SceneAligner: 3D-Grounded Floorplan Localization in the Wild

Junhyeong Cho[1] **Ruojin Cai**[2] **Hadar Averbuch-Elor**[1]
[1]Cornell University [2]Kempner Institute, Harvard University
https://Cornell-VAILab.github.io/SceneAligner



Figure 1: Given a collection of in-the-wild images and a rasterized floorplan, *SceneAligner* reconstructs a gravity-aligned 3D point cloud from the images and globally aligns this reconstruction to the 2D floorplan map, thereby localizing the images within the floorplan. As illustrated above, our approach successfully aligns images capturing large-scale 3D environments, including exterior scenes (Doddabasappa Temple, left) and interior spaces (Église Saint-Martin d'Agonac, right).

Abstract

Many public buildings provide floorplans with a "you are here" indicator to help visitors orient themselves. Floorplan localization seeks to computationally replicate this capability by determining where visual observations were captured within a floorplan. However, existing methods typically assume controlled small-scale environments and precise vectorized floorplans, limiting their ability to operate in large-scale buildings and rasterized floorplans. In this work, we present an approach for performing floorplan localization in the wild by grounding the task in a reconstructed 3D representation of the scene. Given an unconstrained image collection, our method reconstructs a gravity-aligned 3D scene and projects it into a 2D density map that serves as a floorplan proxy. Floorplan localization is then formulated as aligning this proxy with the input floorplan via a 2D similarity transform. To bridge the appearance gap between density maps and architectural floorplans, we adapt a 2D foundation model to learn cross-modal correspondences, introducing a fine-tuning scheme that encourages semantically aligned matches while preserving structural consistency. Extensive experiments demonstrate substantial improvements over prior methods, including in extremely sparse settings with as little as a single input image. Our code and data are publicly available.

1 Introduction

Localizing camera observations within a provided 2D floorplan map is a fundamental task in 3D scene understanding, with applications in navigation, robotics, and augmented reality. Prior approaches [32, 6, 13] typically address this problem by exhaustively searching a discretized pose space, scoring candidate camera locations and orientations based on their consistency with a floorplan. This strategy inherently relies on access to precise, vectorized floorplans that encode fine-grained architectural primitives, such as exact wall layouts and openings. While effective in small-scale environments, these assumptions rapidly break down in large-scale, real-world settings, particularly in historic landmarks

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

and monuments, where floorplans are often available only as rasterized or symbolic drawings and where architectural complexity far exceeds that of existing carefully-curated benchmarks. This raises a natural question: how can camera observations be localized within a floorplan in the *wild*, when precise geometry is unavailable and exhaustive pose enumeration is no longer viable?

The recent emergence of 3D foundation models [45, 49, 25] has enabled accurate reconstruction of scene geometry directly from unconstrained image collections, even for large-scale environments captured under diverse viewpoints and illumination conditions. Given access to such high-fidelity geometric reconstructions, we argue that floorplan localization should be revisited from a fundamentally different perspective. To this end, we introduce SceneAligner (Figure 1), which reinterprets floorplan localization as a reconstruction and alignment problem. Rather than exhaustively enumerating and scoring camera poses over a discretized grid, our approach extracts a floorplan proxy from a reconstruction of the 3D scene, building upon prior floorplan reconstruction methods [54, 7, 24, 26] that recover 2D layouts from input 3D scans. Localization then reduces to globally aligning this proxy to the input floorplan via a 2D similarity transform. Specifically, we derive this proxy by orthographically projecting a gravity-aligned 3D reconstruction into a 2D density map.

To align this density map representation with the provided floorplan, we propose a feature matching learning scheme that estimates reliable cross-modal correspondences between the two modalities. While the density map provides a structurally-grounded proxy of the building layout, it differs significantly in appearance from architectural floorplans. To bridge this gap, we adapt a 2D foundation model (*i.e.*, DINOv3 [40]) to learn a shared feature space, introducing fine-tuning objectives that encourage semantically aligned cross-modal matches while preserving structural consistency. During inference, we extract a subset of reliable correspondences and estimate a 2D similarity transform that aligns the reconstructed 3D scene with the input floorplan, thereby enabling floorplan localization from unconstrained image collections.

We conduct extensive experiments comparing our approach against prior methods under both in-the-wild environments [17] and synthetic indoor settings [55]. Our evaluation shows that SceneAligner achieves substantial performance improvements by factors ranging from two to three across most metrics on the in-the-wild testbed, while also outperforming indoor localization methods that rely on a discretized pose space. We further show that our approach remains effective for sparse image collections, surpassing baselines even when provided with a *single* input view. Finally, we showcase the broader applicability of our approach by demonstrating that it enables the alignment of disjoint interior and exterior 3D reconstructions through registration to a shared floorplan.

2 Related Work

Floorplan Localization. Floorplan localization has been widely studied for indoor scene understanding, reconstruction, and navigation [3, 46, 6, 13, 53]. Early methods rely on depth-based cues from LiDAR [3, 4, 48, 21] or depth cameras [18], often comparing extracted room edges to the 2D floorplan layouts while assuming known camera heights [5, 8]. More recent approaches embed images and floorplans into a shared feature space [15, 14], or predict depth rays and probability volumes over the floorplan [6, 53]. To improve alignment and reduce ambiguities, several works incorporate semantic cues such as scene texts [46], CNN-extracted labels [30], pre-computed 3D maps [19], or estimated semantic volumetric probabilities [13]. However, these methods cannot address the challenge of localizing in-the-wild camera observations, where floorplans may be rasterized or symbolic drawings and input images come from unconstrained photo collections.

Learning Cross-Modal Correspondences. Establishing correspondences is a fundamental problem in computer vision, underpinning tasks such as 3D reconstruction. Early methods rely on handcrafted descriptors [28, 2, 35] with geometric verification [12], while recent approaches learn keypoints and matchers using deep visual features [11, 23, 37], enabling dense prediction (*e.g.*, LoFTR [42], DUSt3R [47]). More recently, diffusion features [43] and self-supervised representations [40] have shown remarkable potential for correspondence estimation even across different visual domains [31]. Nevertheless, matching natural images to symbolic representations remains challenging. C3Po [17] learns correspondences between perspective photographs and symbolic 2D floorplans, but the extreme viewpoint and modality gap make this cross-modal matching highly under-constrained, as it requires reasoning about the underlying 3D geometry connecting perspective views to top-down layouts.

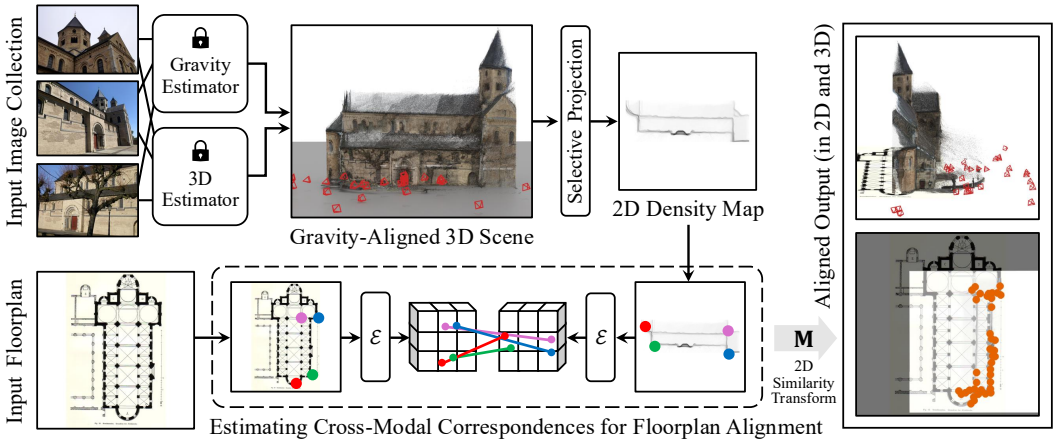


Figure 2: **SceneAligner.** Given in-the-wild images and a floorplan, it reconstructs a gravity-aligned 3D scene, extracts a 2D density map via projection, and solves for a 2D similarity transform $\mathbf{M}$ via correspondence estimation between the density map and floorplan using a shared encoder $\mathcal{E}$. Reliable correspondences used to compute $\mathbf{M}$ are overlaid (in orange) on the aligned density map.

In contrast, our method bridges this gap via an intermediate density map derived from 3D scene reconstruction, naturally connecting unconstrained photographs to abstract floorplans.

3D Scene Understanding from Internet Photo Collections. Prior work has explored spatial understanding of large-scale environments by analyzing visual patterns, viewpoints, or metadata [50, 39, 38, 36, 52, 20]. Notable efforts include assembling disjoint 3D indoor reconstructions using annotated maps and crowd flow [29] or partial RGB-D scans using floorplan priors [51, 22], aligning interior and exterior 3D scenes via scene semantics [9], and registering photo collections to a 3D reference model [10]. However, prior work cannot directly align unconstrained photos with floorplans. Recent advances in 3D foundation models [45, 49, 25] and gravity estimator [44] make gravity-aligned reconstruction feasible, which we leverage for 3D-grounded floorplan localization.

3 Method

As illustrated in Figure 2, our method recovers a gravity-aligned 3D scene (Sec. 3.1), derives a floorplan proxy (Sec. 3.2), and predicts a similarity transform for floorplan alignment by estimating correspondences between the floorplan and proxy (Sec. 3.3). We describe each step below.

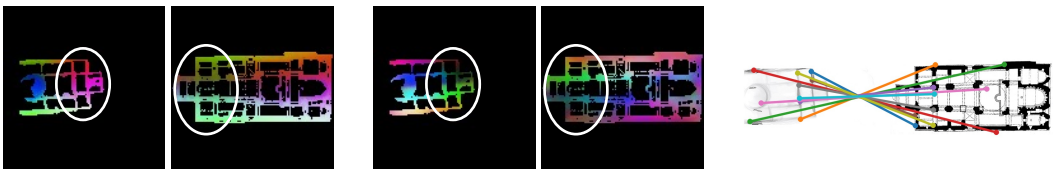
3.1 Reconstructing a Gravity-Aligned 3D Scene

Formulating floorplan localization as a 3D reconstruction and alignment problem, we first recover the scene geometry and camera poses from an unconstrained image collection $\mathcal{I} = \{I_1, \ldots, I_N\}$, where each image I_i has a resolution of $H_i \times W_i$. We leverage a 3D foundation model (*e.g.*, π^3 [49], VGGT [45]) to estimate 3D points $\mathcal{P}_i^{\mathrm{c}} \in \mathbb{R}^{H_i W_i \times 3}$ in a camera coordinate frame, along with relative camera poses that map each camera frame to the reference frame of I_1.

To align the 3D scene geometry with the physical ground plane, we predict a gravity direction per image using GeoCalib [44], transform each gravity vector into the reference frame using the corresponding relative camera pose, and select their medoid as a robust gravity estimate $\mathbf{g}$. We then compute a rigid transformation via Gram-Schmidt orthogonalization to align $\mathbf{g}$ with the vertical y-axis. By applying this transformation, we obtain gravity-aligned 3D points $\mathcal{P}_i^{\mathrm{g}} \in \mathbb{R}^{H_i W_i \times 3}$. This ensures that the reconstructed 3D geometry and the input floorplan share a common horizontal plane.

3.2 Extracting a 2D Density Map as a Floorplan Proxy

With the gravity-aligned 3D scene, we can extract a 2D density map via orthographic projection. However, directly projecting all points $\mathcal{P}_*^{\mathrm{g}} = \bigcup_{i=1}^{N} \mathcal{P}_i^{\mathrm{g}}$ is vulnerable to outliers such as faraway



Pretrained Model | Fine-tuned Model (Ours) | Ground-Truth Correspondences

Figure 3: **Adapting a 2D foundation model for floorplan alignment.** We provide PCA visualizations of features before and after our fine-tuning scheme. As illustrated above, the pretrained DINOv3 [40] struggles to bridge the appearance gap, *e.g.*, corresponding regions (white circles) map to different RGB colors. By contrast, our fine-tuning significantly refines the semantic cross-modal alignment. For reference, we show randomly-sampled ground-truth correspondences on the right.

backgrounds or sky regions. Unlike the clean density maps assumed in the floorplan reconstruction literature [54, 7, 24, 26], these artifacts introduce significant noise into the resulting density map, making the subsequent estimation of the 2D similarity transform (Sec. 3.3) unstable.

To obtain a clean and structurally meaningful density map, we identify a subset of 3D points $\bar{\mathcal{P}}_*^{\mathrm{g}}$ that are (i) *geometrically reliable*, (ii) *spatially bounded*, and (iii) *representative of vertical structures*. Specifically, we first remove unreliable geometry based on the 3D reconstruction model's confidence scores. Next, we discard horizontal outliers to retain points within the spatial extent of the scene. Finally, as floorplans primarily depict vertical structures, we filter points along the gravity-aligned axis to suppress floor and ceiling surfaces while preserving layout-defining geometry such as walls.

After filtering, the remaining points $\bar{\mathcal{P}}_*^{\mathrm{g}}$ are orthographically projected onto the horizontal xz-plane, where we count the number of points falling into each grid cell. We then apply gamma correction and normalization to ensure consistent visibility across scenes, yielding a density map $D \in \mathbb{R}^{H\times W\times 1}$ with a top-down, line-drawing modality similar to the reference floorplan $F \in \mathbb{R}^{H\times W\times 3}$.

3.3 Learning Cross-Modal Floorplan–Density Map Correspondences

Equipped with the extracted density map, we perform floorplan alignment by estimating a 2D similarity transform $\mathbf{M} \in \mathrm{Sim}(2)$, which serves to align the reconstructed scene with the floorplan. This transform is parameterized by a scale $s \in \mathbb{R}^+$, rotation $\mathbf{R} \in \mathrm{SO}(2)$, and translation $\mathbf{t} \in \mathbb{R}^2$. We estimate $\mathbf{M}$ via RANSAC [12] using correspondences between the density map and floorplan.

However, establishing cross-modal correspondences is non-trivial. As demonstrated by the PCA visualizations in Figure 3, even a 2D foundation model [40] fails to produce semantically aligned features due to the severe appearance gap between the noisy density map and the clean architectural drawing. To address this, we propose a fine-tuning scheme that facilitates *semantic alignment* and enforces *structural consistency* among correspondences, enabling robust similarity transform estimation. We adopt DINOv3 [40] as a shared encoder $\mathcal{E}$, freeze its pretrained weights, and inject trainable Low-Rank Adaptation (LoRA) [16] layers. These layers are optimized via:

$$\mathcal{L} = \lambda_{\mathrm{feat}}\mathcal{L}_{\mathrm{feat}} + \lambda_{\mathrm{regr}}\mathcal{L}_{\mathrm{regr}} + \lambda_{\mathrm{topo}}\mathcal{L}_{\mathrm{topo}} + \lambda_{\mathrm{geo}}\mathcal{L}_{\mathrm{geo}}, \tag{1}$$

where $\lambda_{\mathrm{feat}}, \lambda_{\mathrm{regr}}, \lambda_{\mathrm{topo}}, \lambda_{\mathrm{geo}}$ are loss coefficients.

Feature Matching Objective. To establish cross-modal correspondences, we employ a contrastive feature matching loss $\mathcal{L}_{\mathrm{feat}}$. The encoder $\mathcal{E}$ extracts feature maps $\mathbf{F}^D, \mathbf{F}^F \in \mathbb{R}^{H'\times W'\times C}$ from the density map D and floorplan F, where $H' = H/16$ and $W' = W/16$. During training, we sample Q ground-truth correspondence pairs $\{(\mathbf{p}_i^D, \mathbf{p}_i^F)\}_{i=1}^Q$ and compute their feature vectors $\mathbf{f}_i^D, \mathbf{f}_i^F \in \mathbb{R}^C$ via bilinear interpolation and ℓ_2-normalization. We then compute a similarity matrix $\mathbf{S} \in \mathbb{R}^{Q\times Q}$ where $\mathbf{S}_{ij} = \langle \mathbf{f}_i^D, \mathbf{f}_j^F \rangle$ measures the cosine similarity, and minimize a symmetric InfoNCE loss [33]:

$$\mathcal{L}_{\mathrm{feat}} = -\frac{1}{2Q}\sum_{i=1}^{Q}\left[\log\left(\frac{\exp(\mathbf{S}_{ii}/\tau)}{\sum_{k=1}^{Q}\exp(\mathbf{S}_{ik}/\tau)}\right) + \log\left(\frac{\exp(\mathbf{S}_{ii}/\tau)}{\sum_{k=1}^{Q}\exp(\mathbf{S}_{ki}/\tau)}\right)\right], \tag{2}$$

where τ is a temperature scaling parameter.

Coordinate Regression Objective. Existing correspondence estimation approaches typically select the maximum similarity on a patch-level feature map and assign its centroid as the match, leading to quantization errors (*e.g.*, $\pm$ 8 pixels for a 16×16 patch size). To achieve sub-patch precision, we introduce a coordinate regression loss $\mathcal{L}_{\text{regr}}$ using a differentiable soft-argmax.

Given the density map feature vectors $\{\mathbf{f}_i^D\}_{i=1}^Q$ and flattened floorplan features $\bar{\mathbf{F}}^F \in \mathbb{R}^{H'W' \times C}$, we compute a similarity matrix $\mathbf{S}' \in \mathbb{R}^{Q \times H'W'}$, convert it into a spatial probability distribution over floorplan patches via softmax, and estimate floorplan coordinates $\{\hat{\mathbf{p}}_i^F\}_{i=1}^Q$ as the expectation over patch centroids. We supervise the prediction with a confidence-weighted Huber loss $\mathcal{H}_\delta(\cdot)$ between the predicted $\hat{\mathbf{p}}_i^F$ and ground-truth $\mathbf{p}_i^F$ floorplan coordinates:

$$\mathcal{L}_{\text{regr}} = \frac{\sum_{i=1}^Q w_i \mathcal{H}_\delta(\|\hat{\mathbf{p}}_i^F - \mathbf{p}_i^F\|_2)}{\sum_{i=1}^Q w_i}, \tag{3}$$

where w_i is the maximum softmax probability for the i-th correspondence, serving as a confidence weight that reflects the sharpness of the distribution.

Structural Consistency Regularization. Point-wise objectives can lead to degenerate similarity transforms when the spatial structure of correspondences collapses. To prevent this, we introduce a topology preservation loss $\mathcal{L}_{\text{topo}}$ and a geometry consistency loss $\mathcal{L}_{\text{geo}}$ as self-supervised structural priors, leveraging the fact that a similarity transform preserves relative angles and distance ratios.

$\mathcal{L}_{\text{topo}}$ enforces angular consistency on triplets (i, j, k) sampled from the Q correspondences via:

$$\mathcal{L}_{\text{topo}} = \frac{\sum_{(i,j,k)} w_i w_j w_k \left[\mathcal{H}_\delta(\cos\hat{\theta}_{ijk}^F - \cos\theta_{ijk}^D) + \mathcal{H}_\delta(\sin\hat{\theta}_{ijk}^F - \sin\theta_{ijk}^D)\right]}{\sum_{(i,j,k)} w_i w_j w_k}, \tag{4}$$

where $\hat{\theta}_{ijk}^F$ and θ_{ijk}^D are the corresponding angles of triangles by $\{\hat{\mathbf{p}}_i^F, \hat{\mathbf{p}}_j^F, \hat{\mathbf{p}}_k^F\}$ and $\{\mathbf{p}_i^D, \mathbf{p}_j^D, \mathbf{p}_k^D\}$.

$\mathcal{L}_{\text{geo}}$ enforces consistent distance ratios over sampled pairs (i, j) by penalizing deviations of the log-distance ratio $\Delta_{ij} = \log\left(\|\hat{\mathbf{p}}_i^F - \hat{\mathbf{p}}_j^F\|_2 / \|\mathbf{p}_i^D - \mathbf{p}_j^D\|_2\right)$ from the weighted mean $\bar{\Delta}$ via:

$$\mathcal{L}_{\text{geo}} = \frac{\sum_{(i,j)} w_i w_j \mathcal{H}_\delta(\Delta_{ij} - \text{sg}(\bar{\Delta}))}{\sum_{(i,j)} w_i w_j}, \tag{5}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator.

Floorplan Alignment from Reliable Correspondences. To robustly estimate the 2D similarity transform $\mathbf{M}$, we identify *highly-confident* and *mutually-close* correspondences. We retain the top 50% ranked by confidence w_i and apply mutual nearest neighbor (MNN) matching. The resulting reliable correspondences are used to estimate $\mathbf{M}$ via RANSAC [12]. We then apply $\mathbf{M}$ to the 3D points $\mathcal{P}_*^{\text{g}}$ by transforming their horizontal (x, z) coordinates while scaling the vertical y coordinates by the scale s to maintain structural proportions. Camera poses are transformed accordingly, producing a floorplan-aligned 3D scene for accurate floorplan localization in the wild.

4 Experiments

We conduct comprehensive experiments to evaluate our method under in-the-wild environments as well as synthetic indoor settings. In Section 4.1, we describe our experimental setup such as implementation details. In Section 4.2, we evaluate the proposed approach on in-the-wild data, including comparisons with baselines (Sec. 4.2.1), robustness analysis under sparse-view settings (Sec. 4.2.2), and ablation studies on learning objectives and 3D reconstruction models (Sec. 4.2.3). In Section 4.3, we evaluate our method on a synthetic indoor dataset for comparison with prior floorplan localization approaches. In Section 4.4, we showcase downstream applications such as interior-exterior 3D scene alignment using a reference floorplan.

The appendix supplements our main results with additional experiments. For example, we analyze the stability of our model against various hyperparameters (Sec. E.2) and validate our design choices, including correspondence filtering strategies (Table E.1) and LoRA configurations (Table E.2). Our project page also shows 360° view comparisons of floorplan-aligned 3D scenes.

Table 1: **Camera pose estimation on C3 [17].** Our fine-tuning scheme significantly improves the 2D foundation model [40], outperforming prior methods by a large margin across all metrics.

	Angular Recall ↑				Positional Recall ↑			Ang. & Pos. ↑
Method	@ 5°	@ 10°	@ 20°	@ 30°	@ 5%	@ 10%	@ 20%	@ (30°, 20%)
LoFTR [42]	4.67	8.31	15.11	22.80	1.09	4.27	13.96	3.07
DUSt3R [47]	5.14	10.06	18.02	24.59	1.54	5.15	14.71	4.64
C3Po [17]	24.45	36.90	48.59	54.89	16.60	28.59	42.12	32.96
DINOv3 [40]	14.25	19.59	28.87	35.33	6.75	14.23	26.67	18.28
Ours	**65.91**	**75.93**	**80.23**	**83.08**	**50.87**	**65.58**	**76.08**	**73.58**

Table 2: **Correspondence estimation on C3 [17].** Our adaptation of the 2D foundation model [40] achieves substantially higher accuracy than prior methods across all pixel-level metrics.

	PCK ↑						
Method	@ 1%	@ 3%	@ 5%	@ 10%	@ 15%	@ 30%	RMSE ↓
LoFTR [42]	0.14	1.04	2.49	7.40	13.59	37.13	0.3227
DUSt3R [47]	0.09	0.76	2.10	7.69	15.78	40.20	0.2844
C3Po [17]	9.06	28.37	34.91	45.83	54.26	72.25	0.1780
DINOv3 [40]	0.73	3.95	7.95	16.25	24.31	47.01	0.2624
Ours	**20.42**	**58.13**	**69.87**	**79.94**	**83.64**	**90.45**	**0.0776**

4.1 Experimental Setup

Implementation Details. We reconstruct 3D scene geometry using π^3 [49] and predict gravity using GeoCalib [44]. For correspondence estimation, we employ a pretrained DINOv3 ViT-B/16 [40] and inject trainable LoRA [16] layers, which are optimized using AdamW [27] with a learning rate of 10^{-4}. We adopt the same settings (*e.g.*, model hyperparameters) for both in-the-wild and synthetic indoor evaluations. Comprehensive details are provided in the appendix (Sec. C).

Varying-View Inference. We evaluate our method across varying numbers of input views. By default, scenes with $N > 150$ images are partitioned into $\lceil N/150 \rceil$ chunks, whereas smaller scenes are processed entirely. We also evaluate under sparser settings, *e.g.*, Ours ($= 1$) takes a single image as input for 3D reconstruction with other settings unchanged in Figures 5 and 6.

4.2 Evaluation on In-the-Wild Data

We evaluate our method at multiple levels of granularity, reporting both image-level camera pose estimation and pixel-level correspondence metrics.

C3 Dataset. C3 [17] is a large-scale in-the-wild dataset of diverse photographs paired with floorplans, providing camera pose and correspondence annotations. These annotations are derived by registering Structure-from-Motion reconstructions to floorplans, inevitably including geometric misalignments in unconstrained settings. To ensure a reliable testbed, we curate a clean subset by pruning samples with severe errors. For example, images with severe optical distortions or non-photographic content are removed (Figure B.1). Importantly, this filtering preserves the original dataset's diversity, retaining 574 out of 597 scenes (96.15%). Details are provided in the appendix (Sec. B).

Camera Pose Evaluation Metrics. Camera pose estimation is measured using *Angular Recall* and *Positional Recall*, which evaluate camera yaw errors and 2D horizontal center distances against the ground truth, respectively. Following C3Po [17], we report Angular Recall at $\{5°, 10°, 20°, 30°\}$, Positional Recall at $\{5\%, 10\%, 20\%\}$ of the floorplan diagonal length, and their combined recall.

Correspondence Evaluation Metrics. Pixel-level correspondence is measured using *Percentage of Correct Keypoints (PCK)* and *Root Mean Square Error (RMSE)*. PCK quantifies the proportion of correspondences within a distance threshold from the ground truth. Following C3Po [17], distances are normalized by the floorplan resolution. We report PCK at $\{1\%, 3\%, 5\%, 10\%, 15\%, 30\%\}$.

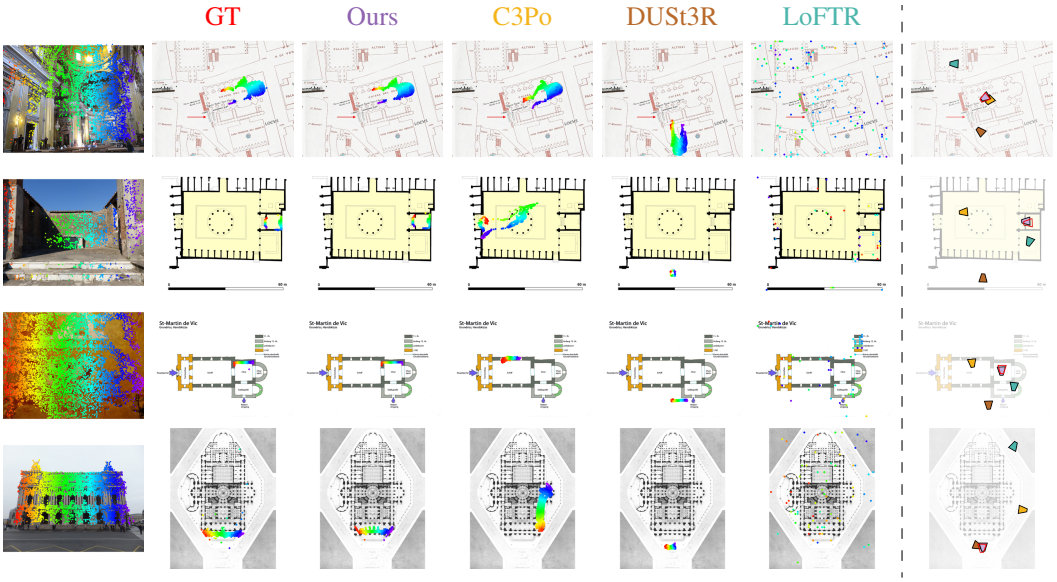


Figure 4: **Qualitative comparison in the wild.** We compare correspondence predictions across baselines and our method, with floorplan localization results shown on the right. Cameras are illustrated in corresponding colors (*e.g.*, for GT, and for Ours).

4.2.1 Comparison with Correspondence-based Methods

Baselines. We compare against correspondence-based methods [17, 47, 42] on the clean subset of C3. C3Po [17] builds on the DUSt3R architecture [47] and is fine-tuned on the C3 dataset to learn correspondences between perspective photographs and abstract floorplans. We follow C3Po's evaluation protocol for these methods: candidate camera poses are predicted by solving epipolar geometry from estimated correspondences, where the candidate closest to the ground truth is selected.

To isolate the contribution of our adaptation scheme, we also compare against pretrained DINOv3 [40] as the encoder $\mathcal{E}$ in our inference pipeline. We ablate our learning objectives in Section 4.2.3, with additional ablations in the appendix (Sec. E).

Quantitative Evaluation. As shown in Tables 1 and 2, our method significantly outperforms all baselines at both image-level and pixel-level, achieving over 100% improvements across most evaluation metrics. For example, in terms of combined angular-positional recall, our approach surpasses the strongest baseline, C3Po, by 123.24% (73.58 vs. 32.96) and the pretrained DINOv3 baseline by 302.52% (73.58 vs. 18.28). With regard to RMSE, C3Po's error is 129.38% (0.1780 vs. 0.0776) higher than ours, and DINOv3's error is 238.14% (0.2624 vs. 0.0776) higher. The large gap over the DINOv3 baseline highlights the contribution of the proposed fine-tuning scheme. These results demonstrate the effectiveness of our approach for floorplan localization in the wild.

Qualitative Evaluation. Figure 4 visualizes correspondence and camera pose estimation results against correspondence-based baselines, showing that our predictions align most closely with the ground truth. In the second row, our estimated correspondences structurally match the ground truth, whereas all baselines produce physically implausible results. This highlights the advantage of our 3D-grounded approach that enforces spatial rigidity through gravity-aligned 3D scene reconstruction. Furthermore, as shown in the third row, our method successfully localizes minimal-context photos (*e.g.*, wall drawings) by leveraging global 3D scene reconstructions, addressing a severely challenging scenario where single-view estimation methods typically fail.

4.2.2 Robustness to Sparse Inputs

We evaluate how effectively our method localizes given a limited number of input images. As shown in Figure 5, the proposed method significantly outperforms C3Po even in the single-view setting, and achieves accuracy comparable to the 150-view setting with as few as 10 to 30 views. Figure 6 further

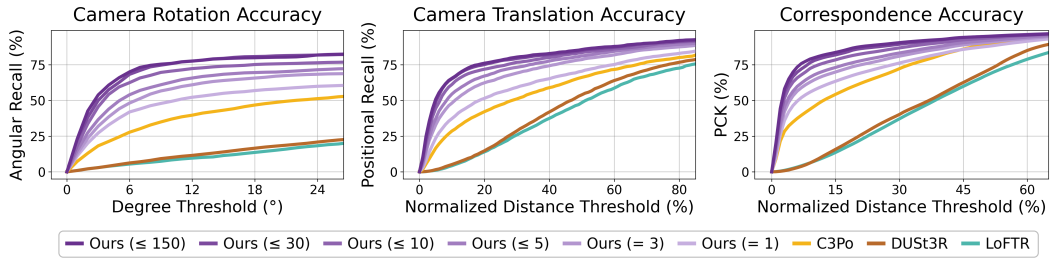


Figure 5: **Performance across varying view counts on C3 [17].** We evaluate the proposed method using different numbers of input images for 3D reconstruction (*e.g.*, ≤ 150 denotes a maximum of 150 images per reconstruction). Notably, Ours ($= 1$) already outperforms C3Po [17] by a large margin, while Ours (≤ 30) is on par with Ours (≤ 150).

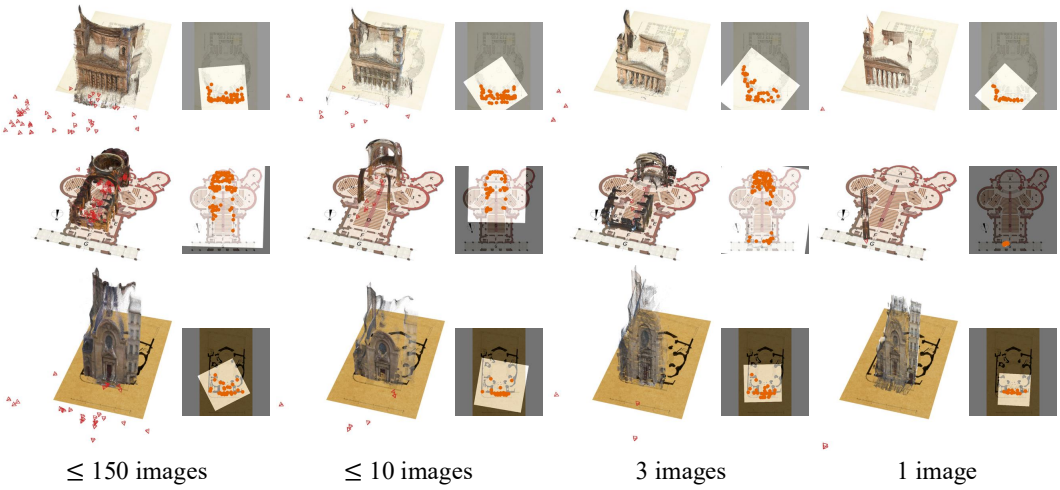


Figure 6: **In-the-wild floorplan alignment across varying view counts.** We visualize 3D points used for density map extraction and resulting density maps with reliable correspondences overlaid (in orange). Sparse views often suffice to recover geometry informative enough for floorplan alignment.

shows that even a small set of images is typically sufficient to construct a structurally meaningful density map, enabling accurate floorplan alignment. However, the single-view setting sometimes suffers from localization ambiguity, as shown in the second row. Because a single image captures only a limited region of the scene, its reconstructed geometry (*e.g.*, a single wall segment) may match multiple similar structures on the floorplan, leading to the accuracy drop observed in Figure 5.

4.2.3 Ablation Study

Loss Functions. As summarized in Table 3, we evaluate the effectiveness of fine-tuning with the proposed learning objectives by progressively integrating them into our cross-modal correspondence learning scheme. The feature matching loss $\mathcal{L}_{\text{feat}}$ establishes semantic alignment, which significantly enhances the pretrained DINOv3 [40]. The coordinate regression loss $\mathcal{L}_{\text{regr}}$ further transitions the model from estimating discrete grid indices to continuous spatial coordinates. To prevent degenerate similarity solutions where the model satisfies the regression objective through structurally inconsistent mappings, we incorporate the structural consistency regularization losses $\mathcal{L}_{\text{topo}}$ and $\mathcal{L}_{\text{geo}}$. These losses explicitly penalize non-rigid deformations in terms of triplet angles and pairwise distances, which should be invariant under similarity transformations. The full objective ensures the learned correspondences are both semantically aligned and structurally consistent.

3D Reconstruction Models. Table 4 compares the effect of using different 3D foundation models for gravity-aligned 3D scene reconstruction. π^3 [49] consistently outperforms VGGT [45] across all metrics, motivating its use as our default 3D reconstruction model.

Table 3: **Ablation study on loss functions.** We evaluate each loss function's contribution to cross-modal correspondence learning by progressively adding the feature matching loss $\mathcal{L}_{\text{feat}}$, coordinate regression loss $\mathcal{L}_{\text{regr}}$, topology preservation loss $\mathcal{L}_{\text{topo}}$, and geometry consistency loss $\mathcal{L}_{\text{geo}}$.

				Ang. ↑		Pos. ↑		Ang. & Pos. ↑	PCK ↑		
$\mathcal{L}_{\text{feat}}$	$\mathcal{L}_{\text{regr}}$	$\mathcal{L}_{\text{topo}}$	$\mathcal{L}_{\text{geo}}$	@ 5°	@ 30°	@ 5%	@ 20%	@ (30°, 20%)	@ 1%	@ 30%	RMSE ↓
				14.25	35.33	6.75	26.67	18.28	0.73	47.01	0.2624
✓				59.23	80.08	48.71	72.42	69.10	20.01	86.85	0.0900
✓	✓			60.50	80.17	48.74	73.78	70.60	**21.12**	86.39	0.0938
✓	✓	✓		64.18	80.84	49.63	75.24	70.54	20.00	89.40	0.0818
✓	✓	✓	✓	**65.91**	**83.08**	**50.87**	**76.08**	**73.58**	20.42	**90.45**	**0.0776**

Table 4: **Effect of 3D reconstruction models.** We compare π^3 [49] and VGGT [45] as the 3D estimator for gravity-aligned 3D scene reconstruction.

3D	Ang. ↑		Pos. ↑		Ang. & Pos. ↑	PCK ↑		
Estimator	@ 5°	@ 30°	@ 5%	@ 20%	@ (30°, 20%)	@ 1%	@ 30%	RMSE ↓
VGGT	61.48	81.43	47.25	75.16	71.52	18.27	88.63	0.0873
π^3	**65.91**	**83.08**	**50.87**	**76.08**	**73.58**	**20.42**	**90.45**	**0.0776**

Selective Projection. In Table 5, we conduct an ablation that directly measures the contribution of each filtering stage in Selective Projection (Sec. 3.2). Without using the proposed projection, the combined angular-positional recall drops from 73.58 to 62.11. Confidence filtering and horizontal filtering account for most of this gap, together recovering from 62.11 to 73.22. Vertical filtering contributes least, but we retain it as floorplans primarily depict vertical structures. The exact threshold values matter far less than the presence of each stage. Figure E.1 plots accuracy across different hyperparameter values for the projection.

Table 5: **Effect of Selective Projection.** We measure the contribution of each filtering: confidence filtering (ρ_{conf}), horizontal filtering (ρ_{xz}), and vertical filtering (ρ_y).

			Ang. & Pos. ↑
ρ_{conf}	ρ_{xz}	ρ_y	@ (30°, 20%)
			62.11
✓			70.11
✓	✓		73.22
✓	✓	✓	**73.58**

4.3 Evaluation on Synthetic Data

In addition to in-the-wild scenarios, we also evaluate our 3D-grounded approach on a standard indoor benchmark [55]. This enables direct comparison with prior localization methods.

Structured3D Dataset. Structured3D [55] is a large-scale synthetic indoor dataset of photorealistic renderings paired with vectorized floorplans, providing precise geometry under controlled environments. This dataset has been widely used as a testbed for floorplan localization. We use its fully furnished perspective images with the official training and test splits.

Camera Pose Evaluation Metrics. Camera pose estimation is measured using *Angular Recall* and *Positional Recall*. Following the standard protocol [32, 6, 53], we report Positional Recall at $\{0.1\text{m}, 0.5\text{m}, 1\text{m}\}$ and the combined angular-positional recall at $(30°, 1\text{m})$.

4.3.1 Comparison with Indoor Localization Methods

Baselines. We compare against state-of-the-art indoor localization methods [32, 6, 53, 13], which rely on pre-defined discretized pose spaces, *e.g.*, an $8\text{m} \times 10\text{m}$ floorplan discretized at 0.1m and $10°$ yields $80 \times 100 \times 36$ camera pose candidates. To exhaustively evaluate these candidates, they further require preprocessing of floorplans, *e.g.*, converting floorplans into ray-based representations. SemRayLoc [13] additionally relies on semantic labels (*e.g.*, walls, doors, and windows). In contrast, our method directly operates on raw rasterized floorplans without such constraints.

Table 6: **Single-view indoor localization on Structured3D [55].** Recent localization methods rely on pre-defined discretized pose spaces and require preprocessing of floorplans, whereas our method directly operates on raw floorplans while achieving competitive results.

Method	Discretized Pose Space	Floorplan Preprocessing	Semantic Labels	Positional Recall ↑ @ 0.1m	@ 0.5m	@ 1m	Ang. & Pos. ↑ @ (30°, 1m)
LASER [32]	✓	✓	✗	0.7	6.4	10.4	8.7
F^3Loc [6]	✓	✓	✗	1.5	14.6	22.4	21.3
UnLoc [53]	✓	✓	✗	5.3	33.9	38.8	37.6
SemRayLoc [13]	✓	✓	✓	**5.4**	**41.9**	53.5	**52.6**
Ours	✗	✗	✗	3.5	37.5	**53.8**	51.6

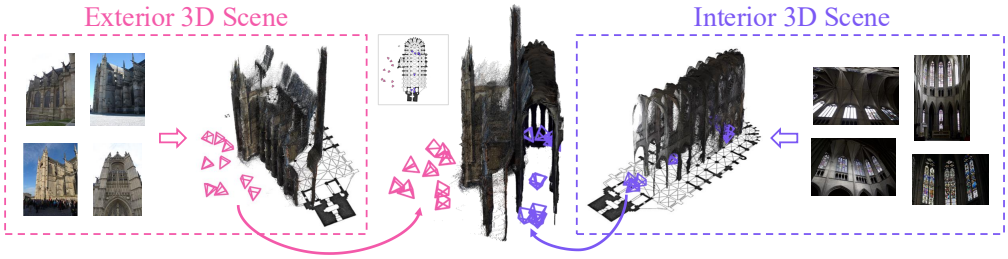


Figure 7: **Alignment of interior and exterior 3D scenes.** Using the floorplan as a shared geometric bridge, our approach enables independent alignment of interior and exterior reconstructions into a unified global coordinate system. This is achieved despite minimal visual overlap and large viewpoint differences between indoor and outdoor scenes.

Quantitative Evaluation. As shown in Table 6, our method achieves strong single-view localization accuracy despite operating under a significantly more general setting. The proposed method is competitive with SemRayLoc [13], which additionally relies on semantic labels, and outperforms UnLoc [53] by 37.2% (51.6 vs. 37.6) on the combined angular-positional recall, demonstrating the versatility of our method alongside in-the-wild scenarios.

4.4 Downstream Applications

A key advantage of floorplan-based alignment is that the floorplan provides a shared geometric reference for image collections that do not directly overlap in appearance or viewpoint. This enables separate reconstructions to be registered into a common coordinate system even when direct 3D alignment is not possible. We demonstrate this capability in the challenging setting of interior and exterior image collections of the same building, where minimal visual overlap and large viewpoint differences typically prevent coherent joint reconstruction (Figure 7). Our method also supports the alignment of disjoint image collections without any overlapping regions (Figure H.2).

5 Conclusion

In this work, we introduced SceneAligner, an approach that revisits floorplan localization through the lens of contemporary visual foundation models. By leveraging the geometric priors of pretrained 3D reconstruction models together with the representational capabilities of 2D visual foundation models, our approach enables reframing floorplan localization as a reconstruction and alignment problem grounded in scene geometry. While our method demonstrates encouraging performance, its reliance on 3D foundation models presents an inherent limitation: inaccuracies in the reconstructed geometry can propagate to the density map, thereby impacting alignment quality. Future work can explore improving robustness to imperfect reconstructions and jointly reasoning about scene geometry and floorplan alignment for more reliable floorplan localization in challenging environments.

Acknowledgments and Disclosure of Funding

This work received no third-party funding, and the authors declare no competing interests.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*, 2025.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[3] Federico Boniardi, Tim Caselitz, Rainer Kummerle, and Wolfram Burgard. Robust LiDAR-based localization in architectural floor plans. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3318–3324. IEEE, 2017.

[4] Federico Boniardi, Tim Caselitz, Rainer Kümmerle, and Wolfram Burgard. A pose graph-based localization system for long-term navigation in CAD floor plans. *Robotics and Autonomous Systems*, pages 84–97, 2019.

[5] Federico Boniardi, Abhinav Valada, Rohit Mohan, Tim Caselitz, and Wolfram Burgard. Robot Localization in Floor Plans Using a Room Layout Edge Extraction Network. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5291–5297. IEEE, 2019.

[6] Changan Chen, Rui Wang, Christoph Vogel, and Marc Pollefeys. F3Loc: Fusion and Filtering for Floorplan Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18029–18038, 2024.

[7] Jiacheng Chen, Chen Liu, Jiaye Wu, and Yasutaka Furukawa. Floor-SP: Inverse CAD for Floorplans by Sequential Room-wise Shortest Path. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[8] Hang Chu, Dong Ki Kim, and Tsuhan Chen. You Are Here: Mimicking the Human Thinking Process in Reading Floor-Plans. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2210–2218, 2015.

[9] Andrea Cohen, Johannes L Schönberger, Pablo Speciale, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. Indoor-Outdoor 3D Reconstruction Alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 285–300. Springer, 2016.

[10] Tamir Cohen, Leo Segre, Shay Shomer-Chai, Shai Avidan, and Hadar Averbuch-Elor. Scene Grounding In the Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[11] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 224–236, 2018.

[12] 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.

[13] Yuval Grader and Hadar Averbuch-Elor. Supercharging Floorplan Localization with Semantic Rays. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 27116–27125, 2025.

[14] Henry Howard-Jenkins and Victor Adrian Prisacariu. LaLaLoc++: Global Floor Plan Comprehension for Layout Localisation in Unvisited Environments. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 693–709, 2022.

[15] Henry Howard-Jenkins, Jose-Raul Ruiz-Sarmiento, and Victor Adrian Prisacariu. LaLaLoc: Latent Layout Localisation in Dynamic, Unvisited Environments. *arXiv preprint arXiv:2104.09169*, 2021.

[16] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2022.

[17] Kuan Wei Huang, Brandon Li, Bharath Hariharan, and Noah Snavely. C3Po: Cross-View Cross-Modality Correspondence by Pointmap Prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[18] Seigo Ito, Felix Endres, Markus Kuderer, Gian Diego Tipaldi, Cyrill Stachniss, and Wolfram Burgard. W-RGB-D: Floor-plan-based indoor global localization using a depth camera and WiFi. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 417–422. IEEE, 2014.

[19] Junho Kim, Jiwon Jeong, and Young Min Kim. Fully Geometric Panoramic Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[20] Yuan Li, Yuanbo Xiangli, Hadar Averbuch-Elor, Noah Snavely, and Ruojin Cai. Long-tail Internet photo reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[21] Zhikai Li, Marcelo H Ang, and Daniela Rus. Online Localization with Imprecise Floor Space Maps using Stochastic Gradient Descent. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8571–8578. IEEE, 2020.

[22] Cheng Lin, Changjian Li, and Wenping Wang. Floorplan-Jigsaw: Jointly Estimating Scene Layout and Aligning Partial Scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[23] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[24] Chen Liu, Jiaye Wu, and Yasutaka Furukawa. FloorNet: A Unified Framework for Floorplan Reconstruction from 3D Scans. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[25] Yifan Liu, Zhiyuan Min, Zhenwei Wang, Junta Wu, Tengfei Wang, Yixuan Yuan, Yawei Luo, and Chunchao Guo. WorldMirror: Universal 3D World Reconstruction with Any-Prior Prompting, 2025.

[26] Yuzhou Liu, Lingjie Zhu, Xiaodong Ma, Hanqiao Ye, Xiang Gao, Xianwei Zheng, and Shuhan Shen. PolyRoom: Room-aware Transformer for Floorplan Reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

[27] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[28] David G Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*, 2004.

[29] Ricardo Martin-Brualla, Yanling He, Bryan C Russell, and Steven M Seitz. The 3D Jigsaw Puzzle: Mapping Large Indoor Spaces. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 1–16. Springer, 2014.

[30] Oscar Mendez, Simon Hadfield, Nicolas Pugeault, and Richard Bowden. SeDAR: Reading Floorplans Like a Human—Using Deep Learning to Enable Human-Inspired Localisation. *International Journal of Computer Vision (IJCV)*, 128:1286–1310, 2020.

[31] Rachel Mikulinsky, Morris Alper, Shai Gordin, Enrique Jiménez, Yoram Cohen, and Hadar Averbuch-Elor. ProtoSnap: Prototype Alignment for Cuneiform Signs. In *International Conference on Learning Representations (ICLR)*, volume 2025, pages 88720–88739, 2025.

[32] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11122–11131, 2022.

[33] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*, 2018.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021.

[35] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.

[36] Bryan C Russell, Ricardo Martin-Brualla, Daniel J Butler, Steven M Seitz, and Luke Zettlemoyer. 3D Wikipedia: Using online text to automatically label and navigate reconstructed geometry. *ACM Transactions on Graphics (TOG)*, 32(6):1–10, 2013.

[37] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning Feature Matching With Graph Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[38] Ian Simon and Steven M Seitz. Scene Segmentation Using the Wisdom of Crowds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 541–553. Springer, 2008.

[39] Ian Simon, Noah Snavely, and Steven M Seitz. Scene Summarization for Online Image Collections. In *2007 IEEE 11th International conference on computer vision*, pages 1–8. IEEE, 2007.

[40] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

[41] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, 2024.

[42] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-Free Local Feature Matching with Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[43] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent Correspondence from Image Diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[44] Alexander Veicht, Paul-Edouard Sarlin, Philipp Lindenberger, and Marc Pollefeys. GeoCalib: Learning Single-image Calibration with Geometric Optimization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

[45] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual Geometry Grounded Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[46] Shenlong Wang, Sanja Fidler, and Raquel Urtasun. Lost Shopping! Monocular Localization in Large Indoor Spaces. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2695–2703, 2015.

[47] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D Vision Made Easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[48] Xipeng Wang, Ryan J Marcotte, and Edwin Olson. GLFP: Global Localization from a Floor Plan. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1627–1632. IEEE, 2019.

[49] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-Equivariant Visual Geometry Learning. In *International Conference on Learning Representations (ICLR)*, 2026.

[50] Tobias Weyand and Bastian Leibe. Discovering Details and Scene Structure with Hierarchical Iconoid Shift. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3479–3486, 2013.

[51] Erik Wijmans and Yasutaka Furukawa. Exploiting 2D Floorplan for Building-Scale Panorama RGBD Alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[52] Xiaoshi Wu, Hadar Averbuch-Elor, Jin Sun, and Noah Snavely. Towers of Babel: Combining Images, Language, and 3D Geometry for Learning Multimodal Vision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 428–437, 2021.

[53] Matthias Wüest, Francis Engelmann, Ondrej Miksik, Marc Pollefeys, and Daniel Barath. UnLoc: Leveraging Depth Uncertainties for Floorplan Localization. In *International Conference on Learning Representations (ICLR)*, 2026.

[54] Yuanwen Yue, Theodora Kontogianni, Konrad Schindler, and Francis Engelmann. Connecting the Dots: Floorplan Reconstruction Using Two-Level Queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[55] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3D: A Large Photo-realistic Dataset for Structured 3D Modeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Appendix

We refer readers to our project page (`https://Cornell-VAILab.github.io/SceneAligner`) for 360° view comparisons of floorplan-aligned 3D reconstructions (Sec. A). In this document, we provide more dataset details (Sec. B), implementation details (Sec. C), qualitative results (Sec. D), quantitative evaluations (Sec. E), analyses (Sec. F), limitations (Sec. G), applications (Sec. H), and broader impacts (Sec. I), which are not included in the main paper due to space constraints.

A 360° View Comparisons of Floorplan Alignment

Our project page shows 360° view renderings of floorplan-aligned 3D reconstructions, allowing visual comparison of alignment results before and after fine-tuning DINOv3 [40].

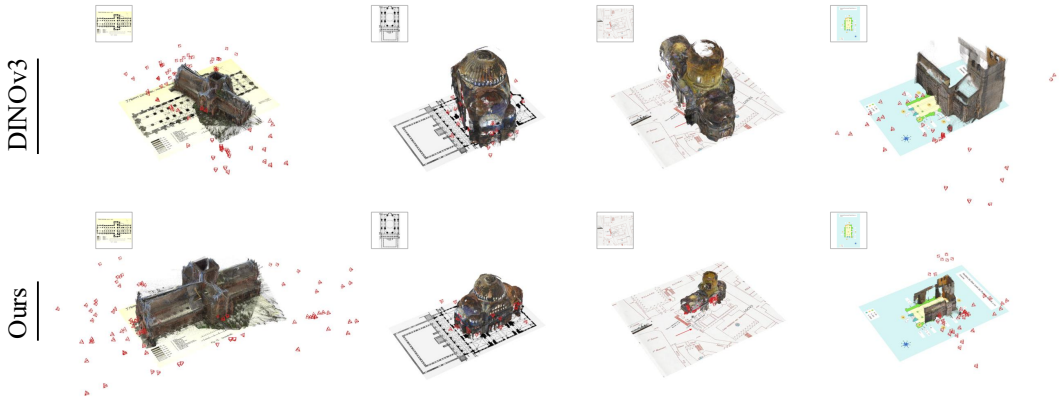


Figure A.1: **Preview.** We compare floorplan alignment results using correspondences estimated by pretrained DINOv3 [40] and our fine-tuned model.

B In-the-Wild Dataset Details

Table B.1 shows statistics for the C3 dataset [17] used in our experiments. The original dataset contains samples with severe annotation errors due to noise in Structure-from-Motion reconstructions and their registration to floorplans. To establish a reliable benchmark, we construct a clean subset by filtering such samples, retaining 463 of 479 training scenes and 111 of 118 test scenes. Furthermore, we distinguish view types of the scenes by annotating each image as `interior` or `exterior`, resulting in 738 unique scenarios. Each scenario is defined by a triplet $\langle \mathrm{s}, \mathrm{v}, \mathrm{f} \rangle$, where s denotes the scene, $v \in \{\mathrm{interior}, \mathrm{exterior}\}$ indicates the scene view type, and f denotes the floorplan.

Filtering Strategy. We curate the clean subset using a multi-stage pipeline designed to remove noise with varying degrees of detectability. We first apply a set of geometric and heuristic filters to eliminate duplicates, images with extreme aspect ratios, and pairs with too few correspondences for reliable matching or excessive out-of-boundary correspondences. These algorithmic filters remove obvious low-quality samples. We then use Qwen2.5-VL [1] to identify failure cases that are harder to capture with simple rules, such as severe optical distortions (*e.g.*, fisheye images) and non-photographic inputs (*e.g.*, blueprints). Finally, we conduct two rounds of visual inspection at both image and scene levels to remove remaining problematic samples. Figure B.1 shows examples of filtered samples.

Interior/Exterior Labeling. We assign each image an `interior` or `exterior` label through a multi-stage pipeline. First, we employ Qwen2.5-VL [1] to classify each image as `interior`, `exterior`, or `unsure` based on camera location and viewpoint. For `unsure` cases, tentative labels are assigned using CLIP [34] as a zero-shot classifier. Predicted labels are then overridden when structural keywords (*e.g.*, "facade", "nave") appear in the file path. Finally, we conduct three rounds of visual inspection to correct any remaining misclassifications.

Table B.1: **Dataset statistics of the clean subset from C3 [17].** Scene-level and image-level statistics are reported, grouped by interior and exterior view types. In the Subtype column, `Scene` denotes the number of unique scenes. `Scene-Floorplan` counts scene-floorplan pairs, as a scene may be associated with multiple floorplans. `Scene-{Interior, Exterior}-Floorplan` further distinguishes view types within each pair, since images from the same scene can be categorized as either `Interior` or `Exterior`.

Split	Level	Subtype	Count
Train	Scene	`Scene`	463
		∟ `Scene-Floorplan`	498
		∟ `Scene-{Interior, Exterior}-Floorplan`	598
		∟ `Scene-Interior-Floorplan`	186
		∟ `Scene-Exterior-Floorplan`	412
	Image	`Interior`	14,781
		`Exterior`	38,096
Test	Scene	`Scene`	111
		∟ `Scene-Floorplan`	120
		∟ `Scene-{Interior, Exterior}-Floorplan`	140
		∟ `Scene-Interior-Floorplan`	38
		∟ `Scene-Exterior-Floorplan`	102
	Image	`Interior`	5,659
		`Exterior`	6,555

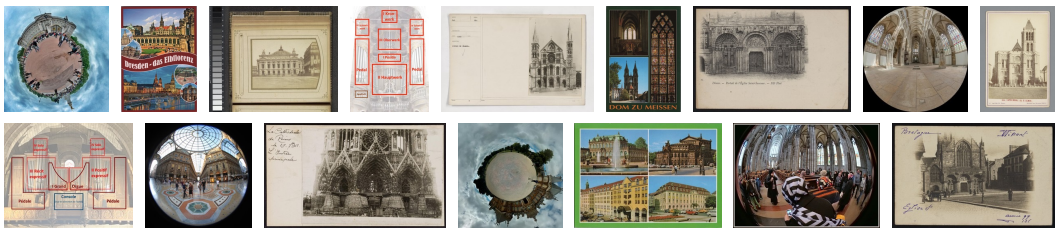


Figure B.1: **Examples of filtered samples.** We show samples removed from the original C3 dataset [17] by our filtering pipeline. For example, images with severe optical distortions (*e.g.*, fisheye lenses) and non-photographic content (*e.g.*, blueprints, illustrations) are excluded.

Evaluation Protocol. All methods are evaluated on the same cleaned test split, using the same input images and correspondence annotations. We summarize the protocol in Table B.2. Following the C3Po [17] evaluation protocol, oracle pose selection is applied to the correspondence-based baselines. Such an oracle protocol is advantageous to those baselines, whereas our 3D-grounded approach obtains camera poses directly from the reconstructed 3D scene. We additionally report C3Po trained on the same cleaned training split (C3Po†), together with our accuracy across input view counts (Figure 5). C3Po† exceeds C3Po under the same training configuration despite fewer training samples after filtering, indicating that our curation removes erroneous samples while retaining the scene diversity of the original dataset. As shown in Table B.2, our method outperforms prior work even in the single-view setting by a large margin.

C Additional Implementation Details

This section provides additional implementation details regarding our cross-modal correspondence estimation for floorplan alignment, extending Section 4.1 of the main paper. The following subsections describe our setup for in-the-wild evaluation. We detail the architecture of the correspondence estimation network (Sec. C.1), followed by data augmentation strategies (Sec. C.2) and training details (Sec. C.3). We then specify the inference settings used to predict a 2D similarity transform (Sec. C.4). The setup for synthetic indoor evaluation is provided in Section C.5.

Table B.2: **Evaluation protocol on C3 [17].** `N/A` denotes off-the-shelf pretrained models without C3-specific training. $\mathcal{O}$ denotes the original C3 split, which contains the problematic samples identified by our cleaning protocol, and $\mathcal{C}$ denotes the resulting split after removing them. Oracle pose selection denotes selecting the best pose candidate with respect to ground truth.

Method	Training Split	Test Split	# Input Views	Oracle Pose Selection	Ang. & Pos. ↑ @ $(30°, 20\%)$
LoFTR [42]	`N/A`	$\mathcal{C}$	1	✓	3.07
DUSt3R [47]	`N/A`	$\mathcal{C}$	1	✓	4.64
C3Po [17]	$\mathcal{O}$	$\mathcal{C}$	1	✓	32.96
C3Po† [17]	$\mathcal{C}$	$\mathcal{C}$	1	✓	**36.58**
DINOv3 [40]	`N/A`	$\mathcal{C}$	≤ 150	✗	18.28
Ours ($= 1$)	$\mathcal{C}$	$\mathcal{C}$	1	✗	46.36
Ours ($= 3$)	$\mathcal{C}$	$\mathcal{C}$	3	✗	56.28
Ours (≤ 5)	$\mathcal{C}$	$\mathcal{C}$	≤ 5	✗	61.73
Ours (≤ 10)	$\mathcal{C}$	$\mathcal{C}$	≤ 10	✗	67.57
Ours (≤ 30)	$\mathcal{C}$	$\mathcal{C}$	≤ 30	✗	72.07
Ours (≤ 150)	$\mathcal{C}$	$\mathcal{C}$	≤ 150	✗	**73.58**

C.1 Model Architecture

Correspondence Estimation Network. We use DINOv3 ViT-B/16 [40] as the shared encoder $\mathcal{E}$. To adapt the encoder to our task while preserving its pretrained visual representations, we freeze the backbone weights and inject trainable Low-Rank Adaptation (LoRA) [16] layers into the attention and MLP modules of each transformer block, with rank $r = 16$, scaling factor $\alpha = 16$, and dropout rate 0.1. Given a density map $D \in \mathbb{R}^{H\times W\times 1}$ and a reference floorplan $F \in \mathbb{R}^{H\times W\times 3}$, the encoder produces C-dimensional feature maps with spatial resolution $H/16 \times W/16$, where $C = 768$. The input resolution $H \times W$ is fixed to 512×512 at inference time, but we vary it during training.

C.2 Data Augmentation

Data Augmentation for Extracting Density Maps. To simulate the noise of in-the-wild 3D reconstructions, we derive density maps under various configurations. Specifically, we use two reconstruction models (π^3 [49] and VGGT [45]) [1] with two image preprocessing modes (cropping and padding) that produce 518×518 input images. For each 3D scene reconstruction, we randomly sample between 50 and 150 images, while scenes with fewer images are processed using all available ones. As shown in Figure C.1, we also randomize these hyperparameters: reconstruction confidence thresholds $\rho_{\text{conf}} \in \{30\%, 45\%, 60\%, 75\%, 90\%\}$, horizontal percentile thresholds $\rho_{xz} \in \{0\%, 2.5\%, 5\%\}$, and gamma values $\gamma \in \{0.25, 0.5, 0.75, 1\}$. Combined with the height slicing percentiles $\left[\rho_y^{\min}, \rho_y^{\max}\right] = [20\%, 95\%]$, these configurations yield $\sim 140K$ density maps. Furthermore, $\sim 140K$ density maps from single-view reconstructions are included to effectively handle sparse-view scenarios.

Data Augmentation on Density Maps and Floorplans. During training, we apply on-the-fly augmentations to density maps and floorplans. For geometric augmentations, we use random rotations sampled from $\{0°, 90°, 180°, 270°\}$ and random cropping with a scale factor of $[0.7, 1.0]$ from a randomly selected top-left corner. For photometric augmentations, we apply color jittering: floorplans are perturbed with brightness 0.5, contrast 0.5, saturation 0.5, and hue 0.1, whereas density maps are adjusted with brightness 0.2 and contrast 0.2.

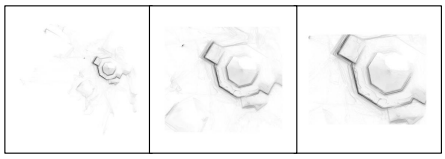
Mixed-Resolution Training Strategy. DINOv3 [40] utilizes Rotary Positional Embeddings (RoPE) [41], allowing the model to process variable input resolutions without architectural modifications. We leverage this property and dynamically vary input resolutions during training to improve correspondence matching across diverse floorplans with varying target region sizes. All images within each batch share the same spatial resolution, which is randomly selected per batch from $\{512 \times 512,\ 768 \times 768,\ 1024 \times 1024,\ 1280 \times 1280\}$.

[1] We use the model weights from [20], trained for robust 3D reconstruction from noisy Internet photo collections.



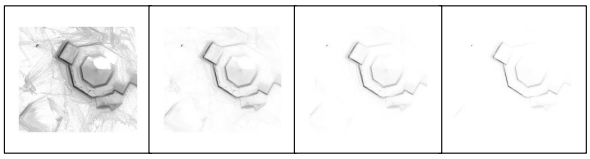
30% 45% 60% 75% 90%

(a) Confidence Percentile Threshold ρ_{conf}



0% 2.5% 5%

(b) Horizontal Percentile Threshold ρ_{xz}



0.25 0.5 0.75 1

(c) Gamma Value γ

Figure C.1: **Extracting density maps across varying hyperparameters.** We visualize density maps obtained by varying the confidence percentile threshold ρ_{conf}, horizontal percentile threshold ρ_{xz}, and gamma value γ, with others fixed at their default values ($\rho_{\text{conf}} = 45\%$, $\rho_{xz} = 2.5\%$, $\gamma = 0.5$).

C.3 Training Details

Loss Coefficients and Curriculum Learning. At each training iteration, $Q = 1024$ randomly sampled correspondence pairs are used for loss calculation. The loss coefficients are set to $\lambda_{\text{feat}} = 1, \lambda_{\text{regr}} = 50, \lambda_{\text{topo}} = 10$, and $\lambda_{\text{geo}} = 10$. We set the temperature parameter to $\tau = 0.07$ for the feature matching loss. To stabilize training, we employ a curriculum learning strategy. During the first 10% of training iterations, the network is trained only using the feature matching loss $\mathcal{L}_{\text{feat}}$. From 10% to 20%, the coordinate regression loss $\mathcal{L}_{\text{regr}}$ is introduced. For the remaining iterations, we also incorporate the structural regularization terms $\mathcal{L}_{\text{topo}}$ and $\mathcal{L}_{\text{geo}}$.

Optimization. The network is optimized using the AdamW optimizer [27] with a learning rate of 10^{-4} and a batch size of 4. Gradient clipping with a maximum norm of 1.0 is applied for stable optimization. We train the network for 1 epoch on a total of $\sim 280K$ augmented samples, which takes $\sim$2 days on a single NVIDIA RTX A6000 GPU.

C.4 Inference Settings

To handle large-scale 3D scenes under GPU memory constraints, scenes with $N > 150$ images are partitioned into $\lceil N/150 \rceil$ chunks, whereas smaller scenes are processed entirely. For 3D scene reconstruction, input images are preprocessed with padding. For density map extraction, we set $\rho_{\text{conf}} = 45\%$, $\rho_{xz} = 2.5\%$, $\gamma = 0.5$, and $\left[\rho_y^{\min}, \rho_y^{\max}\right] = [20\%, 95\%]$. To estimate a 2D similarity transform $\mathbf{M}$ between a density map D and a floorplan F, we sample query points from D with probability proportional to the density values, and identify reliable correspondences by retaining the top 50% confidence points and applying mutual nearest neighbor matching. The remaining correspondences are used to estimate $\mathbf{M}$ via RANSAC [12]. We then apply $\mathbf{M}$ to the reconstructed 3D points $\mathcal{P}_*^{\text{g}}$ and their associated camera poses, yielding a floorplan-aligned 3D scene.

C.5 Structured3D Setup

For Structured3D [55], we adopt the same model hyperparameters, loss functions, and optimization settings as used for in-the-wild setup. We replace the training data with single-view samples from the Structured3D training set, where each density map is simply extracted using our inference settings (Sec. C.4) without augmentations. As this slows convergence, we train the network for 10 epochs.

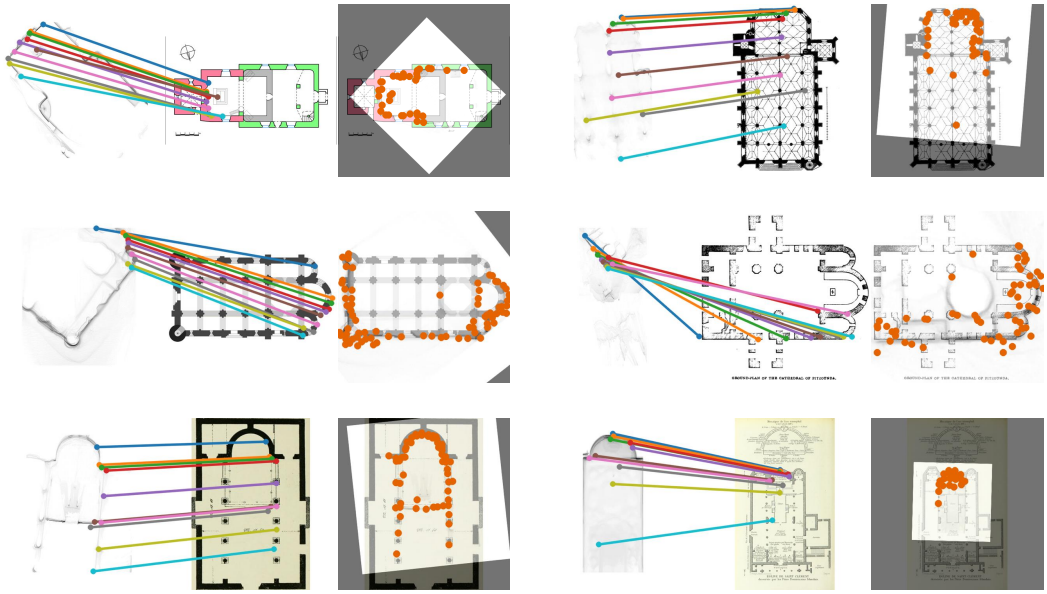


Figure D.1: **Floorplan alignment via correspondence estimation.** We predict a 2D similarity transform $\mathbf{M}$ for floorplan alignment by estimating correspondences between a density map and a reference floorplan. Here, we visualize 10 randomly sampled correspondences predicted by our model. Reliable correspondences used to compute the similarity transform are overlaid (in orange) on the aligned density map.

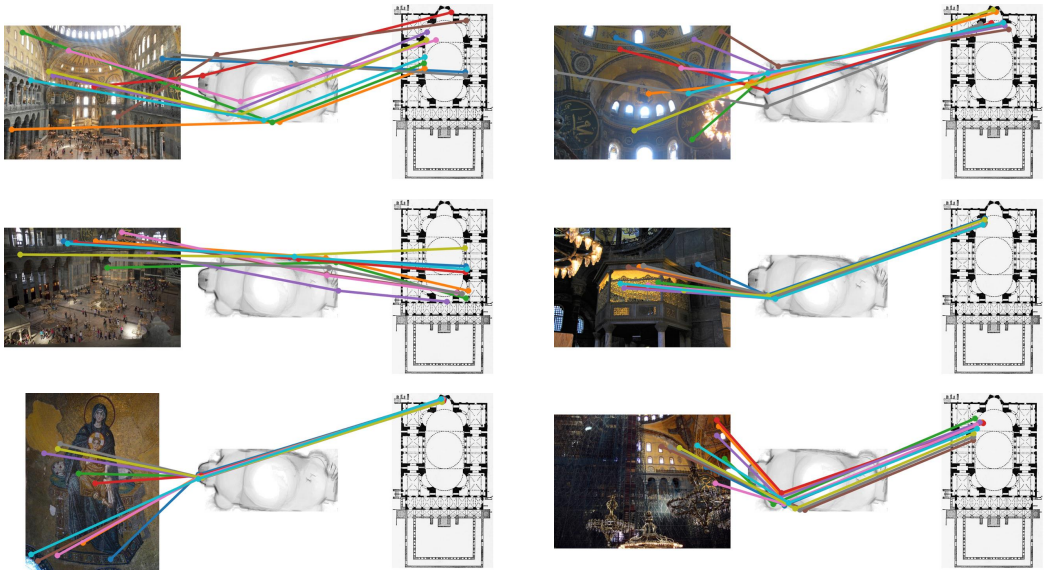


Figure D.2: **Photo-to-Floorplan correspondences via density map.** We localize pixel-level observations by aligning a density map to a reference floorplan. Here, we visualize 10 randomly sampled correspondences. The matches from photo to density map are obtained via 3D reconstruction, and those from density map to floorplan are predicted via cross-modal correspondence estimation.

D Additional Qualitative Results

In this section, we provide additional qualitative results of our method on in-the-wild and synthetic data. We visualize floorplan alignment via correspondence estimation (Figure D.1), photo-to-floorplan correspondences via density maps (Figure D.2), floorplan-aligned 3D scene reconstructions on C3 (Figure D.3), and single-view localization on Structured3D (Figure D.4).

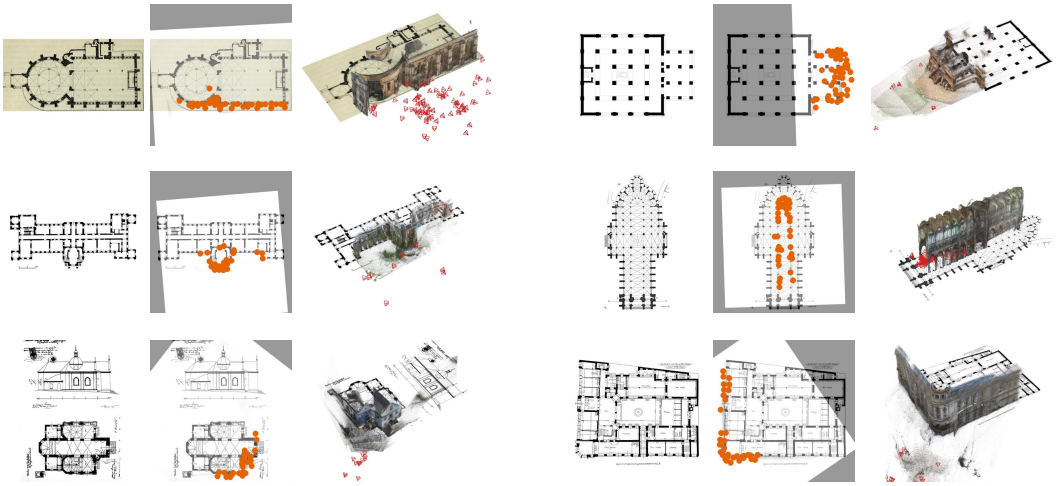


Figure D.3: **Floorplan-aligned 3D scene reconstructions.** For each sample, we show the input floorplan, aligned density map (with reliable correspondences overlaid in orange), and aligned scene.

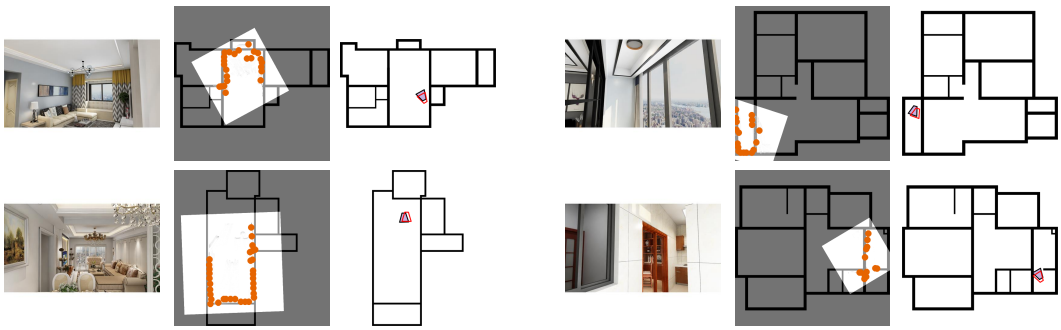


Figure D.4: **Qualitative results on Structured3D [55].** For each sample, we show the input photo, floorplan-aligned density map (with reliable correspondences overlaid in orange), and floorplan localization result. Cameras are illustrated as for GT and for Ours. Despite the limited visual context from a single image, our method successfully recovers camera poses by aligning the extracted density map to the reference floorplan.

As shown in Figure D.1, we predict correspondences between a density map and a reference floorplan to estimate a 2D similarity transform for floorplan alignment. The results demonstrate that our model can handle floorplans with varying target region sizes.

Figure D.2 shows how we estimate pixel-level correspondences between a photo and a floorplan through a density map. After we align the density map to the floorplan, we can localize pixel-level observations as we know the mapping from photo to density map (via 3D reconstruction) and from density map to floorplan (via correspondence estimation). Since the density map captures the global context of the scene, our method can effectively localize minimal-context photos as shown in the last row on the left.

Figure D.3 shows more floorplan-aligned 3D scene reconstructions on C3 [17]. Our method aligns in-the-wild reconstructions to floorplans of diverse drawing styles. We refer readers to our project page for 360° views of floorplan-aligned 3D reconstructions.

We also provide qualitative results on Structured3D [55] in Figure D.4. The results demonstrate that our method extends to single-view indoor localization, despite operating on rasterized floorplans without a discretized pose space or floorplan preprocessing.

Table E.1: **Effect of correspondence filtering.** We evaluate various filtering strategies to obtain reliable correspondences before estimating the similarity transform via RANSAC. $\texttt{top}_{50}$ indicates retaining the top 50% of correspondences ranked by the confidence w_i, and `MNN` denotes mutual nearest neighbor matching.

		Ang. ↑		Pos. ↑		Ang. & Pos. ↑	PCK ↑		
$\texttt{top}_{50}$	MNN	@ 5°	@ 30°	@ 5%	@ 20%	@ (30°, 20%)	@ 1%	@ 30%	RMSE ↓
		62.19	80.31	47.77	74.47	70.19	17.44	89.75	0.0832
✓		65.38	81.88	**51.16**	75.19	71.96	19.08	89.34	0.0806
	✓	65.66	**83.10**	48.19	74.60	71.67	17.63	89.62	0.0812
✓	✓	**65.91**	83.08	50.87	**76.08**	**73.58**	**20.42**	**90.45**	**0.0776**

Table E.2: **Effect of LoRA configurations.** We evaluate our model with different LoRA [16] rank r and scaling factor α. We set $r = \alpha = 16$ in our main experiments.

		Ang. ↑		Pos. ↑		Ang. & Pos. ↑	PCK ↑		
r	α	@ 5°	@ 30°	@ 5%	@ 20%	@ (30°, 20%)	@ 1%	@ 30%	RMSE ↓
4	4	50.10	75.70	39.22	66.66	62.40	13.86	86.49	0.1030
8	8	62.74	81.62	50.43	73.85	71.01	21.69	89.47	0.0794
16	16	65.91	**83.08**	**50.87**	**76.08**	**73.58**	20.42	**90.45**	**0.0776**
32	32	**65.93**	80.02	49.30	74.42	71.00	**22.70**	87.23	0.0873

Table E.3: **Effect of mixed-resolution training.** We evaluate the contribution of our mixed-resolution training strategy (`MR`), which improves localization accuracy.

	Ang. ↑		Pos. ↑		Ang. & Pos. ↑	PCK ↑		
MR	@ 5°	@ 30°	@ 5%	@ 20%	@ (30°, 20%)	@ 1%	@ 30%	RMSE ↓
	62.12	81.98	49.39	75.02	71.88	19.61	88.97	0.0832
✓	**65.91**	**83.08**	**50.87**	**76.08**	**73.58**	**20.42**	**90.45**	**0.0776**

E Additional Quantitative Evaluations on In-the-Wild Data

In this section, we provide additional evaluations to validate the contribution of our design choices (Sec. E.1), including correspondence filtering strategies (Table E.1), LoRA configurations (Table E.2), mixed-resolution training (Table E.3), and density map–floorplan matching (Table E.4). We also examine the robustness of our model against various hyperparameters (Sec. E.2).

E.1 Design Choices

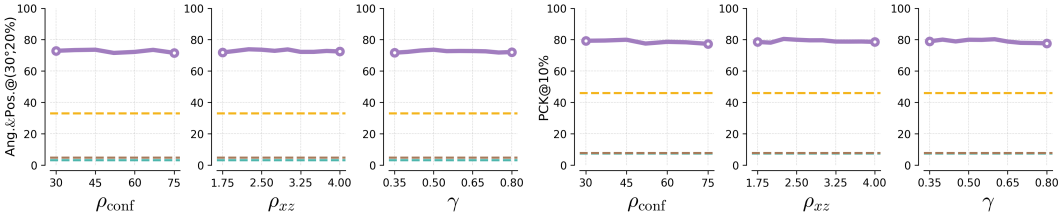
Inference Ablations. Before estimating the 2D similarity transform, we identify reliable correspondences to ensure a high-quality inlier set for RANSAC-based estimation. Specifically, we apply confidence-based pruning ($\texttt{top}_{50}$) to retain only the top 50% of correspondences ranked by the confidence w_i. In addition, mutual nearest neighbor (MNN) matching enforces bidirectional consistency. As demonstrated in Table E.1, these filtering strategies enhance alignment accuracy.

LoRA Configurations. Table E.2 shows the effect of LoRA [16] rank r and scaling factor α used to adapt DINOv3 [40] to our task. When the rank is too low (*e.g.*, $r = 4$), the adaptation capacity is limited, leading to noticeably degraded performance. Increasing the rank substantially improves the image-level and pixel-level localization accuracy. Among the evaluated settings, we observe that $r = \alpha = 16$ leads to the best results.

Mixed-Resolution Training. Table E.3 shows that training with varying input resolutions improves the image-level and pixel-level localization accuracy. We attribute this improvement to enhanced correspondence matching across floorplans with varying target region sizes. This strategy also serves as a form of data augmentation that helps mitigate overfitting during the training of our cross-modal correspondence estimation network.

Table E.4: **Ablations on matching between density map and floorplan.** We evaluate the contribution of LoRA-based fine-tuning (FT) and confidence-based filtering ($\texttt{top}_{50}$) on matching between the 2D density map and reference floorplan.

FT	$\texttt{top}_{50}$	PCK↑ @ 1%	@ 3%	@ 5%	@ 10%	@ 15%	@ 30%	RMSE↓
		0.79	4.60	9.13	20.01	28.58	53.39	0.2689
✓		10.54	48.74	61.67	69.97	73.81	87.10	0.1536
	✓	1.41	8.24	15.64	30.71	39.60	59.21	0.2690
✓	✓	**14.46**	**57.79**	**70.24**	**77.89**	**80.81**	**91.15**	**0.1290**



(a) Camera Pose Accuracy (b) Correspondence Accuracy

Figure E.1: **Stability across hyperparameters for density map extraction.** Our model remains robust across varying confidence percentile threshold ρ_{conf}, horizontal percentile threshold ρ_{xz}, and gamma value γ. C3Po, DUSt3R, and LoFTR are plotted as dashed lines for reference.

Ablations on Density Map and Floorplan Correspondence. The accuracy of floorplan alignment depends on the precision of correspondences between the density map and the reference floorplan. Table E.4 shows the contributions of our fine-tuning scheme and confidence-based filtering. Fine-tuning DINOv3 [40] substantially improves matching accuracy by adapting features originally trained on natural images to top-down architectural images. When we consider only the top 50% of correspondences ranked by the confidence w_i, matching accuracy is further improved. This observation motivates the use of confidence-based filtering for identifying reliable correspondences before predicting a 2D similarity transform.

E.2 Robustness to Density Map Hyperparameters

We evaluate the stability of our model against various hyperparameters used to derive density maps. Figure E.1 shows image-level and pixel-level localization accuracy when varying the confidence percentile threshold ρ_{conf}, horizontal percentile threshold ρ_{xz}, and gamma value γ. Notably, our approach consistently outperforms the baselines (C3Po [17], DUSt3R [47], and LoFTR [42]) by a significant margin. This result indicates that the proposed method does not necessitate exhaustive hyperparameter searches to achieve state-of-the-art results, demonstrating robustness to various hyperparameters for density map extraction.

F More Analyses

F.1 Sparse-View 3D Reconstruction Quality

In Section 4.2.2 and Figures 5-6 of the main paper, we demonstrate that our method achieves high accuracy even under sparse-view conditions. However, as discussed in Section 5, the proposed approach relies on 3D foundation models [49, 45]; geometric errors in reconstructed 3D points may propagate to density maps and thus affect floorplan alignment accuracy. To quantify the geometric degradation caused by view sparsity, we analyze how much 3D geometry reconstructed from sparse-view inputs deviates from dense-view reconstructions. In this context, we consider the dense-view reconstructions as the pseudo ground truth. The results are summarized in Table F.1.

Table F.1: **Evaluation of 3D reconstruction quality from in-the-wild sparse views.** We measure the geometric degradation of 3D points reconstructed from varying numbers of sparse input images (1 to 10). Dense-view reconstructions (≤ 150 images) serve as the pseudo ground truth. All 3D points are estimated using π^3 [49], then normalized and aligned via ICP.

				F-Score ↑			
# Imgs	Accuracy ↓	Completeness ↓	Overall ↓	@ 1%	@ 3%	@ 5%	@ 10%
$= 1$	1.1470	0.9222	1.0346	14.81	32.41	38.87	46.61
$= 3$	0.1178	0.0913	0.1046	27.82	54.65	63.93	74.38
≤ 5	0.0624	0.0553	0.0588	33.88	64.62	74.53	84.51
≤ 10	**0.0303**	**0.0331**	**0.0317**	**43.25**	**76.10**	**85.20**	**92.81**

Table F.2: **In-the-wild camera pose estimation by scene view type.** Our method is more accurate on interior scenes than on exterior scenes. The "All" row reports the aggregated accuracy.

	Angular Recall ↑				Positional Recall ↑			Ang. & Pos. ↑
Type	@ 5°	@ 10°	@ 20°	@ 30°	@ 5%	@ 10%	@ 20%	@ (30°, 20%)
Interior	**73.84**	**81.24**	**83.49**	**87.11**	**64.84**	**81.22**	**92.62**	**84.89**
Exterior	62.96	73.95	79.01	81.58	45.66	59.75	69.91	69.36
All	65.91	75.93	80.23	83.08	50.87	65.58	76.08	73.58

Evaluation Protocol. To isolate the geometry of reconstructed 3D points from global camera pose drifts, we align 3D points from sparse-view and dense-view reconstructions. Specifically, we transform them into a shared camera frame using relative camera extrinsics, and further align them via Iterative Closest Point (ICP) registration. For consistent comparison, both 3D points are centered and scaled using the bounding box of the dense-view points.

Metrics. We report *Accuracy*, *Completeness*, *Overall*, and *F-Score* using $10K$ points randomly sampled from sparse-view and dense-view reconstructions. Let $\mathcal{P}_{\text{pred}}$ be the sampled points from the sparse-view reconstruction and $\mathcal{P}_{\text{gt}}$ be the sampled points from the dense-view reconstruction. Accuracy is defined as the mean Euclidean distance from points in $\mathcal{P}_{\text{pred}}$ to their nearest neighbors in $\mathcal{P}_{\text{gt}}$. Completeness measures the mean Euclidean distance from points in $\mathcal{P}_{\text{gt}}$ to their nearest neighbors in $\mathcal{P}_{\text{pred}}$. Overall is defined as the average of Accuracy and Completeness, which is equivalent to the Chamfer Distance. F-Score measures the harmonic mean of precision and recall at distance thresholds, representing the concordance between points in $\mathcal{P}_{\text{pred}}$ and $\mathcal{P}_{\text{gt}}$.

Discussion. As shown in Table F.1, the geometric quality of sparse-view reconstructions degrades as the number of input images decreases. While reconstructions with ≤ 10 images closely approach the dense-view pseudo ground truth, the gap substantially increases under fewer views, with single-view reconstructions exhibiting the largest deviation. This trend is consistent with the accuracy drop observed in the main paper, where limited geometric context contributes to alignment ambiguity.

F.2 Evaluation by Scene View Type

We analyze our method separately on interior and exterior scenes from the C3 dataset [17]. As shown in Tables F.2 and F.3, our method achieves notably higher accuracy on interior scenes. We attribute this gap to inherent challenges of exterior scenes, where 3D reconstructions are noisier and often include far-away points from distant background regions. These factors introduce artifacts into the resulting density map, making cross-modal correspondence estimation more challenging.

F.3 Inference Time Comparison

We compare the end-to-end inference time of our method against C3Po [17] on a single NVIDIA RTX A6000 GPU, with results summarized in Table F.4. To reflect real-world deployment efficiency, we report the *amortized cost* per image. For a detailed analysis, we decompose the inference time into four stages: (i) gravity-aligned 3D scene reconstruction, (ii) density map extraction, (iii) correspondence estimation, and (iv) camera pose estimation in the floorplan coordinate frame.

Table F.3: **In-the-wild correspondence estimation by scene view type.** Our method is more accurate on interior scenes than on exterior scenes. The "All" row reports the aggregated accuracy.

	PCK ↑						
Type	@ 1%	@ 3%	@ 5%	@ 10%	@ 15%	@ 30%	RMSE ↓
Interior	**24.66**	**63.47**	**74.43**	**84.18**	**87.06**	**95.09**	**0.0550**
Exterior	18.84	56.14	68.17	78.36	82.37	88.73	0.0860
All	20.42	58.13	69.87	79.94	83.64	90.45	0.0776

Table F.4: **Amortized inference time on C3 [17].** We report the amortized end-to-end processing time per image, decomposed into four stages. Ours ($\leq N$) processes input photos in chunks of at most N images. Since the number of available photos varies per scene, the actual chunk size differs accordingly. For reference, we also report the mean chunk size.

	Mean	Amortized Inference Time per Stage ↓				
Method	Chunk Size	3D Scene	Density Map	Correspondence	Camera Pose	Total ↓
C3Po [17]	1.0	—	—	59.6 ms	135.8 ms	195.4 ms
Ours ($= 1$)	1.0	158.8 ms	20.2 ms	17.2 ms	0.2 ms	196.4 ms
Ours (≤ 10)	9.5	**121.8 ms**	**20.0 ms**	1.9 ms	**0.0 ms**	**143.7 ms**
Ours (≤ 150)	66.4	200.2 ms	22.8 ms	**0.4 ms**	**0.0 ms**	223.4 ms

Our approach achieves a substantial speed advantage over C3Po in the latter two stages. During correspondence estimation, C3Po relies on a heavy ViT-L backbone for processing every single photograph. In contrast, we use a lightweight ViT-B backbone (initialized from DINOv3 [40]) and estimate correspondences *once* per density map, thereby effectively amortizing the cost across up to N images. During camera pose estimation in the floorplan frame, C3Po solves epipolar geometry to produce candidate camera poses and selects the one closest to the ground truth. In contrast, we apply a 2D similarity transform to the reconstructed camera poses, incurring only negligible overhead.

We evaluate our method with maximum chunk sizes of $N \in \{1, 10, 150\}$ and identify Ours (≤ 10) as the sweet spot, which is notably faster than C3Po (143.7 ms vs. 195.4 ms). At $N = 1$, each input photo is processed sequentially and therefore the GPU is severely underutilized. At $N = 150$, the quadratic complexity of transformer layers in the 3D reconstruction model becomes the dominant bottleneck. At $N = 10$, these overheads are balanced, yielding the best trade-off.

F.4 Validity of the Density Map as a Floorplan Proxy under Sparse Views

Sparse inputs often capture sufficient structure for matching between the density map and the floorplan, as illustrated by the sparse- and single-view examples in Figure 6 and Figure D.4. Our quantitative results on C3 (Table B.2) further show that our method outperforms prior work in the single-view setting. To further clarify when the density map extracted from such limited input becomes ambiguous and when additional views are needed, we conduct a study in the single-view setting, the most extreme sparse-input case.

Matching Confidence. We examine whether the matching confidence w_i (Sec. 3.3), obtained by matching the density map against the reference floorplan, indicates when the current observation is sufficient for a reliable global alignment. Low confidence scores are typically associated with observed structure that is repeated or symmetric, for which matching is ambiguous. For each test image, we compute an image-level confidence as the median of the per-query w_i over the query points sampled from the density map at inference. We then rank the images by this score, splitting them evenly into low, medium, and high confidence groups.

As shown in Table F.5, localization accuracy increases monotonically with matching confidence, from 35.6 to 55.8. The high-confidence group is comparable to Ours ($= 3$), which reaches 56.28 (Table B.2), indicating that a single confident observation can constrain the alignment about as well as a three-view reconstruction. Since w_i is produced by the matcher itself, it is available at inference time. When confidence is low, we can add additional views and extract the density map from the combined geometry.

Table F.5: **Single-view localization by matching confidence and spatial extent on C3 [17].** Test images are ranked by the median matching confidence w_i (left) or by the spatial extent covered by the single-view reconstruction (right), and split evenly into low, medium, and high groups.

	Matching Confidence		Spatial Extent	
Group	Median Confidence	Ang. & Pos. @ $(30°, 20\%)\uparrow$	Median Coverage	Ang. & Pos. @ $(30°, 20\%)\uparrow$
Low	0.23	35.6	7.7%	33.4
Medium	0.45	48.3	22.2%	49.3
High	0.65	**55.8**	44.1%	**57.0**

Table F.6: **Scene-level matching confidence on C3 [17].** We report the median matching confidence w_i and single-view localization accuracy for the exterior images of two representative scenes.

Scene	Median Confidence	Ang. & Pos. @ $(30°, 20\%)\uparrow$
Church of St. Donatus in Zadar (exterior)	0.16	9.7
Santa Maria Maggiore in Rome (exterior)	0.79	**88.5**

Table F.7: **Sensitivity to reconstruction quality and gravity error on Structured3D [55].** Point sparsity, reconstruction noise (standard deviation σ), and gravity error (angle θ) are each simulated on ground-truth 3D points.

Point Sparsity		Reconstruction Noise		Gravity Error	
	Ang. & Pos. ↑		Ang. & Pos. ↑		Ang. & Pos. ↑
Retained Points (%)	@ (30°, 1m)	σ (cm)	@ (30°, 1m)	θ (°)	@ (30°, 1m)
100	57.2	0	57.2	0	57.2
50	56.3	0.25	56.2	0.5	57.2
20	54.3	0.5	53.8	1	57.0
10	52.5	1	47.0	2	54.9
5	49.1	1.5	38.3	5	50.2
2	45.0	2	29.8	10	40.3

We further examine this at the scene level using the exterior images of two representative scenes (Table F.6). For the Church of St. Donatus, the C3 images cover a rotunda whose rotationally symmetric structure admits many candidate alignments. For Santa Maria Maggiore, they mostly cover the facade, a much more distinctive structure with a unique counterpart in the floorplan. The predicted confidence therefore tracks the structural ambiguity of what is observed, which is what makes it usable directly as an inference-time signal.

Spatial Extent. We also quantify the spatial extent covered by each reconstruction. We define it as the fraction of the scene's full multi-view footprint that the single-view reconstruction covers. For this analysis, we place both footprints in the floorplan coordinate frame using their ground-truth alignments, since the two reconstructions otherwise differ in coordinate frame and scale, preventing direct comparison of spatial extent. Ground truth is used only for this measurement and is never provided to the localization method.

As shown in Table F.5, spatial extent is likewise strongly associated with localization accuracy. This indicates that additional views are especially important when a single observation covers only a limited extent of the scene. Measuring such scene coverage requires ground-truth alignments, whereas grouping by matching confidence spans a similar range of accuracy without them.

F.5 Sensitivity to Reconstruction Quality and Gravity Error

We conduct a controlled sensitivity analysis where point sparsity, reconstruction noise, and gravity error are each simulated on ground-truth data, measuring the resulting localization accuracy (Table F.7). We use Structured3D [55] for this analysis, since its ground-truth geometry is exact, metric-scale, and gravity-aligned by construction, so each factor can be varied by a known amount from an unperturbed ground-truth reference.

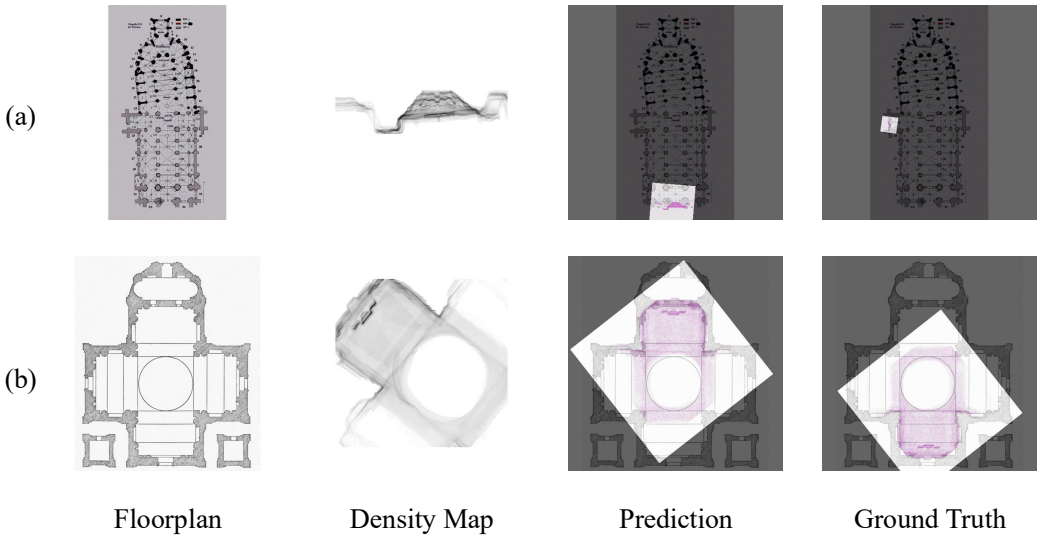


Figure G.1: **Failure cases.** (a) Structural ambiguity. (b) Symmetrical ambiguity. For each row, we show the reference floorplan, extracted density map, predicted floorplan alignment, and ground-truth alignment. For visualization purposes, pink dots overlaid on the density map serve as reference points to visualize alignment quality (the same dots are illustrated in both alignment results).

Point Sparsity. We randomly subsample the 3D points to a decreasing fraction, leaving the positions of the surviving points exact. Accuracy is well preserved down to 10% of the points, and even at 2% it still reaches 45.0.

Reconstruction Noise. We displace every 3D point by isotropic Gaussian noise of standard deviation σ, leaving the number of points unchanged. Accuracy is largely preserved up to $\sigma = 0.5$ cm and degrades gradually thereafter. To relate these values to practice, our method reaches 51.6 with π^3 [49] reconstructions on this benchmark (Table 6). We note that independent per-point noise perturbs local surface structure far more aggressively than the spatially correlated error of reconstructions, so our method would tolerate considerably more error in practice than these thresholds suggest.

Gravity Error. We rotate the reconstruction so that its vertical axis deviates by an angle θ in a random direction. Accuracy is unaffected up to 1°, and a 10° error retains 40.3. To relate these values to practice, we measured the gravity estimate of our pipeline against ground truth on C3 [17] and found a median error of 1.4°, which lies within the range where localization is essentially unaffected.

Taken together, this controlled sensitivity analysis shows that point sparsity and gravity error have limited effect on localization accuracy, while reconstruction noise is the dominant sensitivity under the synthetic perturbation used here.

G Limitations

In Section 5 of the main paper, we noted that errors from 3D reconstruction models can degrade the quality of subsequent density maps and floorplan alignment. Beyond this, our method can also fail even with accurate 3D reconstructions, due to inherent ambiguities in cross-modal correspondence between density maps and floorplans.

Figure G.1 illustrates two such failure modes. The first arises from *structural ambiguity*, where the density map captures only a local region (*e.g.*, a doorway in row (a)) that matches multiple visually similar structures on the floorplan. Without sufficient surrounding context, the model cannot uniquely determine which of these structures the density map corresponds to, which results in incorrect alignments. The second arises from *symmetrical ambiguity*, where the density map captures a partially observed symmetric structure (*e.g.*, three arms of a cross-shaped structure in row (b)). In such cases, multiple alignments can satisfy the local geometry equally well, and the model may choose an orientation that differs from the ground truth by 180°.

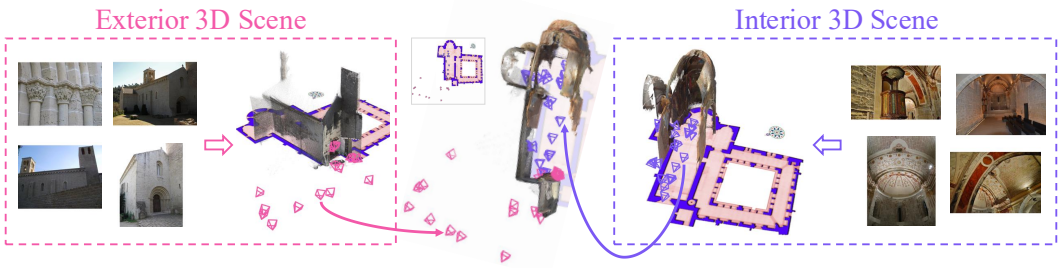


Figure H.1: **Alignment of interior and exterior 3D scenes.** Using a reference floorplan as the shared geometric anchor, our method can align reconstructions from interior and exterior photos into a unified global coordinate system, despite minimal visual overlap and large viewpoint differences.

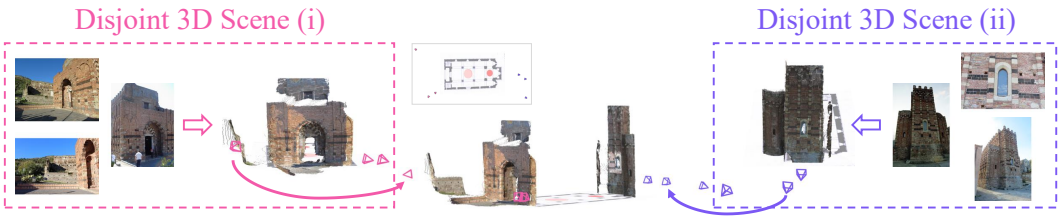


Figure H.2: **Alignment of disjoint 3D scenes.** Leveraging a reference floorplan as the shared geometric bridge, our method can align reconstructions from disjoint image collections into a unified global coordinate system, even without visual overlapping.

Both failure modes stem from a lack of global context in the density map. As such, they tend to be more pronounced under sparse-view settings (Section F.1), where the density map captures only a partial view of the scene. While our method already remains effective in most scenarios, addressing these ambiguities is an interesting direction for future work.

Scope of the 2D Similarity Transform. Our method aligns the density map to the floorplan with a 2D similarity transform. Distorted or non-metric floorplans (*e.g.*, photographed or schematic drawings) may require a projective or non-rigid transform. Incomplete floorplans may remain tractable given sufficient structural overlap, consistent with the gradual degradation in the coverage analysis (Sec. F.4). Multi-level buildings with different layouts across floors require floor-specific floorplans and observations, as in existing floorplan-localization settings.

H Applications

Interior-Exterior 3D Scene Alignment. State-of-the-art 3D foundation models [45, 49] often fail to produce a coherent global reconstruction when jointly processing interior and exterior photos, primarily due to drastic viewpoint changes and the lack of sufficient visual overlap. By using a reference floorplan as a geometric anchor, our method can align separate reconstructions into a unified global coordinate system. We further showcase this application over reconstructed interior and exterior 3D scenes capturing the Monestir de Sant Benet de Bages (a medieval monastic complex in Catalonia) in Figure H.1.

Disjoint 3D Scene Alignment. Similarly, 3D foundation models frequently fail to register disjoint image collections that lack overlapping regions. Our method can address this by independently aligning disjoint reconstructions into a unified global coordinate system, as illustrated over partial reconstructions capturing Santi Pietro e Paolo d'Agrò (a church in Italy) in Figure H.2.

I Broader Impacts

Our work enables localizing in-the-wild photographs within reference floorplans, which can support a wide range of applications, including scene navigation, augmented or virtual reality, and architectural understanding. By aligning unconstrained photo collections to floorplans, our method also facilitates downstream tasks such as interior-exterior 3D scene alignment and disjoint 3D scene alignment, which can be valuable for large-scale 3D reconstruction in challenging scenarios.

We do not expect significant negative societal impacts from this work. Our method operates on photographs and architectural floorplans, which are widely available and commonly used in geographic information systems. Our experiments use only the publicly available C3 dataset [17] and synthetic Structured3D [55] renderings, and the density map used for alignment retains no visual appearance. Nonetheless, as with any localization technology, responsible use is essential when applying it to private spaces. Our method is intended for research and non-surveillance use, and deployment in private spaces requires the property owner's informed consent. We encourage users to follow established guidelines when deploying our localization method in real-world scenarios.

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 abstract and introduction accurately reflect our contributions: reframing floorplan localization as a 3D reconstruction and alignment problem, leveraging foundation models with our fine-tuning scheme for cross-modal correspondence estimation, and achieving the state of the art on both in-the-wild and synthetic indoor settings.

   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 discuss limitations in Section 5 of the main paper and Section G of the appendix, where we describe the dependency on 3D foundation models and two failure modes caused by structural and symmetrical ambiguities.

   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: This paper does not provide theoretical results. Our work focuses on a practical method for floorplan localization by estimating cross-modal correspondences between density maps and floorplans, rather than providing theoretical guarantees.

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: We provide comprehensive implementation details in Section 4.1 of the main paper and Section C of the appendix, including model architecture, hyperparameters, data augmentation strategies, optimization configurations, and inference settings.

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: Our code, model weights, and data are publicly available at `https://github.com/Cornell-VAILab/SceneAligner`, with instructions for installation, data preparation, training, and evaluation on C3 and Structured3D. Implementation details are also provided in Section 4.1 of the main paper and Section C of the appendix.

   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: We provide comprehensive information on training and test details, including dataset statistics, hyperparameters, optimization configurations, and inference settings, in Section 4.1 of the main paper and Sections B and C of the appendix.

   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: Following standard practice in floorplan localization literature [32, 6, 53] and correspondence estimation literature [17, 47], we report evaluation results from a single run rather than error bars due to the substantial computational cost of multiple training runs.

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: We provide information on computation resources in Section C.3 (*e.g.*, training takes approximately 2 days on a single NVIDIA RTX A6000 GPU).

   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: We have carefully reviewed the NeurIPS Code of Ethics and ensure that our research is conducted in accordance with it.

   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 discuss potential positive and negative societal impacts in Section I of the appendix.

    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: Our work does not pose such risks, as we use publicly available datasets and do not release generative models.

    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: We properly cite all assets used in our work (*e.g.*, DINOv3 [40], π^3 [49], C3 dataset [17]), which are used in accordance with their respective licenses and terms of use.

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: [N/A]

Justification: Our work does not introduce new assets. We construct a clean subset of the existing C3 dataset [17], which is publicly available along with our code.

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: Our work does not involve crowdsourcing experiments 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: Our work does not involve research with human subjects.

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: We do not use LLMs as a core component of our methodology. Qwen2.5-VL [1] is used only as a tool for dataset curation (Sec. B), which does not impact the core methodology.

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.