PGGT-Loc: Primitive-Grounded Geometry Transformer for Feed-Forward Camera Localization

Hanqiao Ye[1,2] **Yuzhou Liu**[1,2] **Mengqi Rong**[2] **Jian Liu**[3]
Yangdong Liu[2*] **Shuhan Shen**[1,2*]

[1]School of Artificial Intelligence, University of Chinese Academy of Sciences
[2]Institute of Automation, Chinese Academy of Sciences [3]Beihang University
{yehanqiao2022, liuyuzhou2021, yangdong.liu}@ia.ac.cn
{mengqi.rong,shshen}@nlpr.ia.ac.cn liujian@lynmaxtech.com

Abstract

Camera localization methods are mostly multi-stage pipelines relying on dedicated and complex maps that are costly to construct, store, and maintain over time. Alternatively, well-structured maps that organize scenes with compact yet expressive primitives are often readily available, providing rich structural and semantic cues that humans can interpret. However, their potential for direct camera localization remains largely unexplored due to the significant modality gap between map primitives and visual observations. This motivates a new concept for localization, which estimates camera poses by *grounding query observations to map primitives*. In this paper, we instantiate the concept in the context of planar primitives with **PGGT**, the **P**rimitive-**G**rounded **G**eometry **T**ransformer network. Given a single query image and a set of planar map primitives, the network is trained to predict multiple localization targets for feed-forward 6-DoF camera localization that generalizes to unseen environments. With this straightforward and scalable architecture, our method pushes the state-of-the-art localization performance on both dense surface maps and minimal layout models. The code and models are available at https://github.com/3dv-casia/PGGT-Loc.

1 Introduction

Camera localization, the task of estimating the orientation and position of a query image relative to a pre-existing map, stands as a fundamental problem in computer vision and robotics, with wide applications in augmented reality (AR) [95], robotic navigation [42], and multi-agent collaboration [29].

One popular family of approaches, known as structure-based localization, typically establishes 2D-3D correspondences between the query image and the scene structure, either by matching local features [11, 34, 48, 62, 66, 72] or through direct scene coordinate regression [9, 10, 36, 69, 77]. With these explicitly established correspondences, the camera pose is then estimated by a robust geometric solver such as PnP-RANSAC [23, 24, 40]. Note that these approaches require a dedicated map triangulated or trained from a large database of mapping images before a new image can be localized. While more general map representations have been explored for the task, *e.g.*, meshes [57] and NeRF/3DGS [17, 33, 54, 91], these methods still depend heavily on photorealistic scene appearance and often require an initial pose estimate for keypoint matching or iterative pose refinement.

Well-structured maps are arrangements of high-level primitives and may further be enriched with relationships among them [3–5, 49, 50, 63, 88, 97]. Despite the significant modality gap between the map primitives and visual observations, which hinders the straightforward adaptation of the

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

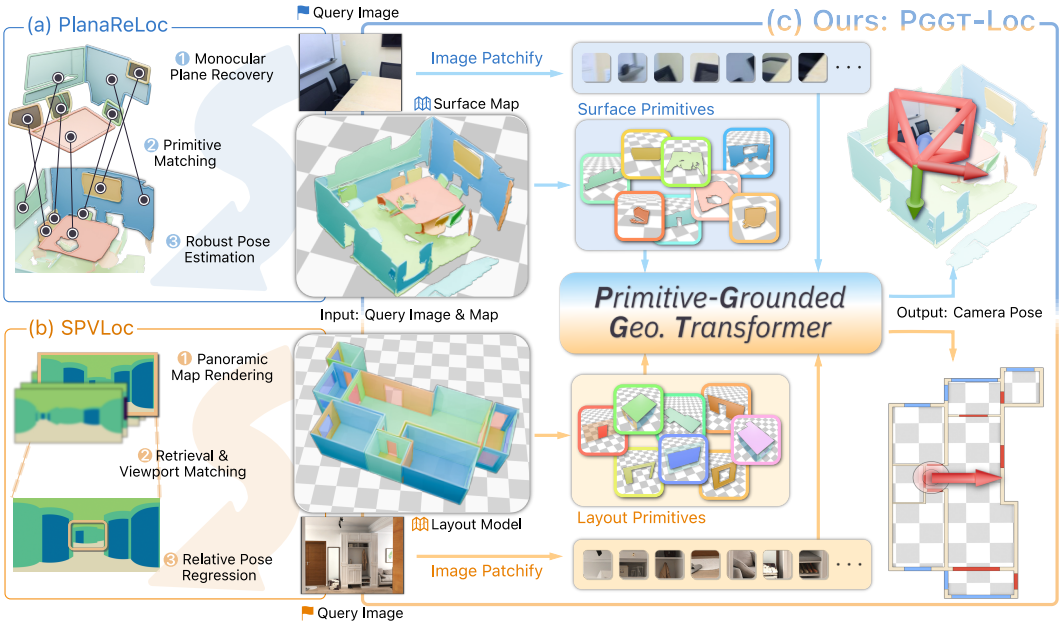


Figure 1: **Overview.** (a) PlanaReLoc [90] localizes cameras against *dense surface maps* by explicitly matching 2D plane proposals with planar map primitives. (b) SPVLoc [25] localizes cameras against *minimal layout models* by retrieving the best-matching panoramic rendering. (c) Our PGGT-Loc operates directly on query patches and map primitives, enabling feed-forward localization across both *dense surface maps* and *minimal layout models* with a unified base architecture.

aforementioned localization paradigms, we argue that these map representations encode rich structural and even semantic information that remains underexploited for direct camera localization.

Among the many map primitives, planes are particularly essential in representing physical surfaces, owing to their notable compactness and prevalence in structured environments [2, 83, 86], which has motivated a recent work, named PlanaReLoc [90], to explore their use for camera localization. As illustrated in Fig. 1a, the method introduces a plane-centric feature-matching pipeline that first establishes cross-modal correspondences between planes in the query and the map, and then estimates the pose via 2D–3D registration. While the method demonstrates the potential of planar primitives for streamlined camera localization without map texture, pose priors, or per-scene training, it still relies on an off-the-shelf plane recovery module to generate reliable 2D plane proposals and geometric priors for primitive matching and pose estimation, which poses a major bottleneck in practice.

Meanwhile, the three-stage pipeline in PlanaReLoc cannot be directly applied to minimal layout models [19, 97], whose primitives are restricted to architectural entities that do not span all physical surfaces. For localization on such minimalist maps, the *render-and-compare* paradigm dominates existing methods. As depicted in Fig. 1b, the state-of-the-art method SPVLoc [25] first renders the layout map into a set of semantic panoramas by exhaustively sampling viewpoints across the scene, and then regresses the relative pose between the best-retrieved panorama and the query image. Although this paradigm neatly bridges the modality gap, reducing structure-based localization to matching 2D renderings inevitably discards structural information that could otherwise be exploited.

In this work, motivated by the goal of fully leveraging high-level map primitives for more straightforward camera localization, we introduce *primitive-grounded localization*, a new concept that casts the task as the grounding of query observations in map primitives. Driven by the concept and inspired by the DUSt3R [82] architecture, we propose **PGGT**, a feed-forward network whose name stands for **P**rimitive-**G**rounded **G**eometry **T**ransformer. As shown in Fig. 1c, PGGT takes a single query image and a set of map primitives as inputs, and predicts the camera pose w.r.t. the map in a single forward pass, eliminating the need to solve for the pose from explicit primitive correspondences or match the query against exhaustive 2D renderings of the map. We summarize our **contributions** as follows:

- We introduce *primitive-grounded localization*, a new concept to localize cameras against well-structured maps through the grounding of query observations in map primitives.

- The concept is concretized in the context of planar primitives by PGGT, a cross-modal transformer network designed to ingest the query image and map primitives as direct inputs, and predict a set of localization-specific targets for feed-forward camera localization.
- PGGT provides a shared base architecture for localization tasks on both surface maps and layout models, surpassing prior works and demonstrating the potential of primitive-grounded localization as a promising paradigm for diverse primitive-based map representations.

2 Related Work

Localization with Higher-Level Maps. Through computer vision, machines position themselves in the real world using diverse map representations, many of which are not encoded with visual observations but instead consist of high-level structural primitives and semantics interpretable by humans. In large outdoor scenarios, prior works have explored 2D semantic maps for 3-DoF localization, evolving from using only a few semantic classes [28, 31, 59, 76] to utilizing all available map elements [1, 63, 64, 84]. CAD/LoD models of urban buildings have also been used for 6-DoF localization [58, 99]. For indoor environments, 2D floorplan localization methods typically employ the *render-and-compare* paradigm, converting maps into diverse codebooks that facilitate intra-modal reasoning with query images [16, 32, 38, 52]. PlanaReLoc [90] and SPVLoc [25] exploit detailed planar surface maps and minimal layout models for 6-DoF camera localization, respectively, while adhering to conventional paradigms where primitive-encoded structural information remains insufficiently exploited.

Multi-view Transformers. DUSt3R [82] comes with the two-view transformer architecture which comprises a shared ViT [22] encoder and view-specific interconnected decoders, realizing feed-forward recovery of scene geometry and reshaping the landscape of 3D vision. Building upon this foundation, subsequent works have expanded the paradigm along multiple dimensions, such as multi-view inputs [43, 79], correspondence grounding [41], online streaming inference [18, 55, 80, 100], more geometric inputs [35, 39], dynamic scene modeling [96] and novel view synthesis [13, 70]. In particular, the paradigm also inspires several *structureless* localization methods [6, 21], which operate on groups of query and mapping images. In this work, we adapt DUSt3R's scalable architecture to *cross-modal* inputs, aiming for feed-forward localization with high-level map primitives.

3 Method

We propose PGGT, a cross-modal transformer network that instantiates the concept we propose: localizing by *grounding query observations to map primitives*. PGGT is designed to ingest the query image and map primitives as direct inputs and predict multiple localization-specific targets, enabling feed-forward camera localization on well-structured maps that generalizes to unseen environments.

3.1 Problem Formulation

We unfold our method in the context of the dense surface map $\mathbb{M}$ [53, 74, 89, 93], where a scene is represented as an inclusive collection of piecewise planar surfaces $\{\Pi_i^m | i = 1, \dots, l_m\}$ with high geometric fidelity preserved. In the absence of additional information such as visual appearance, each map primitive $\Pi_i^m \in \mathbb{M}$ is fully determined by its plane parameter $\boldsymbol{\pi}_i^m \in \mathbb{S}^2 \times \mathbb{R}$ and spatial extent Ω_i^m (represented by a boundary edge loop). We use coefficients of the plane equation: $\mathbf{n} \cdot \boldsymbol{x} = d\,(\boldsymbol{x} \in \mathbb{R}^3)$ to define the plane parameter: $\boldsymbol{\pi} := [\mathbf{n}, d]$, which is the concatenation of the plane normal $\mathbf{n} \in \mathbb{S}^2$ and the plane offset $d \in \mathbb{R}$.

Given a previously *unseen* map $\mathbb{M}$ and a query image $\boldsymbol{I}^q \in \mathbb{R}^{3\times H\times W}$, the objective is to estimate the 6-DoF camera pose $\mathbf{g} \in \mathbb{S}^3 \times \mathbb{R}^3$ w.r.t. $\mathbb{M}$ that corresponds to the query observation. We parametrize the camera pose as $\mathbf{g} := [\mathbf{q}, \mathbf{t}]$, with $\mathbf{q} \in \mathbb{S}^3$ and $\mathbf{t} \in \mathbb{R}^3$ denoting the rotation quaternion and translation vector, respectively. We assume that the query image is captured within the spatial extent of the map, with the observed region sufficiently mapped and exhibiting no major structural discrepancies with the map. In addition, our method does not necessarily require priors on camera intrinsics.

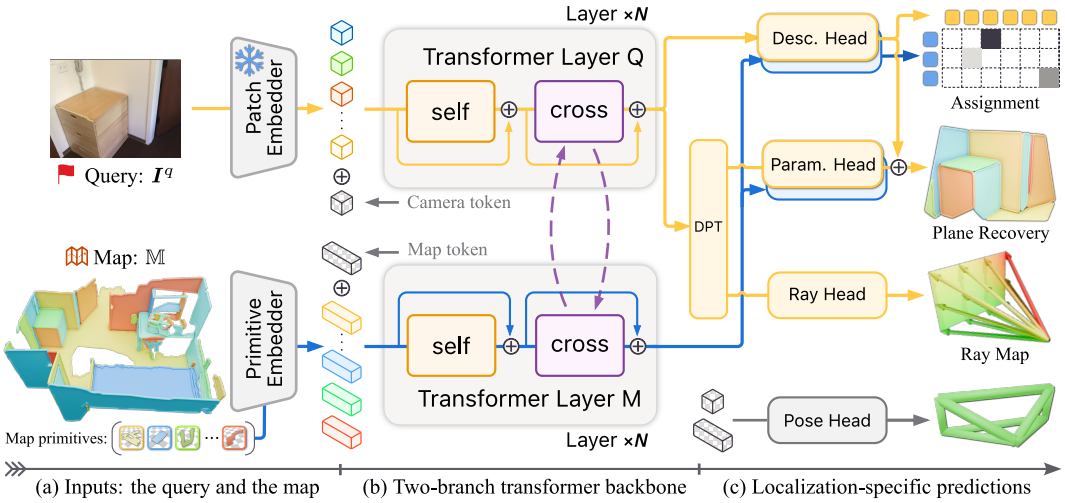


Figure 2: **Model Architecture**. (a) Two embedders first tokenize the inputs into patch and primitive tokens, respectively. (b) The two-branch transformer backbone refines the two sets of tokens, together with the prepended camera and map tokens, and enables interactions between them. (c) Downstream prediction heads produce different localization targets that enable camera pose inference via three pathways, which we refer to as the (1) Param.-based, (2) Ray-based, and (3) Token-based approaches.

3.2 Model Architecture

The PGGT architecture, illustrated in Fig. 2, consists of two main components: (1) a two-branch transformer backbone consuming two modalities of inputs, namely the query and the map; (2) a set of output layers predicting multiple localization-specific targets.

Cross-Modal Transformer Backbone. In light of the tremendous potential of multi-view transformers in feed-forward 3D reconstruction, we inherit the architecture, adopting a backbone comprising two identical stacks of N transformer decoder layers, with one stack processing query tokens and the other processing map tokens. Here, these two inputs are simply tokenized as follows:

- The query image $\boldsymbol{I}^q$ is patchified into l_q tokens through a ViT encoder, which is pre-trained on large-scale image corpora and remains frozen throughout training.
- The map, represented as an unordered set of planar primitives $\{\Pi_i^m | i = 1, \ldots, l_m\}$, is encoded into l_m tokens via a trainable projector, termed the primitive embedder. The primitive embedder tokenizes each map primitive through the integration of: (1) a vanilla 3D encoder that consumes points sampled on the primitive surface to capture its shape and spatial position; (2) a shallow MLP that projects the primitive parameter $\boldsymbol{\pi}_i^m$ into a latent space, serving as the positional encoding.

Within each layer of the transformer backbone, the cross-attention block facilitates cross-modal interactions between the patch tokens and the primitive tokens. To predict dense outputs from the query image, a dedicated DPT layer is attached to the query branch, which converts patch tokens into pixel-aligned dense features. The DPT layer is instantiated with the Dual-DPT architecture proposed in DA3 [43]: patch tokens collected from intermediate transformer layers are initially aggregated through shared reassembly modules, and then fused via separate fusion layers to produce task-specific feature maps.

Prediction on Patch-Primitive Assignment. An intuitive practice to propel the process of primitive grounding is to explicitly learn the unilateral assignment from image patches to map primitives. To this end, two distinct descriptor heads, one for each modality, project the backbone tokens into a shared feature space. Specifically, each token is projected by a 3-layer MLP into a descriptor $\mathbf{f} \in \mathbb{R}^c$ and a confidence score $\sigma \in [0, 1]$. Each entry $\tilde{\boldsymbol{A}}_{ij} (i \in [1, l_q], j \in [1, l_m])$ of the soft assignment matrix, indicating the likelihood of the i-th image patch being matched/grounded to the j-th map primitive, is formulated as a combination of descriptor similarity $\boldsymbol{S}_{ij}$ and confidence scores:

$$\tilde{\boldsymbol{A}}_{ij} = \sigma_i^q \sigma_j^m \cdot \underset{k \in [1, l_m]}{\text{Softmax}} \left(\boldsymbol{S}_{ik} \right)_j, \text{ where } \boldsymbol{S}_{ik} = \mathbf{f}_i^q \cdot \mathbf{f}_k^m. \tag{1}$$

Each image patch with a confidence score above τ is grounded to the primitive with the highest assignment score. Formally, the final assignment matrix is defined as $\hat{\boldsymbol{A}}_{ij} = 1$ if $j = \arg\max_{k \in [1, l_m]} \tilde{\boldsymbol{A}}_{ik}$ and $\sigma_i^q > \tau$, indicating that the i-th patch is assigned to the j-th primitive, and $\hat{\boldsymbol{A}}_{ij} = 0$ otherwise.

Prediction on Primitive Parameters. By predicting viewpoint-invariant 3D pointmaps, multi-view transformers iteratively recover scene structures and camera poses layer by layer [71]. Departing from their point-based representation, we further enhance our primitive grounding with 3D geometry learning by tasking our model to predict frame-invariant primitive parameters, enabling both local and mutual structure reasoning between the query and the map. In the query branch, one of the dense feature maps from the DPT layer is mapped into a per-pixel plane parameter map $\boldsymbol{P} \in \mathbb{R}^{3 \times H \times W}$ through a convolutional layer, while tokens from the map branch are decoded into plane parameters using a 3-layer MLP. Following [47, 68, 87, 94], the prediction of a plane parameter here is a 3D vector defined as $\mathbf{n}/d$. As map coordinate frames could be arbitrarily specified and may vary significantly across different input augmentations, we anchor all predictions of plane parameters in the query coordinate space, which provides a stable and consistent reference across different samples. Conceptually, this design encourages the query branch to focus on perceiving *local* structure, while the map branch learns to align its primitives to the query view.

Prediction on Camera Rays. To directly target localization, a specialized prediction head is attached to the query branch to decode the dense features into a per-pixel ray map $\boldsymbol{R} \in \mathbb{R}^{6 \times H \times W}$. Each camera ray $\mathbf{r}$ corresponds to the pixel through which it passes, and is defined as a 6D vector $[\mathbf{d}; \mathbf{o}]$ where the ray direction $\mathbf{d} \in \mathbb{R}^3$ and ray origin $\mathbf{o} \in \mathbb{R}^3$ are under the map coordinate frame. The ray map serves as an over-parameterized camera representation, implicitly encoding both the pose $\mathbf{g}$ and intrinsics. In the case of a classical perspective camera, the translation $\mathbf{t}$ is estimated by averaging the ray origins, whereas the intrinsics and rotation $\mathbf{q}$ are derived via decomposition of an optimal ray direction homography from an identity camera to the query camera, as detailed in [43].

Prediction from Prepended Tokens. To implement a more straightforward localization alternative while facilitating multi-task learning, we prepend a camera token to the query sequence and a map token to the map sequence. At the final backbone layer, the two tokens are decoded into corresponding poses through a shared MLP. In the map branch, the targeted pose is fixed to the identity pose with the rotation quaternion $\mathbf{q}_0 = [0, 0, 0, 1]$ and translation $\mathbf{t}_0 = [0, 0, 0]$, which represents the map's reference frame. Accordingly, the query pose in this reference frame can be directly decoded from the camera token.

Summary. Towards camera localization, PGGT predicts multiple localization targets in a single forward pass, including patch-primitive assignments, primitive parameters, camera rays, and poses. We then describe three pathways through which the pose of a query can be inferred: (1) solve a point-to-plane registration problem using the predicted patch-primitive assignment $\hat{\boldsymbol{A}}$, the predicted per-pixel parameters $\hat{\boldsymbol{P}}$ and the given parameters of map primitives $\{\boldsymbol{\pi}_i^m\}$, as in [90]; (2) derive the pose from the predicted ray map $\hat{\boldsymbol{R}}$; (3) directly decode the pose from the camera token. We refer to the three alternatives above as *Param.*-based, *Ray*-based and *Token*-based approaches, respectively.

3.3 Training

Supervision. PGGT is trained with a multi-task loss:

$$\mathcal{L} = \mathcal{L}_{\text{assign}} + \mathcal{L}_{\text{param}} + \mathcal{L}_{\text{ray}} + \mathcal{L}_{\text{pose}}, \tag{2}$$

which combines supervision signals on all the prediction targets. Specifically, $\mathcal{L}_{\text{assign}}$ is the negative log-likelihood loss on the soft assignment $\tilde{\boldsymbol{A}}$, with supervision from the ground-truth assignment $\boldsymbol{A}$; $\mathcal{L}_{\text{param}}$ combines the L^1 loss and the cosine similarity loss applied to the predicted parameters; and both the ray map loss $\mathcal{L}_{\text{ray}}$ and the pose decoding loss $\mathcal{L}_{\text{pose}}$ are formulated using the L^1 norm.

Training Datasets. Following the dataset processing workflow constructed by [90], we scale our training dataset up to approximately 400K query-map pairs from three public datasets: ScanNet [20], ScanNet++ [92] and Matterport3D [12], covering over four thousand indoor scenes/regions in total. Since ScanNet is also used for evaluation, scenes for training and testing are strictly separated to ensure that all evaluation scenes are never seen during training.

Implementation Details. The patch embedder is instantiated with the DINOv2 ViT-Large [56] backbone. In the primitive embedder, we employ PointNet [60] as the 3D encoder, which takes 1024

points sampled from each primitive as input and is trained from scratch. The cross-modal transformer backbone is configured with $N = 12$ layers for each modality and we feed patch tokens from the 2nd, 5th, 8th, and 11th layers into the DPT layer. The training images are resized and patchified into l_q=1200 tokens. Maps are isotropically resized and centered into a unit box, and all predictions under the map coordinate frame should follow the same normalization. We use AdamW [51] and a cosine LR schedule with a peak learning rate of 2×10^{-4}. The model is trained for 125K iterations by default, with bfloat16 precision and a batch size of 128 , which requires $\sim$2 days on 8$\times$H20 GPUs.

4 Extension: Localization on 3D Layout

Having introduced PGGT in the dense surface map, where query observations are grounded to map primitives that faithfully preserve the scene's physical surfaces, we note that the same paradigm naturally extends to minimalist 3D semantic models that represent only architectural layout.

Task Setting. The scene layout follows the annotation format of Structured3D [97]. As shown in Fig. 3, it can be translated into a 2D floorplan or a 3D minimal layout model that comprises a set of planar primitives, each corresponding to an architectural element. Compared with primitives in the dense surface map, those in the minimal layout model exhibit stronger regularity in terms of both their parameters and spatial extents, and each primitive is further associated with one of five distinct structural categories: wall, floor, door, window, and ceiling. Given such a minimal layout model, we focus on localizing query images captured in a fully *furnished* scene, where the increased modality gap poses a greater challenge to the detector-based feature matching conventions [90].

4.1 Model Adaptation

Our base model performs reasonably well on the task. Here, we introduce two additional adaptations, highlighting its flexibility and potential in leveraging diverse map characteristics.

Semantics Injection. To integrate the provided primitive semantics, we initialize five learnable semantic embeddings within the primitive embedder. Each primitive token is enhanced with semantics by fusing its corresponding semantic embedding. We further supervise the query branch with an auxiliary semantic segmentation task after the DPT layer, fostering interactions between geometric and semantic representations.

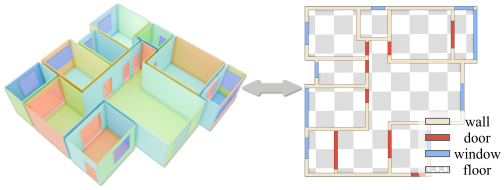


Figure 3: **Scene architectural layout.** *Left*: 3D layout model with ceilings culled. *Right*: corresponding 2D floorplan with semantic labels.

Isolation Injection. Upon close inspection, as shown in Fig. 3, the layout primitives are organized at the room level, *i.e.*, no primitive is shared across rooms. Meanwhile, akin to prior works [16, 25], we do not look into neighboring rooms through doors, treating all room connections as opaque. Consequently, query observations are expected to be grounded only to primitives within the same room. We enforce the isolation bias by applying attention masks to the self-attention blocks in the map backbone, ensuring that each primitive token attends only to other tokens within the same room.

4.2 Training

The total loss in Eq. (2) is augmented with the cross-entropy loss $\mathcal{L}_{\text{seg}}$ for semantic supervision. Following [25], our model is trained from scratch on 3250 layout models from the training split of Structured3D [97], with query images randomly sampled from 21 835 photorealistic panoramic images. Both *empty* and *furnished* panoramas are used for training, and query images are sampled with FoVs ranging from 45° to 135°. Due to the shape regularity, the number of points sampled per primitive is reduced to 256. Unless otherwise specified, the training recipe follows the settings in Sec. 3.3.

5 Experiments

5.1 Results on Dense Surface Maps

Evaluation Datasets. Following the protocol and datasets proposed by PlanaReLoc [90], we first conduct an evaluation of our method on plane-based surface maps using two public indoor datasets: ScanNet [20] and 12Scenes [75]. The test set of ScanNet consists of 7735 real-world images captured in 303 test scenes, while 1023 queries from 12Scenes are used for out-of-domain evaluation.

Evaluation Metrics. To assess localization performance, we report results in terms of the mean/median localization errors: $\Delta\mathbf{q}$ and $\Delta\mathbf{t}$ for rotation and translation, respectively, along with pose recall, defined as the ratio of successful localizations under three thresholds. To assess how well the query observations are grounded to the map primitives, we transform the predicted patch-primitive assignment $\hat{\boldsymbol{A}}$ to 2D–3D plane correspondences by directly aggregating all patches assigned to the same primitive into a single 2D plane. Then, following [90], we analyze the matching performance using Precision, Recall, F-score and Average Precision (AP) at a mask IoU threshold of 0.3.

Table 1: **Quantitative results** on ScanNet. We report rotation and translation errors, along with pose recalls across three thresholds. We also report the average inference time per query, using a single NVIDIA RTX 3090 GPU. Top-3 results are highlighted as the **first**, *second*, and third.

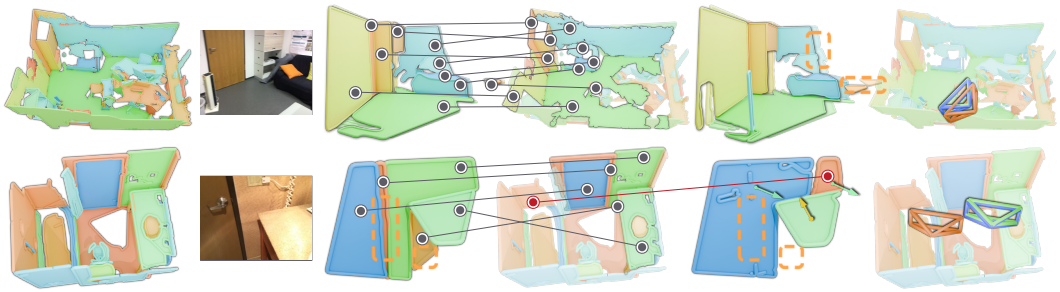
	Methods	$\Delta\mathbf{q}$ (°) ↓		$\Delta\mathbf{t}$ (m) ↓		Pose Recall (%) ↑			Time (s/iter) ↓
		Mean	Med.	Mean	Med.	(0.2 m, 10°)	(0.5 m, 15°)	(1.0 m, 30°)	
MeshLoc	MASt3R [41]	46.0	12.2	1.02	0.43	35.4	49.5	57.6	∼ 0.68
	MatchAnything [30]	35.7	19.9	1.23	0.74	20.0	35.7	52.1	∼ 0.93
	NOPE-SAC [73]	28.7	15.9	0.90	0.77	3.3	21.2	54.6	∼ 0.37
	PLANA3R [44]	26.8	12.9	0.92	0.52	17.9	37.6	57.1	∼ 0.42
	PlaneRecTR++ [68]	20.7	9.9	0.66	0.54	12.2	39.2	75.1	∼ 0.41
I2P	Geotransformer [61]	45.2	26.5	1.42	1.06	24.6	38.8	42.9	∼ 0.30
	FreeReg [78]	36.9	29.4	1.06	0.96	0.8	6.4	33.1	∼ 14.2
RM	PlanaReLoc [90]	17.3	3.9	0.65	0.27	37.1	69.8	79.8	∼ **0.10**
	↪ +Refine [90]	*17.2*	*3.8*	*0.60*	*0.20*	*48.5*	*73.1*	*81.8*	∼ 0.54
	Ours (Ray)	**7.5**	**2.7**	**0.25**	**0.14**	**69.0**	**92.7**	**95.4**	∼ 0.11
	↪ Token	7.8	2.9	0.26	0.14	68.9	92.5	95.3	-
	↪ Param.	8.6	2.5	0.40	0.16	59.5	84.2	90.0	-

Table 2: **Quantitative results** on 12Scenes. We present the cross-dataset performance in terms of both localization (*i.e.*, median errors and pose recall) and primitive matching.

	Med. Err. ↓		Pose Recall (%) ↑			Primitive Matching (%) ↑			
Methods	$\Delta\mathbf{q}$ (°)	$\Delta\mathbf{t}$ (m)	(0.2 m, 10°)	(0.5 m, 15°)	(1.0 m, 30°)	Prec.	Rec.	F_1	AP
MASt3R [41]	12.0	0.30	45.2	51.9	59.0	59.8	42.9	50.0	81.6
MatchAnything [30]	7.9	0.20	46.9	63.4	77.6	51.2	*56.1*	53.5	77.2
NOPE-SAC [73]	17.7	0.54	2.9	25.9	67.2	43.8	22.0	29.3	71.4
PLANA3R [44]	15.9	0.50	7.9	30.1	69.3	-	-	-	-
PlaneRecTR++ [68]	9.8	0.39	11.1	53.7	*83.5*	55.0	24.9	34.3	75.4
PlanaReLoc [90]	4.8	0.28	34.9	66.7	79.9	*63.9*	54.2	*58.6*	*87.8*
↪ +Refine [90]	*4.7*	*0.19*	*50.6*	*70.8*	80.6	-	-	-	-
Ours (Ray)	**3.6**	**0.15**	**68.6**	**90.1**	**93.8**	**77.7**	**74.3**	**76.0**	**92.4**

Quantitative results on ScanNet are presented in Tab. 1. We group our baselines into three categories: (1) MeshLoc [57, 58], which localizes by estimating the relative pose between the query and the photorealistic rendering from the map, given a coarse initialization; (2) Image-to-Point cloud registration (I2P), which localizes by establishing 2D–3D point correspondences; (3) Region Matching (RM), which localizes by matching image regions to map primitives. The results of all baselines are taken from [90]. As the only feed-forward method capable of reconciling the query image and the structured map within a unified and scalable framework, our method achieves leading performance across all

evaluation metrics. Particularly, poses derived from the ray map yield the best results, achieving an average relative improvement of 28.6 % across the three pose recall metrics over *PlanaReLoc +Refine*. As reported in Tab. 2, our method maintains strong performance in cross-dataset evaluation on 12Scenes, while also achieving holistic improvements in primitive matching. Figure 4 presents qualitative comparisons with PlanaReLoc, where our method delivers more accurate and robust pose estimation, with superior plane recovery and primitive correspondences as by-products.



(a) Input: map & query (b) Plane recovery & grounded primitives (c) PlanaReLoc (d) Localization

Figure 4: **Qualitative comparisons with PlanaReLoc.** Our method robustly produces accurate pose estimation, with (b) superior plane recovery and primitive correspondences as additional outputs. In contrast, (c) PlanaReLoc performs suboptimally and even fails due to poor plane recovery (marked by the boxes) and matching outliers (bottom row, colored in red). (d) Comparison of localization results. Legend: the ground truth, PGGT(Ours), and PlanaReLoc.

5.2 Results on Minimal Layout Models

Evaluation Datasets. For the layout localization task, we evaluate our method on two datasets compiled from the test split of the Structured3D [97] dataset. The first, denoted as S3D-*official*, consists of 6405 high-resolution perspective query images from 250 scenes in the test split. The images are photo-realistically rendered with a 16:9 aspect ratio and a fixed horizontal FoV of 80°, including both empty and furnished scene configurations. The second, denoted as S3D-*resampled*, was introduced by [25], in which the authors resampled query images from furnished RGB panoramas with controlled variations in FoV and observed semantics. We employ two subsets of S3D-*resampled* with FoVs of 90° and 120°, respectively, each containing 3426 square perspective images as queries. Note that before evaluation on S3D-*official*, we fine-tune our model using the officially provided perspective images from the training scenes, following SPVLoc [25].

Table 3: **Quantitative results** on S3D-*official*. Different methods employ either 2D floorplans or 3D layout models as the reference map, with some further leveraging map semantics. We report median 2D localization errors over <1 m inliers, along with pose recall at different thresholds. Top-3 results are highlighted as the **first**, *second*, and third. †method not requiring known intrinsics (*i.e.*, FoV).

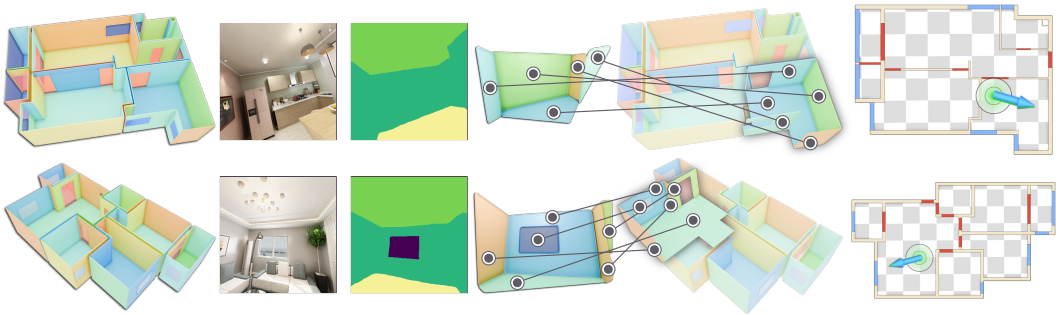
Paradigm	Methods	Map Type	Use Semantics	<1 m Med. Error ↓ (2D)		Pose Recall (%, 2D) ↑			
				Δt_{xy} (cm)	Δq_{yaw} (°)	10 cm	50 cm	1 m	1 m & 30°
Render & Compare	LASER [52]	2D		-	-	7.0	6.4	10.4	8.7
	F^3Loc [16]	2D		-	-	1.5	14.6	22.4	21.3
	3DP [14]	2D		-	-	5.6	27.4	55.5	24.0
	SemRayLoc [27]	2D	✓	-	-	5.7	45.5	58.8	57.5
	RSK [15]	2D		-	-	6.4	28.6	56.9	25.2
	DisCo [98]	2D		-	-	10.0	59.0	67.0	66.0
	UnLoc [85]	2D		-	-	5.3	33.9	38.8	37.6
	SPVLoc [25]	3D	✓	12.9	1.26	33.5	81.6	84.5	84.2
Feed-Forward	**Ours** (Ray)†	3D	✓	11.2	0.42	**40.9**	**91.3**	**93.6**	**93.5**
	↪ Token	3D	✓	11.9	0.46	*37.6*	*91.2*	*93.6*	*93.4*
	↪ Param.	3D	✓	14.4	0.39	27.5	60.5	76.7	76.0

Table 4: **Quantitative results** on S3D-*resampled*. For each FoV subset, we evaluate both 2D and 3D localization errors, along with 2D pose recall under different thresholds. The **best** is highlighted.

		Med. Err. ↓ (3D)		<1 m Med. Error ↓ (2D)		Pose Recall (%, 2D) ↑			
	Methods	$\Delta\mathbf{q}$ (°)	$\Delta\mathbf{t}$ (cm)	$\Delta\mathbf{q}_{\text{yaw}}$ (°)	$\Delta\mathbf{t}_{\text{xy}}$ (cm)	10 cm	50 cm	1 m	1 m & 30°
90° FoV	LASER [52]	-	-	0.67	14.6	23.5	72.8	76.8	76.2
	SPVLoc [25]	-	-	0.83	9.2	50.1	89.7	92.1	91.9
	Ours (Ray)	**0.81**	**11.1**	0.38	8.4	**56.0**	**94.5**	**95.6**	**95.5**
	↪ Token	0.87	11.4	0.41	8.5	55.5	94.4	95.6	95.4
	↪ Param.	0.84	11.5	**0.36**	**7.6**	55.6	87.8	90.6	90.5
120° FoV	LASER [52]	-	-	0.76	12.9	29.1	77.8	82.1	81.4
	SPVLoc [25]	-	-	0.87	**11.7**	39.5	88.0	92.1	91.9
	Ours (Ray)	**0.95**	**14.7**	**0.49**	11.9	**40.9**	**91.7**	**96.4**	**96.3**
	↪ Token	1.00	15.5	0.54	12.8	37.7	91.6	**96.4**	**96.3**
	↪ Param.	1.11	16.8	0.54	12.0	36.7	83.2	87.0	86.2

Evaluation Metrics. Since most prior studies on layout localization focus on estimating 3-DoF camera pose on 2D floorplans, we follow their standard evaluation protocol [16, 52]. Specifically, we extract the 2D translation $\mathbf{t}_{\text{xy}} \in \mathbb{R}^2$ and yaw angle $\mathbf{q}_{\text{yaw}} \in \mathbb{R}$ from our 6-DoF pose predictions, from which we compute recall at multiple thresholds and report median errors over samples successfully localized within 1 m (*i.e.*, $\Delta\mathbf{t}_{\text{xy}} \leq 1$ m).

We first present results on S3D-*official* in Tab. 3, where we also highlight several methodological differences: (1) all prior methods adopt the render-and-compare paradigm, which involves rendering the maps into alternative representations (*e.g.*, a set of semantic panoramas) by sweeping the entire candidate pose space; (2) only SPVLoc [25] and our method utilize 3D layout models for 6-DoF localization. Table 3 shows that our method achieves top performance across evaluation metrics, and it is the only approach that (1) operates in a feed-forward pipeline without requiring exhaustive map rendering, and (2) does not depend on FoV priors. Moreover, our method surpasses SPVLoc on the S3D-*resampled* dataset, as reported in Tab. 4. In Fig. 5, we show two examples on S3D-*resampled*, illustrating localization results along with additional predicted outputs.



(a) Input: map & query (b) Segmentation & plane recovery & grounded primitives (c) Localization

Figure 5: **Qualitative results on Structured3D.** (b) We present the predicted layout segmentation, the 3D layout recovered from the query image, and the grounded map primitives, with the ceilings removed for clarity. (c) We show our localization result together with the ground truth.

5.3 Analysis

Multi-task Learning. We conduct ablation studies to evaluate the benefit of predicting multiple targets in our model. To reduce computational cost and better assess convergence behavior, models in all ablation experiments are trained for a reduced budget of 75K iterations. For the basic components introduced in Sec. 3.2, we verify the benefit of training the network to predict multiple localization targets that go beyond the ray map. As shown in Tab. 5, training without downstream predictions on patch-primitive assignment and primitive parameters leads to a noticeable drop in localization performance, whereas excluding the prepended camera and map tokens has a milder and mixed effect.

Layout Adaptations. Next, we ablate the two adaptations for layout models in Sec. 4.1, with results on S3D-*resampled*-90° FoV reported in Tab. 6. Notably, incorporating semantics alone results in a performance drop, which we attribute to slower convergence caused by the fusion of semantics. However, the joint integration of semantics and room isolation improves overall performance, indicating a synergistic effect of the two adaptations for this task.

Table 5: **Ablation study of each downstream prediction head** in the base model on ScanNet.

Methods	Med. Err. ↓ $\Delta\mathbf{q}$ (°)	Med. Err. ↓ $\Delta\mathbf{t}$ (m)	Pose Recall (%) ↑ (0.2 m, 10°)	Pose Recall (%) ↑ (0.5 m, 15°)	Pose Recall (%) ↑ (1.0 m, 30°)	Matching F_1
Full (Ray)	*3.0*	**0.16**	**62.0**	**92.8**	*95.2*	**81.0**
↪ *w/o* $\mathcal{L}_{\text{assign}}$	3.1	0.18	54.7	87.0	92.5	-
↪ *w/o* $\mathcal{L}_{\text{param}}$	3.4	0.19	54.9	89.6	95.0	80.0
↪ *w/o* $\mathcal{L}_{\text{pose}}$	**2.9**	*0.17*	*59.7*	*91.0*	**95.5**	*80.8*

Table 6: **Ablating adaptations** for layout localization on S3D-*resampled*-90° FoV.

Methods	<1 m Med. Error ↓ $\Delta\mathbf{q}_{\text{yaw}}$ (°)	<1 m Med. Error ↓ $\Delta\mathbf{t}_{\text{xy}}$ (cm)	Recall@ 1 m & 30° ↑
Base (Ray)	0.69	15.2	90.5
+ *w/* sem.	0.88	17.3	91.6
+ *w/* isolation	*0.53*	*11.2*	*93.8*
Full (Ray)	**0.46**	**9.3**	**95.5**

Explicit Plane Parameters. To explore the potential for extending our framework to non-planar structures, we examine its reliance on explicit plane parameters. These parameters enter training through two components: an MLP that provides positional encodings for map primitives and a parameter prediction task supervised by $\mathcal{L}_{\text{param}}$. We ablate these components individually and jointly on ScanNet, with results reported in Tab. 7. These components primarily improve fine-grained localization accuracy, while coarse recall remains largely unchanged even when both are removed. This suggests that primitive grounding need not rely on explicit plane parameters, motivating future extensions to non-planar map primitives.

Methods	Med. Err. ↓ $\Delta\mathbf{q}$ (°)	Med. Err. ↓ $\Delta\mathbf{t}$ (m)	Pose Recall (%) ↑ (0.2 m, 10°)	Pose Recall (%) ↑ (0.5 m, 15°)	Pose Recall (%) ↑ (1.0 m, 30°)
Full (Ray)	**3.0**	**0.16**	**62.0**	**92.8**	*95.2*
↪ *w/o* pos. enc.	*3.2*	*0.17*	*59.3*	*90.7*	**95.4**
↪ *w/o* $\mathcal{L}_{\text{param}}$	3.4	0.19	54.9	89.6	95.0
↪ *w/o* both	3.4	0.19	51.1	89.4	95.0

Table 7: **Ablating plane-specific components** on ScanNet.

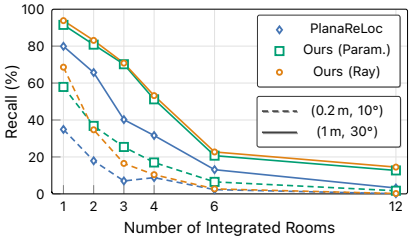


Figure 6: **Quantitative results** on Integrated Rooms [8].

Evaluation on Multiple Rooms. To assess generalization to larger maps, we follow the standard Integrated Rooms protocol [8], arranging 12Scenes [75] rooms on a 2D grid with 5 m cells and progressively merging adjacent rooms. We evaluate the original model without fine-tuning, localizing each query within the integrated map. As shown in Fig. 6, our method consistently outperforms PlanaReLoc [90] across evaluated scales, although recall decreases substantially as more rooms are integrated, likely due to repetitive structures and the distribution gap from room-scale training. These results demonstrate generalization to artificially integrated multi-room maps while highlighting the limitations of full-map inference, motivating coarse-to-fine localization for larger environments.

6 Conclusion

In this paper, we have presented PGGT, a cross-modal transformer network that estimates camera poses from 3D planar primitive-based maps in a feed-forward manner. The design of PGGT serves as an instantiation of the new concept that we refer to as *primitive-grounded localization*. By jointly reasoning over query image patches and map primitives through two modality-specific interconnected transformer stacks, PGGT predicts multiple localization targets, with poses derived from the densely predicted ray map yielding the best performance. Comprehensive evaluation across hundreds of surface maps that comprise dense planar primitives demonstrates the superior performance of our method. Furthermore, our method naturally extends to minimal layout models, outperforming existing layout localization methods and highlighting the potential of primitive-grounded localization as a promising paradigm for diverse primitive-based map representations.

7 Acknowledgements and Disclosures

This work was supported in part by the Beijing Natural Science Foundation (No. L223003), the National Natural Science Foundation of China (No. U22B2055, 62402494, 62273345, 62402495), the Leader-recruiting Bidding Projects for Key Industrial Technology Research of Henan Province (No. 261000320200), the Shandong Provincial Natural Science Foundation (No. ZR2025LGY009) and the CIE-Tencent Robotics X Rhino-Bird Focused Research Program. The authors declare no competing interests.

References

[1] Obed Noé Samano Abonce, Mengjie Zhou, and Andrew Calway. You are here: Geolocation by embedding maps and images. In *ECCV*, 2020. 3

[2] ARCore. Fundamental Concepts: Environmental Understanding. Accessed: 2026-04-25. Available at https://developers.google.com/ar/develop/fundamentals#environmental_understanding. 2

[3] Iro Armeni, Zhi-Yang He, Amir Zamir, Junyoung Gwak, Jitendra Malik, Martin Fischer, and Silvio Savarese. 3D Scene Graph: A Structure for Unified Semantics, 3D Space, and Camera. In *ICCV*. 2019. 1

[4] Armen Avetisyan, Tatiana Khanova, Christopher Bongsoo Choy, Denver Dash, Angela Dai, and Matthias Nießner. SceneCAD: Predicting object alignments and layouts in RGB-d scans. In *ECCV*, 2020.

[5] Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, Jakob Engel, Edward Miller, Richard Newcombe, and Vasileios Balntas. SceneScript: Reconstructing Scenes With An Autoregressive Structured Language Model. In *ECCV*. 2024. 1, 22

[6] Axel Barroso-Laguna, Tommaso Cavallari, Victor Adrian Prisacariu, and Eric Brachmann. A Scene is Worth a Thousand Features: Feed-Forward Camera Localization from a Collection of Image Features. In *ICLR*. 2026. 3

[7] Blender Online Community. Blender - a 3D modelling and rendering package. 2026. Available at http://www.blender.org. 21

[8] Eric Brachmann and Carsten Rother. Expert Sample Consensus Applied to Camera Re-Localization. In *ICCV*. 2019. 10

[9] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC — differentiable RANSAC for camera localization. In *CVPR*, 2016. 1

[10] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using RGB and poses. In *CVPR*. 2023. 1

[11] Federico Camposeco, Andrea Cohen, Marc Pollefeys, and Torsten Sattler. Hybrid scene compression for visual localization. In *CVPR*. 2019. 1

[12] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D Data in Indoor Environments. In *3DV*. 2017. 5, 17

[13] David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D gaussian splats from image pairs for scalable generalizable 3D reconstruction. In *CVPR*, 2024. 3

[14] Bolei Chen, Jiaxu Kang, Haonan Yang, Ping Zhong, and Jianxin Wang. Perspective from a Higher Dimension: Can 3D Geometric Priors Help Visual Floorplan Localization? In *ACM MM*. 2025. 8

[15] Bolei Chen, Shengsheng Yan, Yongzheng Cui, Jiaxu Kang, Ping Zhong, and Jianxin Wang. Perspective from a Broader Context: Can Room Style Knowledge Help Visual Floorplan Localization? In *AAAI*. 2026. 8

[16] Changan Chen, Rui Wang, Christoph Vogel, and Marc Pollefeys. F^{3}Loc: Fusion and Filtering for Floorplan Localization. In *CVPR*. 2024. 3, 6, 8, 9

[17] Shuai Chen, Yash Bhalgat, Xing Hui Li, Jia Wang Bian, Ke Jie Li, Zirui Wang, and Victor Adrian Prisacariu. Refinement for Absolute Pose Regression with Neural Feature Synthesis. In *CVPR*. 2024. 1

[18] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. TTT3R: 3D Reconstruction as Test-Time Training. In *ICLR*. 2026. 3

[19] Steve Cruz, Will Hutchcroft, Yuguang Li, Naji Khosravan, Ivaylo Boyadzhiev, and Sing Bing Kang. Zillow indoor dataset: Annotated floor plans with 360° panoramas and 3D room layouts. In *CVPR*, 2021. 2

[20] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017. 5, 7, 17

[21] Siyan Dong, Shuzhe Wang, Shaohui Liu, Lulu Cai, Qingnan Fan, Juho Kannala, and Yanchao Yang. Reloc3r: Large-Scale Training of Relative Camera Pose Regression for Generalizable, Fast, and Accurate Visual Localization. In *CVPR*. 2025. 3

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*. 2021. 3

[23] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of The Acm*, 24(6):381–395, 1981. 1, 17

[24] Xiaoshan Gao, Xiaorong Hou, Jianliang Tang, and Hangfei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(8):930–943, 2003. 1

[25] Niklas Gard, Anna Hilsmann, and Peter Eisert. SPVLoc: Semantic Panoramic Viewport Matching for 6D Camera Localization in Unseen Environments. In *ECCV*. 2024. 2, 3, 6, 8, 9, 17

[26] GEOS contributors. GEOS computational geometry library. 2025. Available at https://libgeos.org/. 17

[27] Yuval Grader and Hadar Averbuch-Elor. Supercharging floorplan localization with semantic rays. In *ICCV*, 2025. 8

[28] Chengcheng Guo, Minjie Lin, Heyang Guo, Pengpeng Liang, and Erkang Cheng. Coarse-to-fine semantic localization with HD map for autonomous driving in structural scenes. In *IROS*, 2021. 3

[29] Matthew Hanlon, Boyang Sun, Marc Pollefeys, and Hermann Blum. Active Visual Localization for Multi-Agent Collaboration: A Data-Driven Approach. In *ICRA*. 2024. 1

[30] Xingyi He, Hao Yu, Sida Peng, Dongli Tan, Zehong Shen, Hujun Bao, and Xiaowei Zhou. MatchAnything: Universal Cross-Modality Image Matching with Large-Scale Pre-Training, *arXiv preprint arXiv:2501.07556*, 2025. Available at http://arxiv.org/abs/2501.07556. 7

[31] Ziyang Hong, Yvan R. Pétillot, David M. Lane, Yishu Miao, and Sen Wang. TextPlace: Visual place recognition and topological localization through reading scene texts. *ICCV*, pp. 2861–2870, 2019. 3

[32] Henry Howard-Jenkins, Jose-Raul Ruiz-Sarmiento, and Victor Adrian Prisacariu. Lalaloc: Latent layout localisation in dynamic, unvisited environments. In *ICCV*, 2021. 3

[33] Zhiwei Huang, Hailin Yu, Yichun Shentu, Jin Yuan, and Guofeng Zhang. From sparse to dense: Camera relocalization with scene-specific detector from Feature Gaussian Splatting. In *CVPR*, 2025. 1

[34] Martin Humenberger, Yohann Cabon, Nicolas Guerin, Julien Morat, Vincent Leroy, Jérôme Revaud, Philippe Rerole, Noé Pion, Cesar de Souza, and Gabriela Csurka. Robust Image Retrieval-based Visual Localization using Kapture, *arXiv preprint arXiv:2007.13867*, 2022. Available at http://arxiv.org/abs/2007.13867. 1

[35] Wonbong Jang, Philippe Weinzaepfel, Vincent Leroy, Lourdes Agapito, and Jerome Revaud. Pow3R: Empowering unconstrained 3D reconstruction with camera and scene priors. In *CVPR*, 2025. 3

[36] Xudong Jiang, Fangjinhua Wang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. R-SCoRe: Revisiting Scene Coordinate Regression for Robust Large-Scale Visual Localization. In *CVPR*. 2025. 1

[37] Linyi Jin, Shengyi Qian, Andrew Owens, and David F. Fouhey. Planar surface reconstruction from sparse views. In *ICCV*. 2021. 17

[38] Peter Karkus, David Hsu, and Wee Sun Lee. Particle filter networks with application to visual localization. In *CoRL*. 2018. 3

[39] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal Feed-Forward Metric 3D Reconstruction. In *3DV*. 2025. 3

[40] Viktor Larsson and contributors. PoseLib - minimal solvers for camera pose estimation. 2020. Available at https://github.com/vlarsson/PoseLib. 1

[41] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding Image Matching in 3D with MASt3R. In *ECCV*. 2024. 3, 7

[42] Jiajie Li, Boyang Sun, Luca Di Giammarino, Hermann Blum, and Marc Pollefeys. ActLoc: Learning to Localize on the Move via Active Viewpoint Selection. In *CoRL*. 2025. 1

[43] Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the Visual Space from Any Views. In *ICLR*, 2026. 3, 4, 5

[44] Changkun Liu, Bin Tan, Zeran Ke, Shangzhan Zhang, Jiachen Liu, Ming Qian, Nan Xue, Yujun Shen, and Tristan Braud. PLANA3R: Zero-shot Metric Planar 3D Reconstruction via Feed-Forward Planar Splatting. In *NeurIPS*. 2025. 7

[45] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. PlaneNet: Piece-wise planar reconstruction from a single RGB image. In *CVPR*, 2018. 20

[46] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. PlaneRCNN: 3D plane detection and reconstruction from a single image. In *CVPR*. 2019. 17, 20

[47] Jiachen Liu, Rui Yu, Sili Chen, Sharon X. Huang, and Hengkai Guo. Towards In-the-wild 3D Plane Reconstruction from a Single Image. In *CVPR*. 2025. 5, 20

[48] Liu Liu, Hongdong Li, and Yuchao Dai. Efficient global 2d-3d matching for camera localization in a large-scale 3d map. In *ICCV*, 2017. 1

[49] Yuzhou Liu, Lingjie Zhu, Xiaodong Ma, Hanqiao Ye, Xiang Gao, Xianwei Zheng, and Shuhan Shen. PolyRoom: Room-aware transformer for floorplan reconstruction. In *ECCV*. 2024. 1

[50] Yuzhou Liu, Lingjie Zhu, Hanqiao Ye, Yujun Liu, Shangfeng Huang, Xiang Gao, Ruisheng Wang, and Shuhan Shen. BuildingGPT: Auto-Regressive Building Wireframe Reconstruction Model with Reinforcement Learning. In *CVPR*, 2026. 1

[51] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[52] Zhixiang Min, Naji Khosravan, Zachary Bessinger, Manjunath Narayana, Sing Bing Kang, Enrique Dunn, and Ivaylo Boyadzhiev. LASER: LAtent SpacE Rendering for 2D Visual Localization. In *CVPR*. 2022. 3, 8, 9

[53] Áron Monszpart, Nicolas Mellado, Gabriel J. Brostow, and Niloy J. Mitra. RAPter: Rebuilding man-made scenes with regular arrangements of planes. *ACM Trans. Graph.*, 34(4):103:1–103:12, 2015. 3

[54] Arthur Moreau, Nathan Piasco, Moussab Bennehar, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. CROSSFIRE: Camera relocalization on self-supervised features from an implicit representation. In *ICCV*. 2023. 1

[55] Riku Murai, Eric Dexheimer, and Andrew J. Davison. MASt3R-SLAM: Real-Time Dense SLAM with 3D Reconstruction Priors. In *CVPR*. 2025. 3

[56] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *TMLR*, pp. 2835–8856, 2024. 5

[57] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. MeshLoc: Mesh-based visual localization. In *ECCV*, 2022. 1, 7

[58] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. Visual Localization using Imperfect 3D Models from the Internet. In *ICCV*. 2023. 3, 7

[59] Pilailuck Panphattarasap and Andrew Calway. Automated map reading: Image based localisation in 2-D maps using binary semantic descriptors. In *IROS*, 2018. 3

[60] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *CVPR*. 2017. 5

[61] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, Slobodan Ilic, Dewen Hu, and Kai Xu. GeoTransformer: Fast and Robust Point Cloud Registration with Geometric Transformer. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(8):9806–9821, 2023. 7

[62] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In *CVPR*. 2019. 1

[63] Paul-Edouard Sarlin, Daniel DeTone, Tsun-Yi Yang, Armen Avetisyan, Julian Straub, Tomasz Malisiewicz, Samuel Rota Bulò, Richard Newcombe, Peter Kontschieder, and Vasileios Balntas. OrienterNet: Visual localization in 2D public maps with neural matching. In *CVPR*, 2023. 1, 3

[64] Paul-Edouard Sarlin, Eduard Trulls, Marc Pollefeys, Jan Hosang, and Simon Lynen. SNAP: Self-supervised neural maps for visual positioning and semantic understanding. In *NeurIPS*, 2023. 3

[65] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *ICCV*, 2019. 17

[66] Johannes L. Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic Visual Localization. In *CVPR*. 2018. 1

[67] Jingjia Shi, Shuaifeng Zhi, and Kai Xu. PlaneRecTR: Unified Query Learning for 3D Plane Recovery from a Single View. In *ICCV*, 2023. 20

[68] Jingjia Shi, Shuaifeng Zhi, and Kai Xu. PlaneRecTR++: Unified Query Learning for Joint 3D Planar Reconstruction and Pose Estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 48(1): 962–981, 2026. 5, 7, 20

[69] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-d images. In *CVPR*, 2013. 1

[70] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3R: Zero-shot gaussian splatting from uncalibrated image pairs, *arXiv preprint arXiv:2408.13912*, 2024. Available at https://arxiv.org/abs/2408.13912. 3

[71] Michal Stary, Julien Gaubil, Ayush Tewari, and Vincent Sitzmann. Understanding Multi-View Transformers. In *ICCVW*. 2025. 5

[72] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. In *CVPR*. 2018. 1

[73] Bin Tan, Nan Xue, Tianfu Wu, and Gui-Song Xia. NOPE-SAC: Neural one-plane RANSAC for sparse-view planar 3D reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(12): 15233–15248, 2023. 7

[74] Bin Tan, Rui Yu, Yujun Shen, and Nan Xue. PlanarSplatting: Accurate Planar Surface Reconstruction in 3 Minutes. In *CVPR*. 2025. 3

[75] Julien Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In *3DV*. 2016. 7, 10

[76] Tomas Vojir, Ignas Budvytis, and Roberto Cipolla. Efficient large-scale semantic visual localization in 2D maps. In *ACCV*, 2020. 3

[77] Fangjinhua Wang, Xudong Jiang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. GLACE: Global local accelerated coordinate encoding. In *CVPR*, 2024. 1

[78] Haiping Wang, Yuan Liu, Bing Wang, Yujing Sun, Zhen Dong, Wenping Wang, and Bisheng Yang. FreeReg: Image-to-Point Cloud Registration Leveraging Pretrained Diffusion Models and Monocular Depth Estimators. In *ICLR*, 2024. 7

[79] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual Geometry Grounded Transformer. In *CVPR*. 2025. 3

[80] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3D Perception Model with Persistent State. In *CVPR*. 2025. 3

[81] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate Monocular Geometry with Metric Scale and Sharp Details. In *NeurIPS*. 2025. 20

[82] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D Vision Made Easy. In *CVPR*. 2024. 2, 3

[83] Jamie Watson, Filippo Aleotti, Mohamed Sayed, Zawar Qureshi, Oisin Mac Aodha, Gabriel Brostow, Michael Firman, and Sara Vicente. AirPlanes: Accurate plane estimation via 3D-consistent embeddings. In *CVPR*, 2024. 2

[84] Hang Wu, Zhenghao Zhang, Siyuan Lin, Xiangru Mu, Qiang Zhao, Ming Yang, and Tong Qin. MapLocNet: Coarse-to-fine feature registration for visual re-localization in navigation maps. In *IROS*, 2024. 3

[85] Matthias Wüest, Francis Engelmann, Ondrej Miksik, Marc Pollefeys, and Daniel Barath. UnLoc: Leveraging Depth Uncertainties for Floorplan Localization. In *ICLR*. 2026. 8

[86] Yiming Xie, Matheus Gadelha, Fengting Yang, Xiaowei Zhou, and Huaizu Jiang. PlanarRecon: Realtime 3D Plane Detection and Reconstruction from Posed Monocular Videos. In *CVPR*, 2022. 2, 17

[87] Fengting Yang and Zihan Zhou. Recovering 3D planes from a single image via convolutional neural networks. In *ECCV*, 2018. 5, 20

[88] Yandan Yang, Baoxiong Jia, Peiyuan Zhi, and Siyuan Huang. PhyScene: Physically interactable 3D scene synthesis for embodied AI. In *CVPR*, 2024. 1

[89] Hanqiao Ye, Yuzhou Liu, Yangdong Liu, and Shuhan Shen. NeuralPlane: Structured 3D reconstruction in planar primitives with neural fields. In *ICLR*, 2025. 3

[90] Hanqiao Ye, Yuzhou Liu, Yangdong Liu, and Shuhan Shen. PlanaReLoc: Camera Relocalization in 3D Planar Primitives via Region-Based Structure Matching. In *CVPR*. 2026. 2, 3, 5, 6, 7, 10, 17

[91] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. iNeRF: Inverting Neural Radiance Fields for Pose Estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2021. 1

[92] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *ICCV*, 2023. 5, 17

[93] Mulin Yu and Florent Lafarge. Finding good configurations of planar primitives in unorganized point clouds. In *CVPR*. 2022. 3

[94] Zehao Yu, Jia Zheng, Dongze Lian, Zihan Zhou, and Shenghua Gao. Single-image piece-wise planar 3D reconstruction via associative embedding. In *CVPR*. 2019. 5

[95] Hongjia Zhai, Xiyu Zhang, Boming Zhao, Hai Li, Yijia He, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. SplatLoc: 3D Gaussian Splatting-based Visual Localization for Augmented Reality. *IEEE Trans. Vis. Comput. Graph.*, 2024. 1

[96] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. MonST3R: A Simple Approach for Estimating Geometry in the Presence of Motion. In *ICLR*. 2025. 3

[97] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3D: A Large Photo-realistic Dataset for Structured 3D Modeling. In *ECCV*. 2020. 1, 2, 6, 8, 17

[98] Ping Zhong, Shiyong Meng, Tao Zou, Bolei Chen, Chaoxu Mu, and Jianxin Wang. DisCo-floc: Using dual-level visual-geometric contrasts to disambiguate depth-aware visual floorplan localization, 2026. Available at https://arxiv.org/abs/2601.01822. 8

[99] Juelin Zhu, Shen Yan, Long Wang, zhang shengYue, Yu Liu, and Maojun Zhang. LoD-loc: Aerial visual localization using LoD 3D map with neural wireframe alignment. In *NeurIPS*. 2024. 3

[100] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming visual geometry transformer. In *ICLR*, 2026. 3

A Appendix

In the appendix, we provide the following:

- Details on data preparation (Sec. A.1) and a brief summary of dataset statistics (Tab. 8).
- Implementation details on data augmentations, training losses, and plane recovery (Sec. A.2).
- Ablation of pathway-specific supervision and its comparison with full multi-task supervision (Tab. 9).
- Evaluations on ScanNet and 12Scenes using cumulative accuracy curves to account for all thresholds (Fig. 7).
- Analysis on how the number of map primitives observed affects localization (Fig. 8).
- Additional evaluations on plane recovery (Tab. 10 and Fig. 9).
- Additional evaluation results on layout localization (Tab. 11).
- Analysis on how localization performance varies with parameter estimation quality (Fig. 10).
- More qualitative results on surface maps (Fig. 11) and layout models (Fig. 12).
- Discussions on failure cases (Figs. 13 and 14), limitations, and future work.

A.1 Datasets

Our model is trained on two independent sets of data, targeting localization on the dense surface map and the minimal layout model, respectively. Table 8 summarizes the statistics of these training datasets, and we provide detailed descriptions in the following.

Dense Surface Map. For surface maps, we inherit the data processing pipeline from PlanaReLoc [90], expanding the training set to three public indoor datasets. Specifically, maps are extracted from the officially provided dense reconstructions using the sequential RANSAC [23] plane-fitting scripts from [46, 86]; the extracted maps are further compressed with polygon simplification algorithms [26]. Training images of the real-world datasets are sampled from the original RGB-D sequences at regular intervals, and each sampled frame will be checked for adequate primitive observations and precise pose alignment. MP3D-Habitat is the only synthetic dataset curated from Matterport3D by [37] whose training images are generated using Habitat [65]. The resolution of training images is standardized to 640×480. Finally, per-pixel map primitive IDs are rendered onto training images to obtain the dense pixel-primitive associations, which can further be leveraged to calculate the ground-truth per-pixel parameter map $\boldsymbol{P}$ or downsampled into the patch-primitive assignment matrix $\boldsymbol{A}$ for supervision.

Table 8: **Statistics of training datasets.** Two groups of datasets are used to train the model for localization on surface maps and layout models, respectively.

Dataset	Map Type	Domain	Camera Type	#Frames	#Scenes (regions)	Avg.#Frames per Scene
ScanNet [20]	Surface	Real-world	Perspective	98 533	1210	81
ScanNet++ [92]		Real-world	Perspective	216 645	856	253
Matterport3D [12]		Real-world	Perspective	54 578	2195	25
MP3D-Habitat [65]		Synthetic	Perspective	24 568	2195	11
S3D-*official* [97]	Layout	Synthetic	Perspective	148 341	3250	46
S3D-*resampled* [25]		Synthetic	Panoramic	21 835	3250	7

Minimal Layout Model. S3D-*official* and S3D-*resampled* share the same collection of layout models, but differ in the query images used for training and evaluation. S3D-*official* comprises officially provided high-resolution perspective images in a format of 16:9 with a fixed FoV of 80°, which are resized to the resolution of 640×360 in our experiments. On the other hand, the S3D-*resampled* dataset proposed by [25] starts from the original panorama images and samples training perspective images on the fly with random camera orientations (yaw ∈ [-180°, 180°], pitch ∈ [-10°, 10°] and roll ∈ [-10°, 10°]) as well as random FoVs from 45° to 135°. These perspective images

are sampled at a resolution of 320×320. The sampled images will be verified for observations of at least three semantic classes (*e.g.*, wall-ceiling-floor and wall-window-floor) and adequate coverage of the room interior (*i.e.*, doors and windows not occupying more than 70 % of the image). For efficiency, we generate at least 100 verified candidates of yaw, pitch, roll and FoV for each panorama when preparing the dataset, and during training, images are directly sampled from these candidates, avoiding the redundant computation for verification. Likewise, annotations of per-pixel primitive IDs are rendered onto the original panorama, and then can be sampled to generate perspective images for training on the fly.

A.2 Implementation Details

Input Augmentations. During training, we apply brightness, color and contrast jittering, and randomly rescale the query images to heights between 384 and 600 pixels before cropping or padding them to the final resolution of 640×480. The maps are normalized into a unit box and randomly rescaled and rotated by a random yaw angle, with map primitives shuffled before being fed into the model. For samples from Matterport3D, maps are assembled online from submaps of multiple regions according to the query images. Specifically, we merge the submaps corresponding to regions observed by the query image together with those of spatially nearby regions, resulting in dynamic map coverage.

Training Losses. We elaborate on the loss terms introduced in Eq. (2). Given the ground-truth patch-primitive assignment matrix $\boldsymbol{A}$, we denote the set of positive patch-primitive pairs as $\mathcal{A}^* = \{(i,j)|\boldsymbol{A}_{ij} = 1\}$ and unassignable patch and primitive sets as $\mathcal{U}^q = \{i|\sum_j \boldsymbol{A}_{ij} = 0\}$ and $\mathcal{U}^m = \{j|\sum_i \boldsymbol{A}_{ij} = 0\}$, respectively. Then, the assignment loss is defined as the negative log-likelihood of the soft assignment matrix $\tilde{\boldsymbol{A}}$ and unassignable predictions:

$$\mathcal{L}_{\text{assign}} = -\left(\frac{1}{|\mathcal{A}^*|} \sum_{(i,j)\in\mathcal{A}^*} \log \tilde{\boldsymbol{A}}_{ij} + \frac{1}{2|\mathcal{U}^q|} \sum_{i\in\mathcal{U}^q} \log(1-\sigma_i^q) + \frac{1}{2|\mathcal{U}^m|} \sum_{j\in\mathcal{U}^m} \log(1-\sigma_j^m) \right). \quad (3)$$

For geometric supervision on primitive parameters, $\mathcal{L}_{\text{param}}$ is the combination of the L^1 loss and the cosine similarity loss $L_{\cos}$, computed on the per-pixel parameter prediction $\hat{\boldsymbol{P}}^q \in \mathbb{R}^{3\times H\times W}$ of the query image as well as the per-primitive prediction $\hat{\boldsymbol{P}}^m \in \mathbb{R}^{3\times l_m}$ of the map:

$$\mathcal{L}_{\text{param}} = \sum_{r\in\{q,m\}} \left(L^1(\hat{\boldsymbol{P}}^r, \boldsymbol{P}^r) + L_{\cos}(\hat{\boldsymbol{P}}^r, \boldsymbol{P}^r) \right). \quad (4)$$

The ray map loss $\mathcal{L}_{\text{ray}}$ and the pose decoding loss $\mathcal{L}_{\text{pose}}$ are both based on the L^1 norm:

$$\mathcal{L}_{\text{ray}} = \boldsymbol{R}_u \odot L^1(\hat{\boldsymbol{R}}, \boldsymbol{R}) - \lambda_u \log \boldsymbol{R}_u, \quad (5)$$

$$\mathcal{L}_{\text{pose}} = L^1(\hat{\mathbf{g}}^q, \mathbf{g}^q) + L^1(\hat{\mathbf{g}}^m, \mathbf{g}^m). \quad (6)$$

Here, $\boldsymbol{R}_u$ is the ray uncertainty map and $\odot$ is the channel-broadcast element-wise product. The target pose $\mathbf{g}^m$ for the map token is set to the identity pose with the rotation quaternion $\mathbf{q}_0 = [0,0,0,1]$ and translation $\mathbf{t}_0 = [0,0,0]$.

Plane Recovery. Recovering piecewise planar primitives from the query image is not the main focus of this work, since camera poses can be directly derived from the predicted ray map. Nevertheless, plane recovery is a geometric by-product of our method, and we briefly describe how its results are obtained for visualizations (as shown in Figs. 4 and 5) and the subsequent analysis in Sec. A.3. The task of plane recovery requires both 2D plane segments and their corresponding plane parameters. The 2D plane segments are first obtained by partitioning the predicted parameter map $\hat{\boldsymbol{P}}^q \in \mathbb{R}^{3\times H\times W}$ according to the predicted patch-primitive assignment matrix $\hat{\boldsymbol{A}}$. Since the network does not enforce local consistency, *i.e.*, densely predicted per-pixel parameters within each plane segment are not guaranteed to be consistent, the final plane parameters for each segment are robustly estimated with RANSAC.

A.3 Additional Evaluation and Analysis

Pathway-specific Supervision. The ablation in Tab. 5 evaluates auxiliary prediction tasks while keeping Ray-based inference fixed; removing $\mathcal{L}_{\text{ray}}$ and switching inference pathways would change

both supervision and inference. To assess each pathway separately, we train models using only the supervision required by their respective inference pathways and compare them with the full model on ScanNet in Tab. 9. All three pathways independently enable localization, with Ray-based inference performing best overall among the individually supervised variants. Full multi-task supervision further improves Ray-based localization, demonstrating the complementary benefits of the prediction tasks.

Table 9: **Pathway-specific supervision** on ScanNet. Each pathway is trained with its required supervision only; Full (Ray) uses all supervision terms.

		Med. Err. ↓		Pose Recall (%) ↑		
Pathway	Supervision	$\Delta\mathbf{q}$ (°)	$\Delta\mathbf{t}$ (m)	(0.2 m, 10°)	(0.5 m, 15°)	(1.0 m, 30°)
Param.-based	$\mathcal{L}_{\text{assign}} + \mathcal{L}_{\text{param}}$	3.1	0.23	44.2	70.6	79.5
Token-based	$\mathcal{L}_{\text{pose}}$	4.4	0.28	39.9	83.7	87.2
Ray-based	$\mathcal{L}_{\text{ray}}$	3.4	0.20	51.7	87.6	90.9
Full (Ray)	$\mathcal{L}_{\text{assign}} + \mathcal{L}_{\text{param}} + \mathcal{L}_{\text{pose}} + \mathcal{L}_{\text{ray}}$	**3.0**	**0.16**	**62.0**	**92.8**	**95.2**

Cumulative Accuracy Curves. As shown in Fig. 7, the cumulative accuracy curves illustrate the proportion of camera poses whose translation and rotation errors fall below varying thresholds from 0 m to 1 m and from 0° to 30°. Serving as a more detailed performance comparison, the curves on both the ScanNet and 12Scenes datasets highlight the superior robustness and accuracy of our method across both translation and rotation.

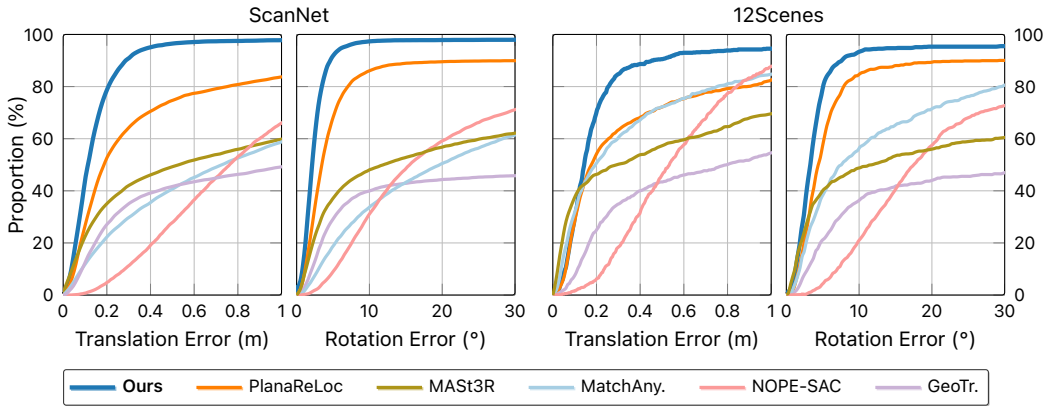
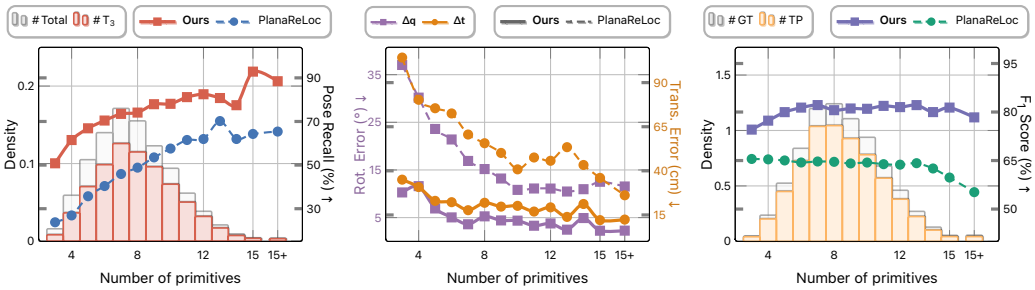


Figure 7: **Cumulative accuracy curves** on surface maps. We plot the proportion of successfully localized frames as a function of varying translation and rotation error thresholds.

Impact of Primitive Richness. In practice, camera localization becomes unreliable when query observations are ambiguous and uninformative, *e.g.*, when only a few map primitives are observed. To analyze the impact of primitive richness on localization performance, we plot the localization performance as a function of the number of actually observed primitives in Fig. 8. As shown in Fig. 8a and b, observing more primitives generally improves localization accuracy. Notably, compared to the feature-matching method PlanaReLoc, which depends on explicit primitive correspondences for geometric pose estimation and can become unstable when few primitives are observed, our method maintains robust performance, yielding reliable camera poses even under sparse observations. Moreover, as shown in Fig. 8c, when the query observation becomes cluttered with more primitives, there is a notable drop in the matching F_1 score of PlanaReLoc due to its degraded plane recovery performance as planes in richer observations tend to be less salient and harder to recover. In contrast, our method performs exhaustive interaction between image patches and map primitives during the single feed-forward pass, allowing the model to finely ground the query observations to the map primitives, and thereby exhibiting a more stable matching performance.

Evaluation on Plane Recovery. While poses can be derived directly from the predicted ray maps and plane recovery is thus not the primary focus of this work, we also use our plane recovery results



(a) Pose recall. (b) Average pose error. (c) Matching performance.

Figure 8: **Analysis of the impact of primitive richness** on ScanNet. (a) *Left:* Distribution of all test samples and successfully localized samples (by our method) at (0.2 m, 10°) thresholds w.r.t. the number of actually observed primitives. *Right:* Pose recall at (0.2 m, 10°) versus the number of actually observed primitives. (b) Average pose error curves, where pose errors are averaged over all samples within each bin. (c) *Left:* Distribution of actually observed map primitives and those successfully matched (by our method). *Right:* Curves of the matching F_1 score.

Table 10: **Comparison of plane recovery results** on our ScanNet test set. Although existing monocular methods are not direct competitors, we quantitatively compare their plane recovery quality with ours using standard segmentation and depth estimation metrics. The **best** and *second* are highlighted.

	Segmentation			Depth Estimation					
Methods	RI ↑	VOI ↓	SC ↑	RMSE ↓	Log_{10} ↓	Rel ↓	σ_1 ↑	σ_2 ↑	σ_3 ↑
PlaneRecTR [68]	0.91	0.93	0.78	*0.24*	*0.05*	*0.09*	*90.8*	*97.7*	*99.2*
ZeroPlane [47]	*0.92*	*0.90*	*0.80*	0.34	0.09	0.18	65.9	92.1	97.7
MoGe-2 [81]+RANSAC	0.89	1.28	0.71	0.65	0.58	0.20	82.3	86.0	86.5
Ours	**0.94**	**0.78**	**0.86**	**0.16**	**0.03**	**0.07**	**96.0**	**99.1**	**99.7**

to evaluate the predictions on primitive parameters enabled by the corresponding head in our model, to assess the quality of geometry learning. Following the standard metrics in the literature [46, 87], the performance of plane recovery is evaluated from two perspectives: the segmentation quality of 2D plane instances and the accuracy of the depth estimation from the predicted plane parameters. We compare our method with two end-to-end monocular plane recovery methods [47, 67], as well as the method used in PlanaReLoc on our ScanNet test set, and the results are summarized in Tab. 10. A more comprehensive evaluation using per-pixel and per-plane recalls [45] is shown in Fig. 9. Although not trained with explicit supervision on plane recovery, by reasoning jointly over map primitives, our method significantly outperforms the monocular methods.

Additional Evaluation for Layout Localization. Since existing layout-based localization methods mostly operate in 2D, we primarily report 2D localization errors in the main text for comparison. As our method performs 6-DoF camera localization using a 3D map, we report the 3D median localization errors in Tab. 11. Furthermore, we evaluate the predicted layout geometry with depth estimation metrics and assess segmentation results using the standard mIoU and mACC metrics.

Table 11: **Additional Evaluation for Layout Localization.**

	Med. Err. ↓ (3D)		Depth Estimation						Segmentation ↑	
Dataset	$\Delta\mathbf{q}$ (°)	$\Delta\mathbf{t}$ (cm)	RMSE ↓	Log_{10} ↓	Rel ↓	σ_1 ↑	σ_2 ↑	σ_3 ↑	mIoU	mACC
official	0.75	13.0	0.18	0.03	0.08	96.1	98.6	99.1	81.6	84.9
resampled-90° FoV	0.81	11.1	0.11	0.02	0.05	98.3	99.6	99.8	78.3	80.1

Relationship between Parameter and Pose Estimation. We are interested in finding out how the pose estimation accuracy varies with parameter estimation quality, as in PlanaReLoc, the pose estimation is based on the predicted plane parameters and can be significantly affected by the parameter estimation quality. Therefore, we plot the distribution of depth RMSE together with the

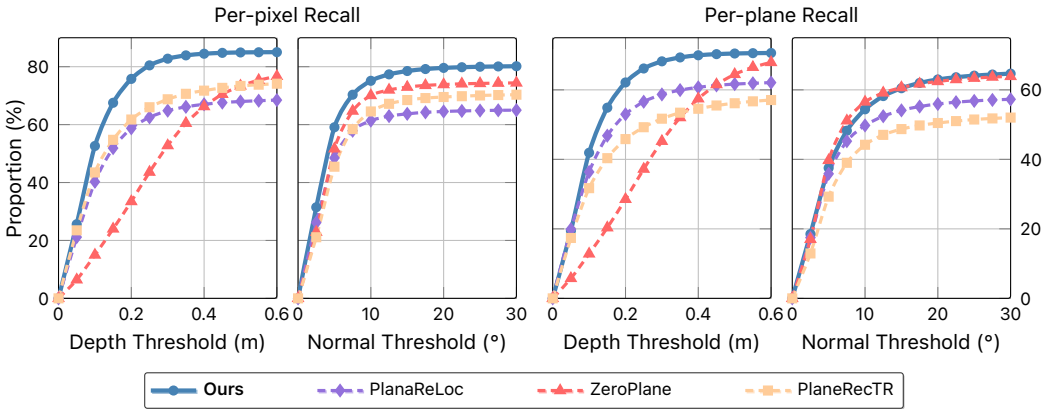
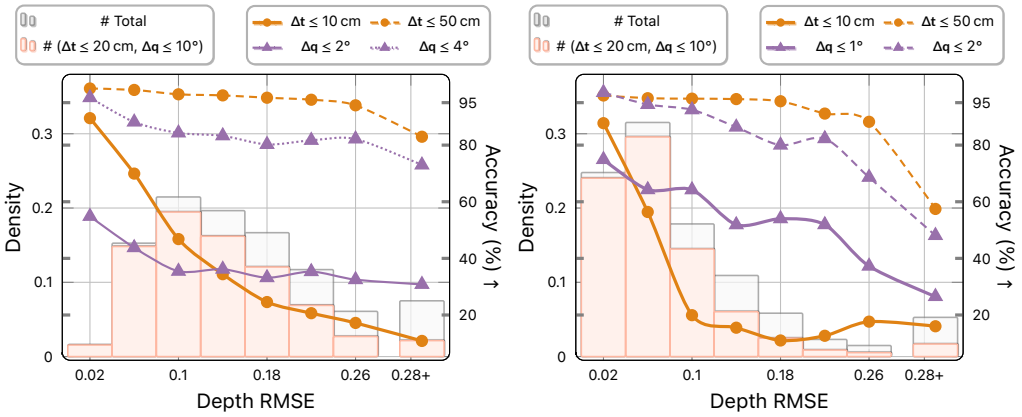


Figure 9: **Per-pixel and per-plane recalls** on our ScanNet test set, which are comprehensive metrics for evaluating the plane recovery quality.



(a) ScanNet.

(b) S3D-*resampled*-90° FoV.

Figure 10: **Performance trends of depth and pose estimation** on two datasets. We plot the depth RMSE distribution as histograms (left y-axis) and corresponding pose accuracy curves (right y-axis).

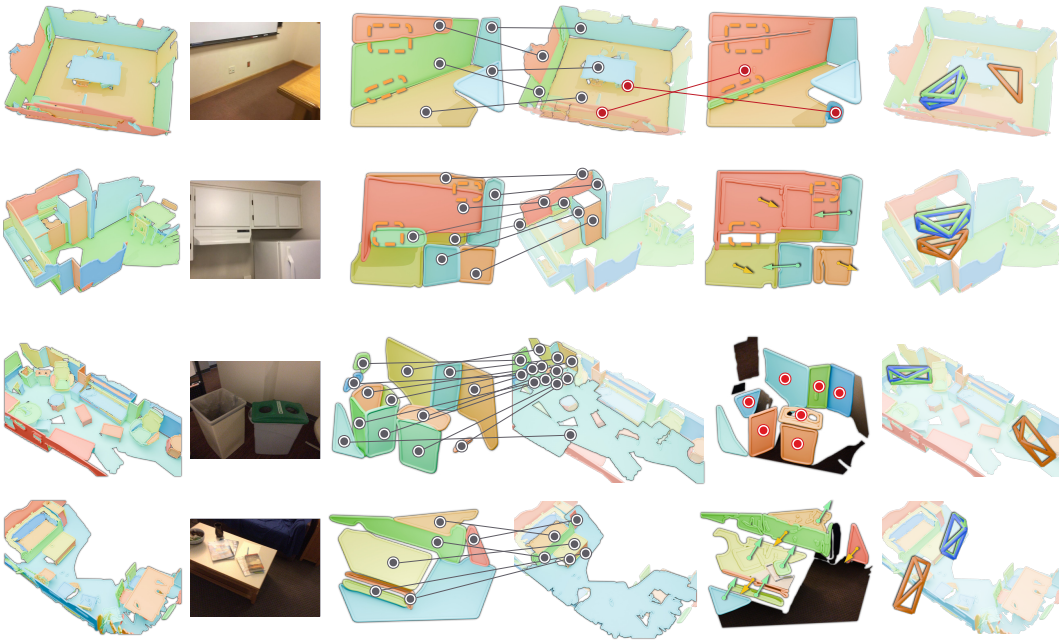
corresponding pose accuracy curves on both the ScanNet and S3D-*resampled* datasets in Fig. 10. It can also be observed that for our method, the pose accuracy generally drops as depth RMSE increases. However, we note that accurate depth estimation does not necessarily indicate successful localization (*e.g.*, due to ambiguous repetitive patterns), as discussed later in Sec. A.4.

Additional Qualitative Results. More visualizations of surface map localization results as well as layout localization results are provided in Fig. 11 and Fig. 12, respectively. All results in this work are visualized using Blender [7].

A.4 Limitations and Future Work

Failure Cases. We show representative failure cases of our method on both surface maps and layout models in Fig. 13 and Fig. 14, respectively. In some cases (Fig. 13a, b, and Fig. 14a, b), poses derived from the ray map are less accurate than those estimated using predicted primitive parameters. We also observe that the failure cases mainly arise from three aspects: (1) challenging visual conditions, such as motion blur and low lighting; (2) limited structural details or highly repetitive patterns in the scene; and (3) uninformative query observations.

Limitations. First, although our primitive-grounded localization paradigm naturally extends from dense surface maps to minimal layout models, the network is trained separately for each task. Then, our work is currently limited to maps that are assembled from planar primitives, which assume the presence of planar surfaces and regularized layouts, and thus may not be suitable for unstructured environments or scenes exhibiting complex or non-planar structures. Meanwhile, our model is



(a) Input: map & query (b) Plane recovery & grounded primitives (c) PlanaReLoc (d) Localization

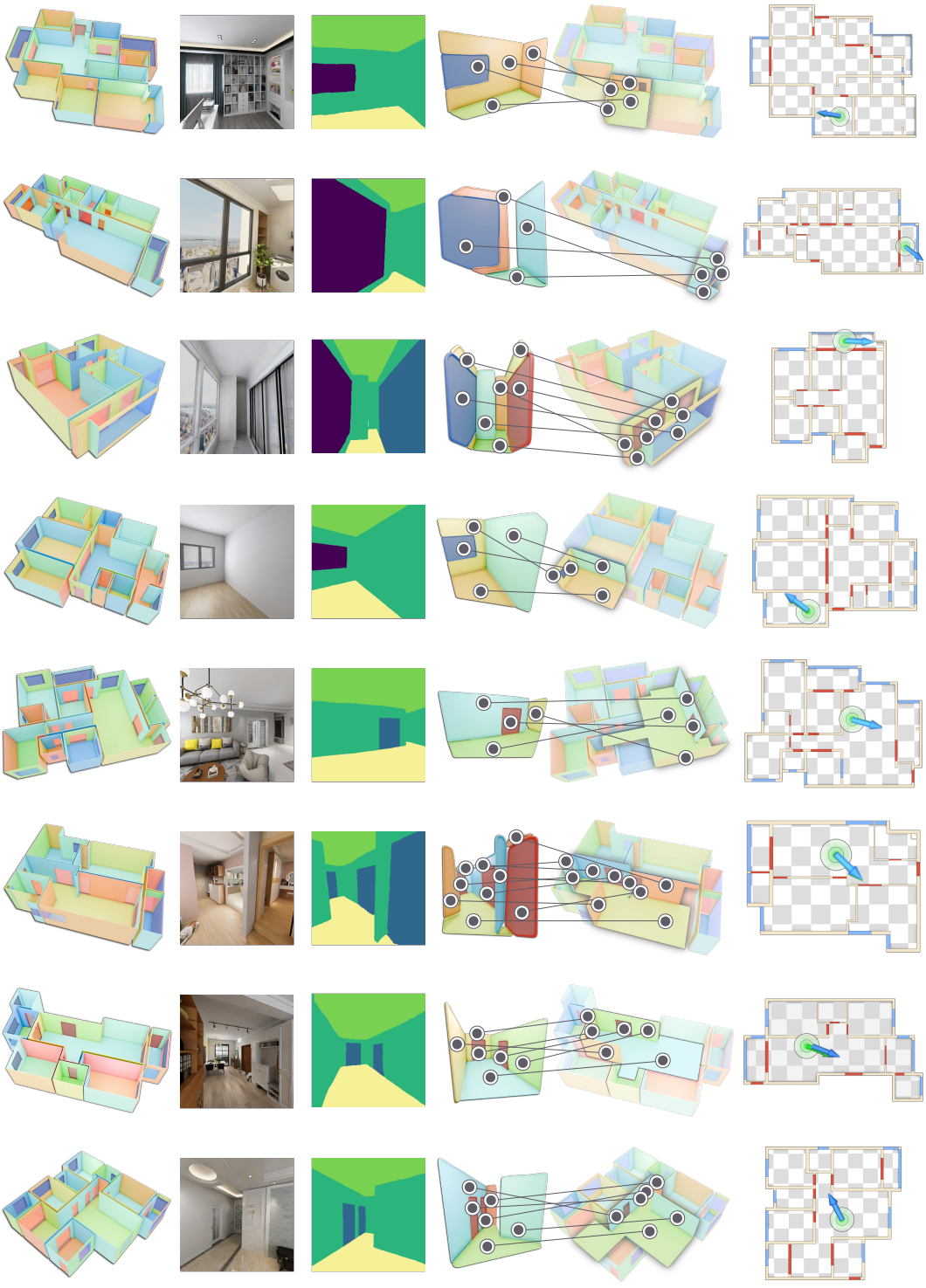
Figure 11: **Qualitative results on surface maps.** The first two examples are from ScanNet and the last two examples are from 12Scenes. Our method robustly produces accurate pose estimation, with (b) superior plane recovery and primitive correspondences as by-products, while (c) PlanaReLoc performs suboptimally or even fails due to poor plane recovery, matching outliers and inadequate pose estimation constraints. (d) Comparison of localization results. Legend: the ground truth, PGGT(Ours), and PlanaReLoc.

trained primarily on room-scale indoor environments. Although it outperforms PlanaReLoc on artificially integrated multi-room maps without fine-tuning (Fig. 6), recall decreases substantially as map size grows, highlighting the limitations of full-map inference. These experiments do not establish performance in real large-scale buildings or outdoor environments, where the distribution of map primitives may differ significantly.

Future Work. The proposed primitive-grounded localization paradigm can potentially extend to a variety of high-level map representations in the future, such as LoD city models and structured language commands [5]. This is enabled by the flexibility and scalability of our feed-forward localization framework, which allows it to accommodate diverse map representations and even unify localization across different map modalities within a single model. In addition, extending the method to panoramic or sequential inputs offers another promising direction for practical application, as richer visual cues and temporal context could effectively help resolve structural ambiguities.

A.5 Societal Impacts

Our work focuses on the development of a novel camera localization method that leverages high-level map primitives under a unified and scalable framework, which is highly suitable for memory-constrained and real-time applications such as AR/VR and robotics. However, with the rapid growth in the popularity of cloud-based localization services, potential privacy concerns arise during localization, as users' scene-related information may be exposed to unauthorized access or misuse. One possible mitigation is to anonymize sensitive scene data on-device, for instance by tokenizing the query image and the map prior to transmission.



(a) Input: map & query (b) Segmentation & plane recovery & grounded primitives (c) Localization

Figure 12: **Qualitative results on layout models.** (b) We present the predicted layout segmentation, the layout geometry recovered from the query image, and the grounded map primitives, with ceilings removed for clarity. (c) We show the localization result as well as the ground truth.

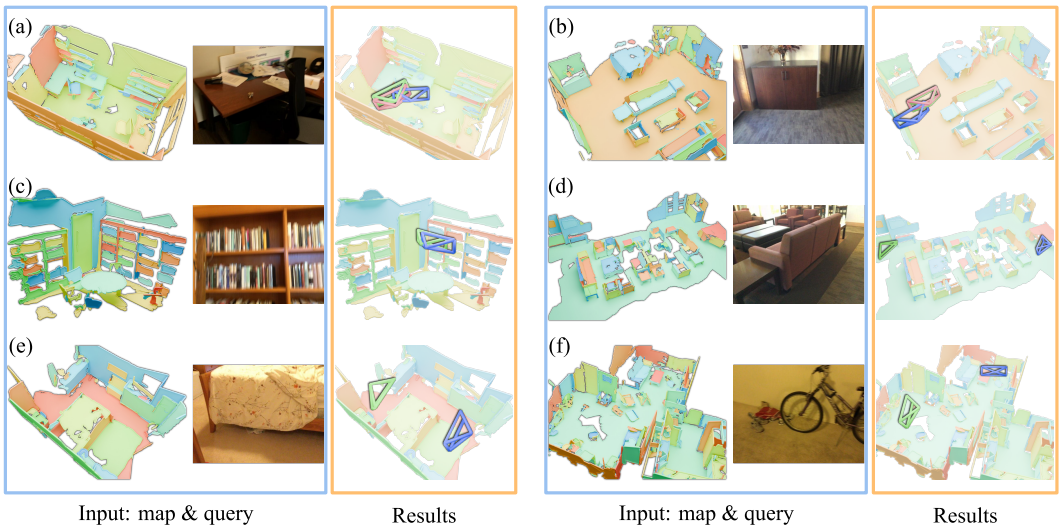


Figure 13: **Representative failure cases on surface maps.** (a–b) Cases where the Param.-based pose is more accurate than the Ray-based pose. (c) Pose shift under blurry observations. (d–f) Failure cases occurring in scenes with limited structural detail or highly repetitive patterns, or caused by uninformative query observations.

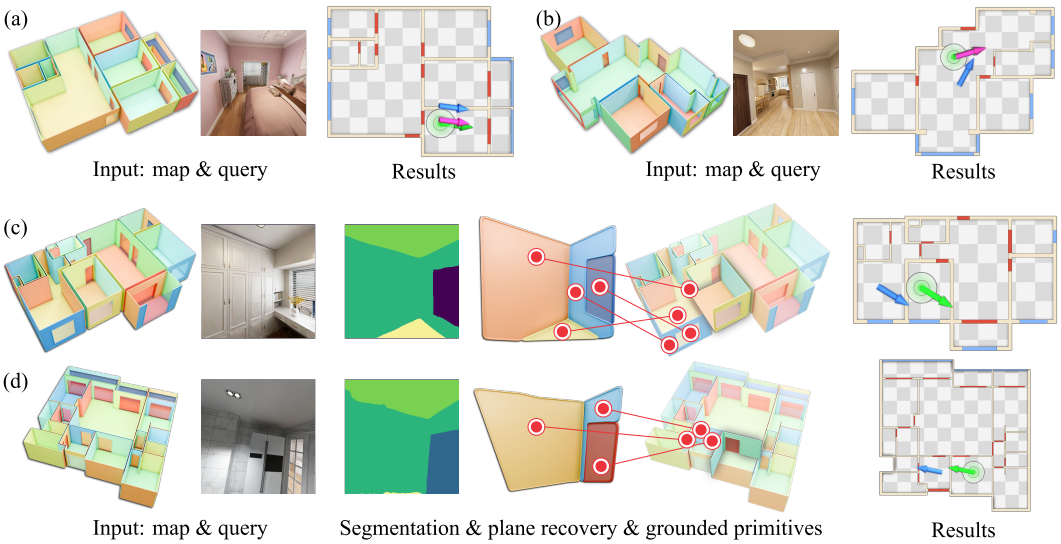


Figure 14: **Representative failure cases on layout models.** (a–b) Cases where the Param.-based pose is more accurate than the Ray-based pose. (c–d) Failure cases due to structural ambiguity.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main contributions are clearly stated and well supported by the experiments. The scope of this paper is localization using planar primitives, which is consistently reflected in the abstract and introduction, with the aspirational goal of extending the proposed primitive-grounded localization paradigm to more diverse map representations in the future.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide a detailed discussion of the limitations of our method in Sec. A.4, including representative failure cases and their causes, as well as the scope of applicability of our method and potential directions for future work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include theoretical results or proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Our main contribution is primitive-grounded localization, which directly associates visual observations with compact map primitives for feed-forward localization in unseen scenes, instantiated by the model described in Sec. 3.2. In Sec. 3.3, we provide key descriptions on training settings and architectural details of our base model. In Sec. 4.2, we further detail the training settings for the extension to layout localization. Empirical evaluation metrics and protocols are described in Sec. 5.1 and Sec. 5.2. All experiments are conducted on open-source datasets and evaluated on established public benchmarks.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code and models are publicly available at https://github.com/3dv-casia/PGGT-Loc, with instructions for data preparation, training, and evaluation to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All necessary details for training can be found in Secs. 3.3 and 4.2, and details for testing can be found in Secs. 5.1 and 5.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The experiments are established benchmark evaluations on fixed test sets with deterministic metrics, for which reporting error bars or statistical significance measures is not common practice.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Details on the computational cost of training our model can be found in the implementation details in Sec. 3.3. We also report the average inference time per query image in Tab. 1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research complies with the NeurIPS Code of Ethics in all aspects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We provide a discussion on societal impacts of our work in Sec. A.5.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The work does not involve releasing models or datasets that pose a high risk of misuse. The experiments use publicly available datasets and the model performs a standard computer vision task (camera localization).

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets and codebases used in the paper are properly cited with their official papers and licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code and trained models are publicly available at `https://github.com/3dv-casia/PGGT-Loc`, along with documentation for data preparation, training, and evaluation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve research with human subjects and therefore does not require IRB approval.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were not used as part of the core methodology or experiments. Any LLM usage was limited to writing assistance and did not impact the scientific content.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.