbf{t}^*)]||^2. \tag{11}$$

We compute the Jacobian matrix of the residual function f_i with respect to the pose parameters as followed:

$$J_i = \frac{\partial f_i}{\partial \xi^*} = \frac{\partial \mathbf{F}_{rf}}{\partial p_i} \frac{\partial p_i}{\partial \mathbf{P}_i^{cam}} \frac{\partial \mathbf{P}_i^{cam}}{\partial \xi^*}, \tag{12}$$

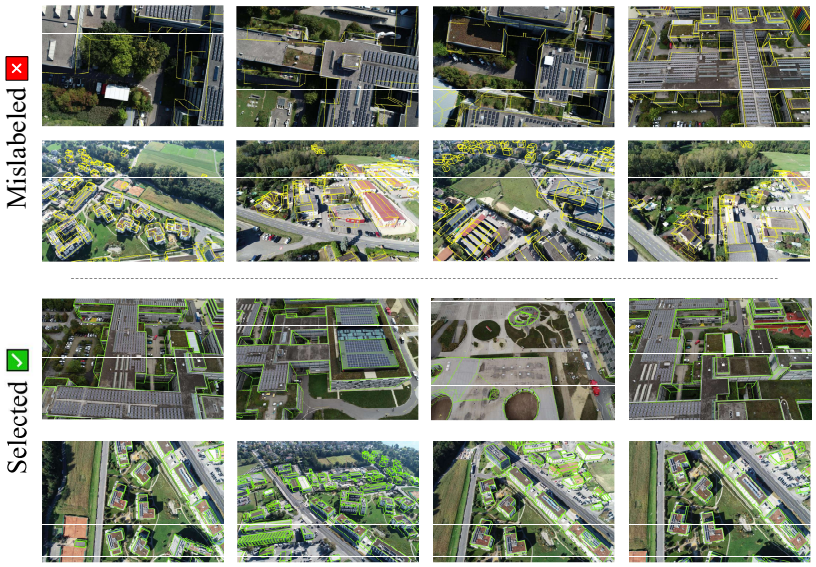

Figure 9: **Samples of mislabeled and selected query images over Swiss-EPFL dataset**. We eliminate mislabeled query images by manually identifying the alignment between the projected 2D wireframe and the corresponding RGB image.

where $\frac{\partial \mathbf{F}_{rf}}{\partial p_i}$ is the gradient of the feature map $\mathbf{F}_{rf}$ at the 2D location p_i and

$$\begin{aligned} p_i =& \Pi(\mathbf{P}_i^{cam}) = \begin{pmatrix} \frac{X_i^{cam}}{Z_i^{cam}} f_x + c_x \\ \frac{Y_i^{cam}}{Z_i^{cam}} f_y + c_y \end{pmatrix}, \\ \frac{\partial p_i}{\partial \mathbf{P}_i^{cam}} =& \begin{pmatrix} \frac{1}{Z_i^{cam}} f_x & 0 & -\frac{X_i^{cam}}{(Z_i^{cam})^2} f_x \\ 0 & \frac{1}{Z_i^{cam}} f_y & -\frac{Y_i^{cam}}{(Z_i^{cam})^2} f_y \end{pmatrix}. \end{aligned} \tag{13}$$

Besides, $\mathbf{P}_i^{cam}$ is the point which transformed to the camera space. To compute the last derivative of Eq. 12, we add a perturbation $\Delta\xi$ to the transformation:

$$\mathbf{P}_i^{cam} = \mathbf{R}^*(\Delta\mathbf{R}\mathbf{P}_i + \Delta\mathbf{t}) + \mathbf{t}^*, \tag{14}$$

Finally, the derivatives w.r.t the translation component and rotation component are:

$$\begin{aligned} \frac{\partial \mathbf{P}_i}{\partial \xi_t^*} =& \frac{\partial \mathbf{P}_i}{\partial \Delta\mathbf{t}} = \mathbf{R}^* \\ \frac{\partial \mathbf{P}_i}{\partial \xi_r^*} =& \frac{\partial \mathbf{P}_i}{\partial \Delta\mathbf{R}} = -\mathbf{R}^*[\mathbf{P}_i]_\times, \end{aligned} \tag{15}$$

where $[]_\times$ is the skew-symmetric matrix.

E Details on Baseline

E.1 Sensor-guided Image Retrieval

For baselines, a retrieval-and-matching process is used upon the reference images in the dataset. To ensure a fair comparison, we apply the sensor poses to guide the image retrieval process for UAVD4L [72] and Cad-Loc [44]. For each query image $\mathbf{I}$, we narrow the retrieval candidates ${}^q\mathcal{I}$ using Eq. 16.

$${}^q\mathcal{I} = \{\mathbf{I}_i^r \mid \forall \ \|\mathbf{t}_i^r - \mathbf{t}^q\| \leq \gamma_t, \arccos(\mathbf{R}_i^r, \mathbf{R}^q) \leq \gamma_o\}, \tag{16}$$

where $\|\cdot\|$ denotes the Frobenius Norm between two translation matrices, $\arccos(\cdot)$ calculates the rotation angles between two matrices, γ_t and γ_o are the threshold for translation and orientation, respectively. To determine the proper values for γ_t and γ_o for the baseline methods, a series of

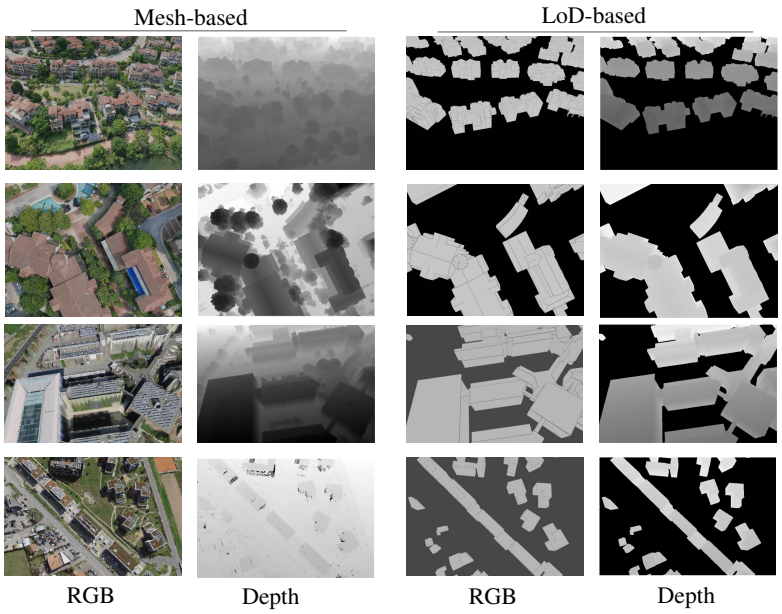

Figure 10: **Visualization of reference RGB and depth maps.** RGB and depth maps are rendered using a textured mesh model or a 3D LoD map.

experiments are conducted on the UAVD4L dataset. In these experiments, we hypothesize that if no reference image could be located within the defined search area, the sensor pose would be utilized as the localization result. Table 7 shows that stricter thresholds result in worse outcomes. Consequently, we set $\gamma_t = 150$ and $\gamma_o = 30$ to ensure a sufficient search space. Furthermore, we measure the impact of retrieval number k in Table 8. The results suggest that while a larger k value enhances the performance of the benchmark algorithm, it also leads to an increase in inference time. Following previous work [72], the retrieval number is set at $k = 3$. It is worth noting that regardless of the choice of k, our method exhibits a substantial acceleration in speed, outperforming by several-fold, or even an order of magnitude.

E.2 Reference Image Details

We provide a detailed description of the reference images used in the two datasets. These images serve dual purposes: they function as the database images for retrieval and matching in baselines, and they are also utilized as training data for the proposed LoD-Loc method. Specifically, for the UAVD4L-LoD dataset, we use a subset dataset of synthesis images in UAVD4L [72], excluding data that does not contain buildings. For the Swiss-EPFL dataset, synthetic images rendered in Latin Hypercube Sampling (LHS) [74] pattern have been employed as reference images. Notably, the CrossLoc dataset [74] did not include images in proximity to the $out\text{-}of\text{-}Place$ area. To address this, we adopted the same synthetic scheme from [74] to generate synthetic reference images for this region. Figure 10 shows reference samples of RGB images and $Depth$ images for both the mesh-based model and LoD-based model.

E.3 Failure Cases in Baselines

Although baselines have achieved impressive performance, they suffer from retrieving and matching repetitive texture images and cross-modal images. For example, Figure 13 exhibits deficiencies in retrieving repetitive texture images, and Figure 14 depicts poor matching results between RGB and LoD-rendered images.

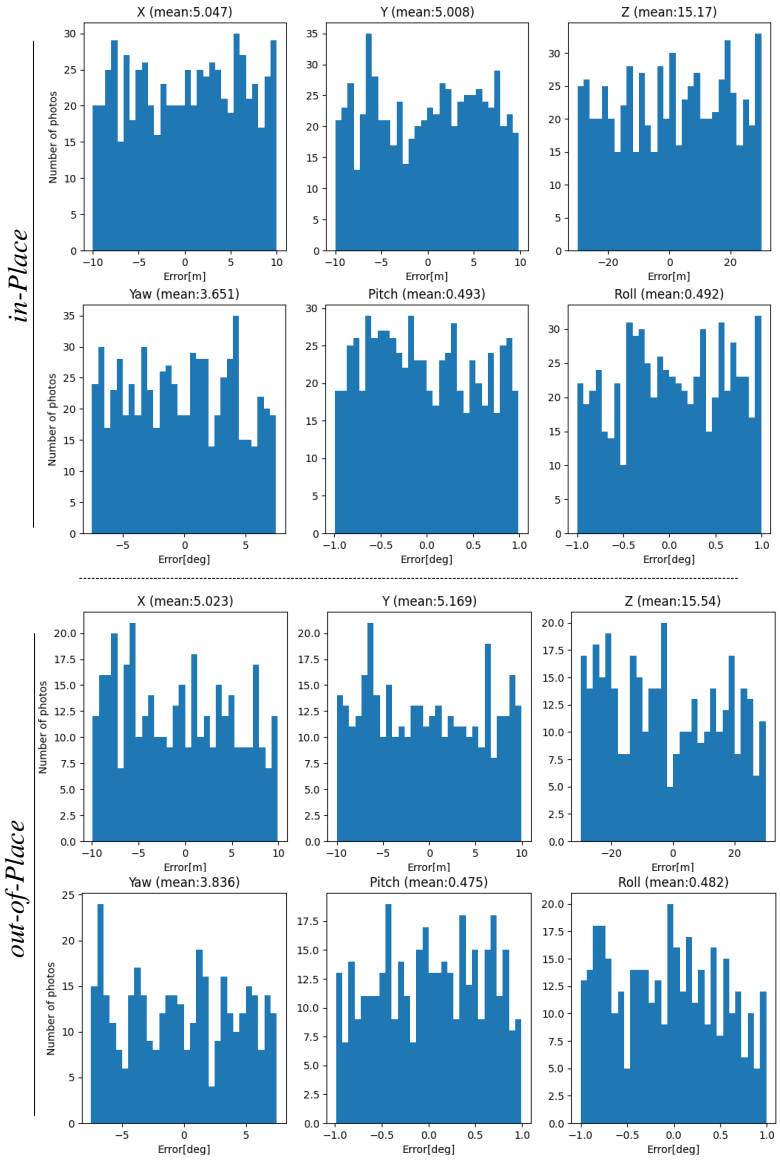

Figure 11: **The discrepancy between our generated poses and GT poses over the Swiss-EPFL dataset.** We use the generated poses to simulate the pose of the sensor.

F Details of Experiments

F.1 Visualization of Training Data

We visualize some synthetic training samples of LoD-Loc, as shown in Figure 15. For the Swiss-EPFL dataset, the reference 3D model is derived from LiDAR point clouds, Terrain Models, and Orthophotos. In contrast, in the UAVD4L-LoD dataset, the reference 3D model is generated from

Table 6: **The architecture of our multi-scale feature extractor**. We discuss the details of each convolutional unit. $conv$ represents a unit consisting of a 2D convolutional layer, a batch normalization layer and a ReLU layer. While $fine\_conv$ denotes a general convolutional layer. $deconv$ means a deconvolutional unit. The colored cells are the outputs for each level l with a single channel.

Layer	Stride	Kernel	Channel	Input
conv0_0	1×1	3×3	3→8	rgb
conv0_1	1×1	3×3	8→8	conv0_0
conv1_0	2×2	5×5	8→16	conv0_1
conv1_1	1×1	3×3	16→16	conv1_0
conv1_2	1×1	3×3	16→16	conv1_1
conv2_0	2×2	5×5	16→32	conv1_2
conv2_1	1×1	3×3	32→32	conv2_0
conv2_2	1×1	3×3	32→32	conv2_1
conv_out1	1×1	1×1	32→1	conv2_2
deconv1_0	2×2	3×2	32→16	conv2_2
concat1	-	-	-	deconv1_0, conv1_2
conv3_0	1×1	3×3	32→16	concat1
conv_out2	1×1	1×1	16→1	conv3_0
deconv2_0	2×2	3×3	16→8	conv3_0
concat2	-	-	-	deconv2_0, conv0_1
conv4_0	1×1	3×3	16→8	concat2
conv_out3	1×1	1×1	8→1	conv4_0
concat3	-	-	-	conv4_0, conv_out3, rgb
fine_conv0	1×1	5×5	12→24	concat3
fine_conv1	1×1	5×5	24→12	fine_conv0
conv_out4	1×1	1×1	12→1	fine_conv1

Table 7: **Ablation study on different threshold γ_t and γ_o for baselines**.

	Method	Threshold (γ_t, γ_o)	*in-Traj.*			*out-of-Traj.*		
			2m-2°	3m-3°	5m-5°	2m-2°	3m-3°	5m-5°
UAVD4L	SIFT+NN	(30, 7.5)	0.62	0.69	4.99	25.87	26.82	27.42
		(50, 15)	27.00	28.30	32.29	55.66	57.44	58.26
		(150, 30)	**73.13**	**78.62**	**80.42**	**82.39**	**85.13**	**86.36**
	SPP+SPG	(30, 7.5)	0.94	0.94	5.24	30.11	30.20	30.29
		(50, 15)	33.92	33.92	37.28	60.99	61.13	61.18
		(150, 30)	**91.71**	**92.02**	**92.14**	**93.43**	**93.70**	**93.80**
	LoFTR	(30, 7.5)	0.94	0.94	5.20	29.79	30.02	30.16
		(50, 15)	33.29	33.35	36.72	60.90	60.90	60.99
		(150, 30)	**84.98**	**88.09**	**88.90**	**91.56**	**92.02**	**92.11**

high-resolution aerial imagery through oblique photography reconstruction. As a result, the synthetic images from the former are of a lower quality. This could partly elucidate why our method yields lower results on the Swiss-EPFL dataset compared to UAVD4L-LoD.

F.2 Additional Ablation Studies

We provide more ablation studies in this section, which include the pose sampling number, the sample density δ of 3D wireframes, the sampling range controller lambda λ. Additionally, we explore the convergence and generalization of our method.

Pose sampling number. As illustrated in Table 9, we report the experimental results with varying numbers of pose samples. The findings suggest that a reduction in the number of sampled poses brings about a decrease in accuracy.

Table 8: **Ablation study on different Top-k for baselines**.

	Method	Top-k	*in-Traj.*			*out-of-Traj.*			Time (s)
			2m-2°	3m-3°	5m-5°	2m-2°	3m-3°	5m-5°	
UAVD4L	SIFT+NN	3	73.13	78.62	80.42	82.39	85.13	86.36	1.85
		10	85.97	89.65	90.52	90.28	92.43	93.66	1.96
		20	88.09	91.33	92.64	92.75	94.71	95.99	2.13
	SPP+SPG	3	91.71	92.02	92.14	93.43	93.70	93.80	1.79
		10	99.25	99.31	99.31	98.45	98.49	98.49	3.31
		20	99.75	99.81	99.81	99.91	99.95	99.95	5.44
	LoFTR	3	84.98	88.09	88.90	91.56	92.02	92.11	1.70
		10	90.21	91.65	92.08	94.75	94.89	94.89	3.78
		20	85.97	87.53	87.91	90.37	90.83	91.29	6.26
Ours		–	84.41	91.77	96.95	95.94	99.00	99.36	0.34

Table 9: **Ablation study on different pose sampling numbers for LoD-Loc.**

Category	Numbers on [θ, x, y, z]	Recall (%)			Median Error	
		2m-2°	3m-3°	5m-5°	T.e. (m)	R.e. (°)
in-Traj.	[2 , 3 , 3 , 8]	18.83	24.94	36.03	7.67	4.37
	[4, 5 , 5 , 15]	77.68	84.98	90.15	1.07	0.59
	[8, 10, 10, 30]	**84.41**	**91.77**	**96.95**	**0.97**	**0.52**
out-of-Traj.	[2 , 3 , 3 , 8]	12.36	16.93	23.81	11.49	5.51
	[4, 5 , 5 , 15]	87.27	93.25	94.25	1.15	0.54
	[8, 10, 10, 30]	**95.94**	**99.00**	**99.36**	**1.06**	**0.49**

3D wireframe points sampling density. We conduct ablation studies for varying sampling densities, which affects the interpolation process on the feature map. As depicted in Table 10, there is no significant fluctuation in localization accuracy with changes in sampling density.

Sampling range controller. The parameter lambda λ adjusts the length of the sampling range. Through ablation studies, we demonstrate that the sensitivity of this parameter during the testing phase is low. The results are shown in Table 11.

Convergence and initial poses. Table 12 reports the localization recall with different initial prior errors on the UAVD4L-LoD dataset. It can be observed that the success rate of localization decreases as the initial prior error increases. Such issues occur when the GPS signal in the air is heavily interfered with. In such cases, we believe using sequence information could be a possible solution.

Cross-scene generalization. Table 13 illustrates the generalization capability of LoD-Loc through training and testing in diverse regions. Figure 16 delineates regional data using distinctive symbols and colors. On the UAVD4L-LoD dataset (A1 and A2), cross-scene testing yields results slightly lower than those obtained from training on the entire scene. For the Swiss-EPFL dataset (B1 and B2), we employ a model trained on the synthetic UAVD4L-LoD dataset, which achieves similar or even better performance compared to a model trained specifically on the Swiss-EPFL dataset. Additionally, the supplementary materials include two demo videos showcasing the model's capacity to localize cross-modal thermal images.

Computational cost comparison. We conducted test experiments on a single batch (Batch Size = 1) of images using the NVIDIA GeForce RTX 4090 device, and recorded the average peak CUDA usage as well as the average inference time. The details are provided in Table 14

F.3 Visualization of Results

We present more visualization results, including examples of corner houses (Figure 12), feature maps (Figure 17) and prediction results (Figure 18) at different levels. We found that the preset zig-zag route in a selected region resulted in some images capturing only the corners of houses, as shown in Figure 18. This led to poorer performance under strict 2m-2° metrics. However, it is important

Table 10: **Ablation study on different wireframe sampling density**. x-m means sampling per-x meter on each wireframes.

	Category	Density	Recall (%)			Median Error	
		δ	2m-2°	3m-3°	5m-5°	T.e. (m)	R.e. (°)
LoD-Loc	*in-Traj.*	4-m	85.10	92.39	96.51	0.95	0.52
		2-m	84.16	91.08	96.95	0.97	0.52
		1-m	84.41	91.77	96.95	0.97	0.52
	out-of-Traj.	4-m	95.21	98.68	99.18	1.00	0.45
		2-m	95.44	98.91	99.32	1.06	0.48
		1-m	95.94	99.00	99.36	1.06	0.49

Table 11: **Ablation study on different Lambda λ.**

	LoD-Loc	Lambda	Recall (%)			Median Error	
		λ	2m2°	3m3°	5m5°	T.e. (m)	R.e. (°)
LoD-Loc	*in-Traj.*	1.5	83.42	91.02	96.57	1.00	0.49
		1	84.41	91.77	97.01	0.95	0.53
		0.8	84.41	91.77	96.95	0.97	0.52
		0.5	84.04	91.58	96.45	0.97	0.52
	out-of-Traj.	1.5	91.97	97.54	98.45	1.11	0.53
		1	95.71	99.04	99.36	1.07	0.50
		0.8	95.94	99.00	99.36	1.06	0.49
		0.5	95.71	98.86	99.32	1.06	0.49

to note that in the *in-Traj.* scenario, our method achieves comparable or superior results for coarse metrics. For instance, we achieve 96.95% on 5m-5° while the closest baseline achieves 92.14%.

Table 12: **Impact of the initial pose for LoD-Loc.** The parameters Δx and Δy denote the error range in the horizontal plane, while Δz represents the error range in the vertical dimension. For instance, $\Delta x = 10$ implies that the initial error in the x value lies within the interval [-10, 10]. The rotation error remains consistent with the real sensor data. All error ranges are measured in meters.

	Category	Prior Error Range $[\Delta x, \Delta y, \Delta z]$	Recall (%)		
			2m-2°	3m-3°	5m-5°
LoD-Loc	*in-Traj.*	[10, 10, 30]	84.41	91.77	96.95
		[20, 20, 30]	87.28	90.77	91.65
		[30, 30, 30]	78.93	82.98	83.85
		[50, 50, 30]	43.08	48.82	50.69
		[100, 100, 30]	5.67	7.36	8.79
	out-of-Traj.	[10, 10, 30]	95.94	99.00	99.36
		[20, 20, 30]	82.07	88.05	89.55
		[30, 30, 30]	74.27	80.66	81.79
		[50, 50, 30]	46.53	53.60	55.98
		[100, 100, 30]	6.93	9.95	11.99

Table 13: **Cross-scene generalization**. We assess the generalization ability of our method by training and testing on different regions. The regional divisions are illustrated in Figure 16, identified by a specific color and letter.

	Train region *Synthesis*	Test region *Real*	Recall (%)		
			2m-2°	3m-3°	5m-5°
LoD-Loc	A2	A1	83.39	91.50	96.81
	A1, A2	A1	**89.51**	**95.01**	**97.98**
	A1	A2	82.54	91.01	91.52
	A1, A2	A2	**95.56**	**98.66**	**99.38**
	A1, A2	B1	**55.41**	**71.77**	**84.17**
	B1, B2	B1	37.73	57.26	77.57
	A1, A2	B2	**50.00**	59.27	65.45
	B1, B2	B2	48.60	**65.31**	**79.78**

Method		Memory (Mb)	Time (s)
UAVD4L	SPP	610	1.79
	SIFT	443	1.85
	LoFTR	2631	1.70
	RoMA	5488	4.68
	eLoFTR	1650	1.06
ours		4810	0.34

Table 14: **Computational cost comparison.**

Figure 12: **Example of corner houses.**

Figure 13: **Failure retrieval cases of baselines**. Even with narrowed searching scopes, the retrieval phase still suffers from issues such as repetitive textures and cross-modal challenges.

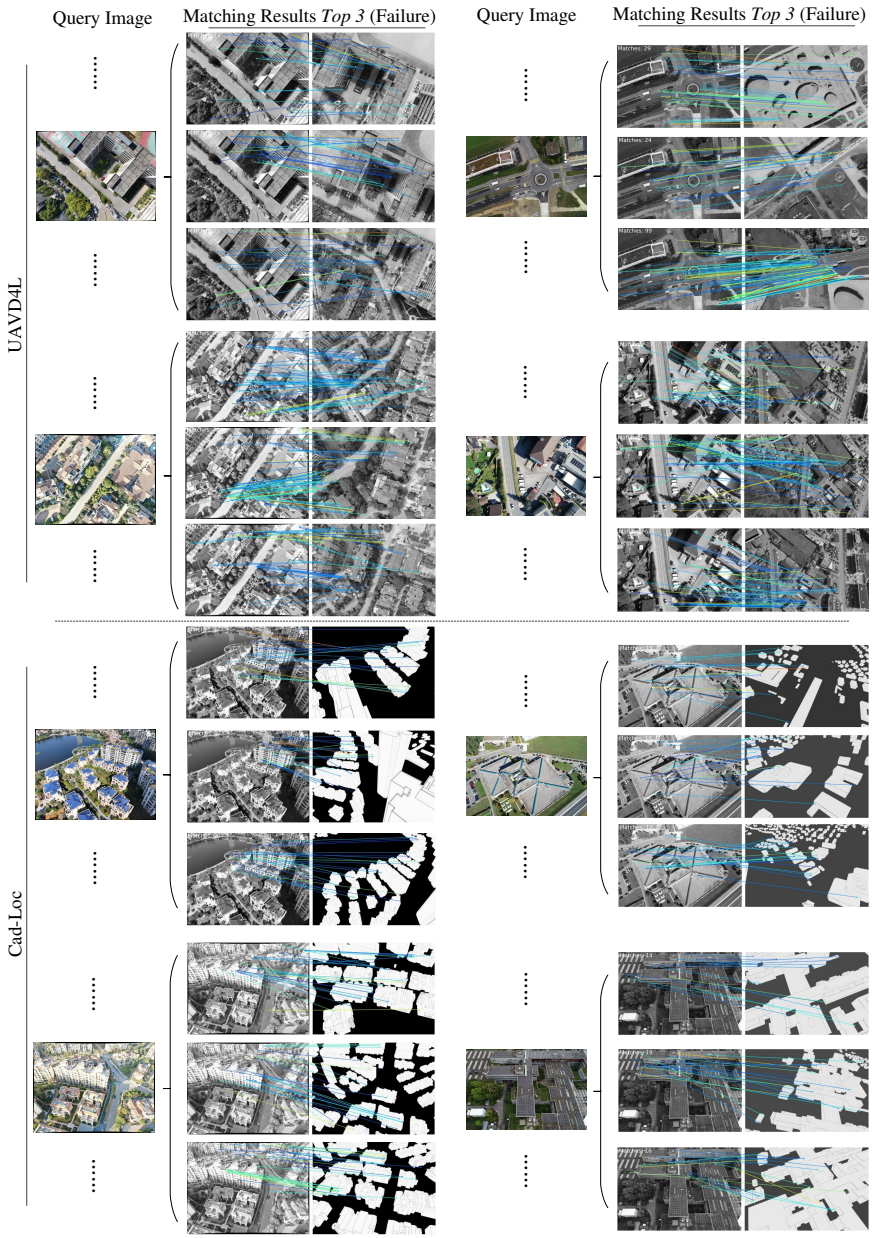

Figure 14: **Failure matching cases of baselines**. The differences in viewpoint and modality influence the results for image matching.

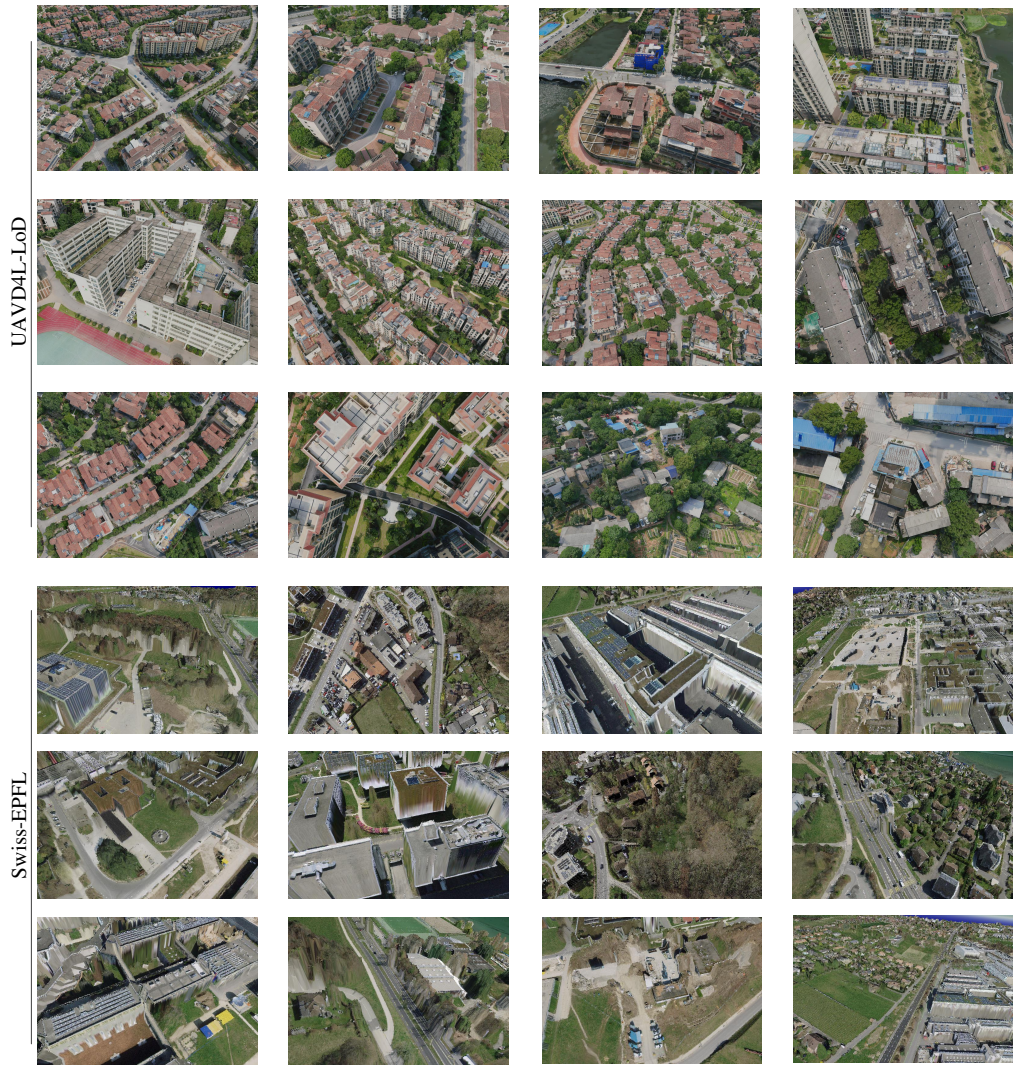

Figure 15: **Samples of the training dataset**. The UAVD4L-LoD dataset offers high-quality training set, while the Swiss-EPFL dataset suffers from lower quality, as evidenced by issues such as blurriness and voids on the sides of buildings.

UAVD4L-LoD Swiss-EPFL

Figure 16: **Region of training and testing**. We use boxes with different colors and symbols to delineate different regions.

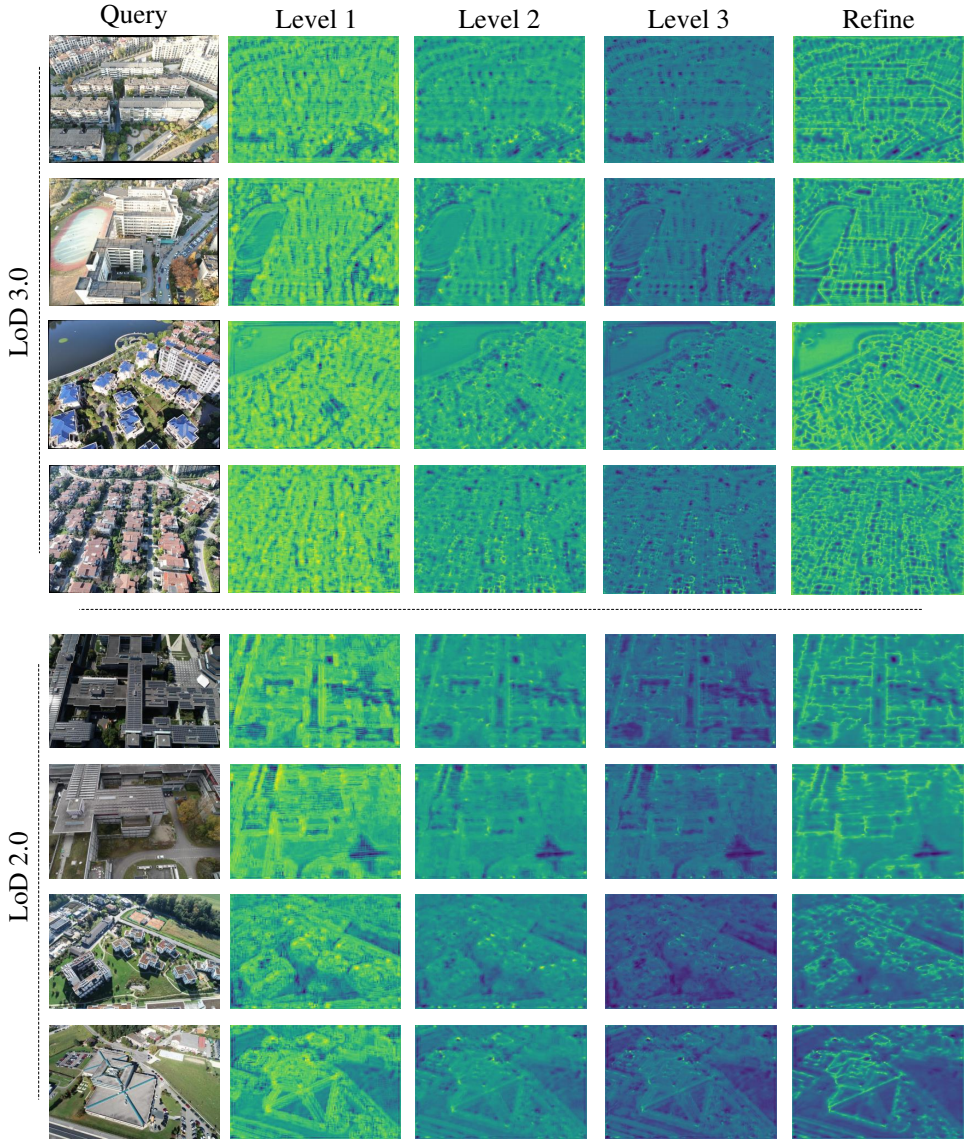

Figure 17: **Visualization of feature maps at different levels**. The feature maps at different levels reflect varying degrees of fineness in wireframe extraction.

Figure 18: **Visualization of predictions at different levels**. Based on the predicted poses at each stage, we can obtain 2D projected wireframe and overlay them on the query image to check the accuracy of the poses. It can be observed that as the levels progress, the projected wireframes gradually align with the edges of the buildings. Please zoom in to see the details of the alignment.