MvFFN: Multi-view Floor plan Feed-Forward Network for Unposed Wide-Baseline Panorama Layout Reconstruction

Yuguang Li[1]*✉ **Yichuan Deng**[1]* **Zixuan Liu**[1]* **Ivaylo Boyadzhiev**[2]
Xiuyuan Wu[1] **Xinyu Zou**[1] **Linda Shapiro**[1] **Alex Colburn**[1]
[1]University of Washington [2]Independent Researcher

Abstract

Reconstructing accurate camera poses and floor plan layouts from sparse, unposed, wide-baseline RGB panoramas remains a challenging open problem. Direct multi-room floor plan prediction is especially difficult in this setting: pose, geometry, semantics, and global layout structure are tightly coupled, yet a complete floor plan is hard to learn as a single global output. We address this challenge with *Multi-view Ordered Wall Instance Segmentation* (*MOIS*), a new formulation that replaces direct global wall-sequence prediction with local cross-view wall instances and within-room wall connectivity. Building on this formulation, we propose *MvFFN*, a feed-forward multi-view transformer that jointly predicts coarse camera poses, dense global geometry, and wall-instance-level semantic outputs with globally consistent identities and within-room connectivity. A downstream *Layout Chaining* (*LC*) post-processing pipeline then assembles these predictions into multi-room floor plans with shared global coordinates, consistent wall identities, and connected room structure. The resulting pipeline substantially outperforms prior methods on wall- and junction-level accuracy. Moreover, the predicted wall instances enable fully automatic downstream layout refinement, with planar bundle adjustment improving a majority of scenes and yielding clear gains on the subset that benefits. We also introduce `ZInD-CrossView`, a re-annotated dataset with globally unique wall instances and cross-view segmentation labels for this task. Visit the project website at: https://yuguang-li.github.io/mvffn.

1 Introduction

Reconstructing floor plan layouts and camera poses from panoramic images is important for applications such as augmented reality, indoor digital twins, immersive walkthrough generation, and autonomous navigation. Recovering complete multi-room layouts requires multiple views because of occlusions and limited visibility. However, existing methods typically avoid the setting of unposed, wide-baseline panoramas: they assume known camera poses Yue et al. [2023], Avetisyan et al. [2024], Liu et al. [2024], require small-baseline captures Avetisyan et al. [2024], or rely on human-in-the-loop adjustment Cruz et al. [2021], Li et al. [2025]. A fully automatic approach for reconstructing layouts and poses from unposed, wide-baseline panoramas would substantially reduce capture and annotation effort and enable broader deployment at scale.

A floor plan encodes wall geometry, local connectivity, and room-level organization. Recovering it from multiple views requires associating wall instances across views, inferring their connectivity despite occlusion, and separating structures that belong to different rooms. Geometry alone is often insufficient. When camera poses are unknown and fused point maps are noisy, reconstruction often

*Equal contribution. ✉ Corresponding author: `ylee3@cs.washington.edu`.

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

suffers from duplicate walls, missing walls, and incorrect junctions, which reduce both wall and junction-level precision and recall. In contrast, images provide critical semantic cues for grouping pixels across views by wall instance, inferring local connectivity, and resolving room assignments. At the same time, directly predicting a complete multi-room floor plan as a single global output is difficult to learn. Large multi-room layouts often contain ambiguous room partitions, limited observations can obscure small structural details, and small local sequence errors can cascade into large global drift. The problem is further compounded by the scarcity of datasets that provide structured multi-room layouts together with camera-registered multi-view panoramas. This motivates the central insight of our paper: rather than predicting a global floor plan directly, we recast multi-view, multi-room layout estimation around local wall relations that are easier to supervise and more robust to noisy annotations.

To this end, we study structured floor plan and camera-pose reconstruction from sparse, unposed 360° panoramas, producing complete layouts when view coverage is sufficient and partial layouts under occlusion. We formalize this idea as *Multi-view Ordered Wall Instance Segmentation* (*MOIS*), which replaces direct global floor plan prediction with local wall instances and within-room wall connectivity that remain consistent across views. Based on *MOIS*, we propose *MvFFN*, a feed-forward multi-view network that predicts pixel-grounded wall instances and their within-room connectivity from cross-view geometric and semantic cues, and a *Layout Chaining* (*LC*) post-processing pipeline that assembles these local predictions into multi-room floor plans with consistent wall identities in a shared global frame. *MvFFN* also predicts coarse camera poses and dense global geometry, while the inferred wall instances enable downstream layout bundle adjustment for further refinement. Together, these design choices make multi-view layout estimation more learnable under scarce data and noisy supervision. Accordingly, the learned contribution is *MOIS*/*MvFFN*, while global assembly remains deterministic and the geometric priors address current data scarcity; Appendix I details these limitations and future directions.

The contributions of this work are as follows: (1) We introduce *MOIS*, a new formulation for multi-view, multi-room layout estimation based on local cross-view wall instances and within-room wall connectivity. (2) We introduce *MvFFN*, a feed-forward multi-view network that predicts these local wall structures from cross-view geometric and semantic cues. We show gains both before and after layout assembly: dense *MOIS* predictions improve raw cross-view wall-slot quality, while the assembled global layouts substantially improve wall and junction accuracy over prior modular baselines across both 5-view and 8-view settings. (3) We introduce `ZInD-CrossView` (`ZInD-XV`) dataset with globally unique wall instances and cross-view segmentation annotations.

2 Related Work

Layout estimation. Most layout estimation methods combine a dense geometric front-end with a back-end that vectorizes predictions into wall and corner instances and room topology Sun et al. [2021a]. In the single-view setting, methods predict monocular 1D floor boundaries Yang et al. [2019], Wang et al. [2021], Jiang et al. [2022], Sun et al. [2021b], Shen et al. [2023], Tsai et al. [2024] and recover wall and corner sequences through post-processing Zou et al. [2018], Sun et al. [2019], Pintore et al. [2020], relying on the clockwise image-column order within a single panorama. This cue does not directly extend to sparse multi-view settings, where each view observes only a partial layout under occlusion. For multi-room floor-plan estimation, most methods instead assume accurate camera poses and dense point clouds, and infer structure mainly from fused geometric cues in top-down space. *Plane-DUSt3R* Huang et al. [2025] predicts structural-plane point maps and recovers wall instances and topology through post-processing. Under imperfect geometry, connectivity is typically recovered with heuristic optimization Chen et al. [2019], Wang et al. [2022], Hanning et al. [2025] or learned polygon and graph decoders Liu et al. [2024], Yue et al. [2023], Chen et al. [2023], Su et al. [2023], Pintore et al. [2025]. By contrast, cross-view semantic cues remain underused for resolving wall identity and connectivity under pose uncertainty. *SceneScript* Avetisyan et al. [2024] shows that, given camera poses and RGB video, semantic context can improve structured layout estimation.

Wide-baseline pose and geometry estimation. Wide-baseline indoor pose estimation remains difficult for traditional *Structure-from-Motion* Schonberger and Frahm [2016], Moulon et al. [2016], Lambert et al. [2022]. Recent learning-based models improve dense 3D geometry and camera pose estimation from wide-baseline inputs Wang et al. [2024, 2025], Tang et al. [2025], Jung et al. [2025]. For indoor 360° panoramas, limited training data often constrains geometry estimation to 1D floor boundary representations rather than dense per-pixel 3D geometry Hutchcroft et al. [2022], Tu et al. [2024], Nejatishahidin et al. [2023]. *BADGR* Li et al. [2025] shows that semantic

layout correspondences and geometry are complementary: once pixel-to-wall correspondences and initial estimates are available from unposed wide-baseline images, poses and layouts can be jointly optimized to reduce initial errors.

3 Overview

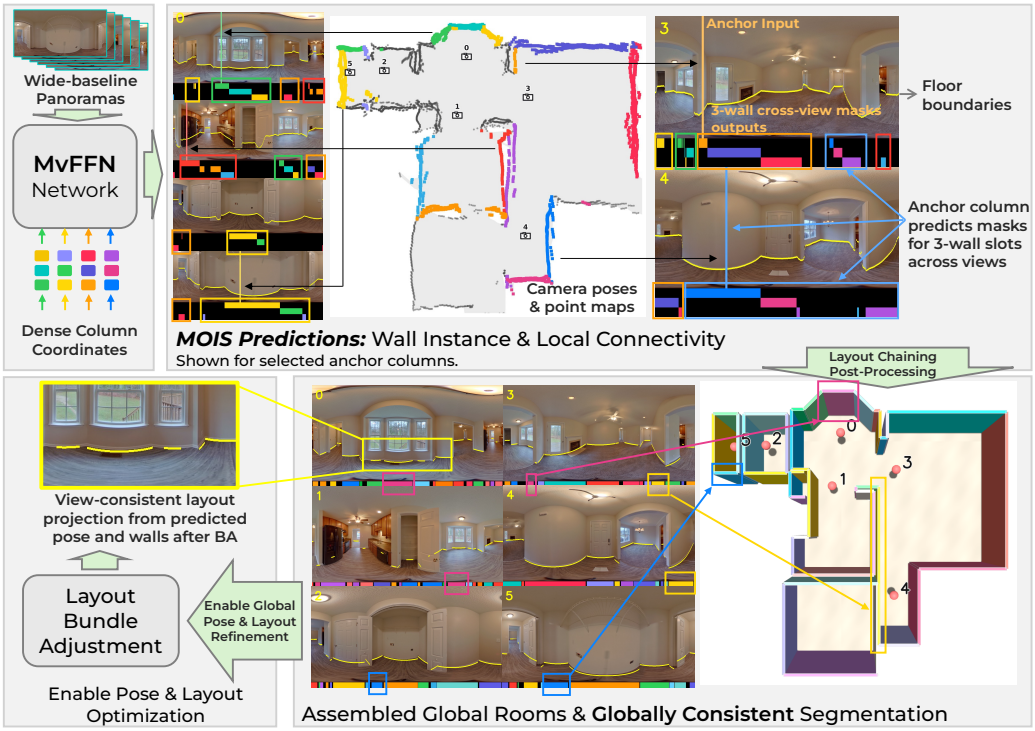


Figure 1: *MvFFN* reformulates floor plan reconstruction from direct global prediction to local *MOIS* prediction: each anchor column, shown as a vertical line in the panorama, predicts cross-view masks for the anchor wall and two clockwise adjacent walls. Colors indicate corresponding wall slots/instances across panoramas and in the assembled layout. *Layout Chaining* aggregates these local predictions into global wall identities, room-level wall sequences, and metric floor plan layouts, enabling downstream refinement.

Large multi-room floor plans often contain ambiguous room boundaries, occluded wall connectivity, and distant structural details, making direct floor-level sequence prediction brittle to error accumulation. We therefore reconstruct layouts through dense local *MOIS* predictions rather than direct global floor plan prediction.

Our bottom-up pipeline (Figure 1) takes N panoramas and evaluates *MvFFN* densely at every column of every view. Each anchor column defines one *MOIS* target: the cross-view assignments of the current wall and its within-room neighbors. Aggregating all anchors yields dense anchor-conditioned *MOIS* assignments over multiple views, predicted jointly with global geometry under pose uncertainty.

MvFFN uses a multi-view geometric encoder, trained from scratch and inspired by *VGGT* Wang et al. [2025], adapted to panoramic 2D column tokens and dense column-wise outputs. A floor-layout semantic branch combines image and geometric features for local wall-instance reasoning. We describe the geometric encoder and floor-layout decoder in Sec. 4.1 and Sec. 4.2, respectively.

Finally, the dense local *MOIS* predictions are assembled into global room layouts via *Layout Chaining* (Sec. 4.5). The predicted pixel-to-wall correspondences and initial layouts can also support downstream layout refinement, which we evaluate with layout bundle adjustment in Sec. 4.6.

4 Method

We formulate multi-room layout reconstruction from sparse panoramas Zhang et al. [2014] as the recovery of global 3-DoF camera poses (2D position and 1D rotation), structured 2D floor layouts, and a local anchor-conditioned wall representation across views.

Inputs. We assume a set of N straightened 360° RGB panoramas $\{I_i\}_{i=1}^N$ Zhang et al. [2014], where each $I_i \in \mathbb{R}^{3\times H\times W}$ is given in equirectangular projection. The panoramas are sparsely captured in an indoor environment and are assumed to form a single pose graph whose neighboring views have at least 20% co-visibility similar to Cruz et al. [2021], Hutchcroft et al. [2022], Li et al. [2025]. We further assume that relative camera heights are known across the floor.

***MvFFN* outputs.** For each panorama I_i, *MvFFN* predicts four intermediate quantities: (1) a 3-DoF camera pose $E_i \in SE(2)$; (2) dense per-column floor boundaries $\mathcal{B}_i \in \mathbb{R}^{2\times W}$; (3) dense per-column floor point maps $\mathcal{P}_i \in \mathbb{R}^{4\times W}$ and wall-line parameters $\Lambda_i \in \mathbb{R}^{3\times W}$; and (4) for each anchor column, a dense *MOIS* probability field $\hat{\mathcal{S}}$ over columns across all views, encoding assignments to the underlying global wall instance intersected by the anchor ray and its within-room neighboring wall slots.

***MOIS* formulation.** We formalize the local wall-instance and within-room connectivity predictions described above with *MOIS*, an anchor-conditioned ordered wall representation defined for each image column (see Fig. 1 for intuition). Let $\mathcal{C}_{i,c}$ denote anchor column c in panorama I_i. The anchor ray determines the current room by the first wall it intersects: if it intersects a closed door, the door is assigned to the attached wall instance and the current room remains the attached room; if it passes through an open doorway, the current room is the room containing the first visible wall it intersects. The corresponding *MOIS* target is then a local ordered K-wall tuple

$$\mathbf{w}_{i,c} = \left(w_{i,c}^{(0)}, w_{i,c}^{(1)}, \ldots, w_{i,c}^{(K-1)}\right),$$

where $w_{i,c}^{(0)}$ is the intersected wall instance defining the current room, and $w_{i,c}^{(r)}$ denotes the r-th clockwise physically adjacent wall within that room. We convert this local tuple into a dense binary assignment tensor $\mathcal{S}^{\mathrm{gt}}$, where each entry indicates whether column (j, d) is assigned to a given ordered wall slot of the anchor column. This dense local representation serves as the semantic target for the floor-layout semantic branch in Sec. 4.2 and is assembled into global layouts by *Layout Chaining* in Sec. 4.5.

Final layout outputs. The primary outputs are 3-DoF camera poses $\{E_i\}_{i=1}^N$ and global room layouts $\{\mathcal{W}^m\}$. Unlike the local anchor-conditioned *MOIS* tuples, each $\mathcal{W}^m$ represents the final reconstructed wall geometry of room m in the global floor coordinate frame after *Layout Chaining*. We also maintain an auxiliary column-to-wall map $\mathcal{M} : \mathcal{C}_{i,c} \to w^\ell$, which associates each image column with at most one global wall instance. The room corners $V^m = (v^{m,1}, \ldots, v^{m,K_m})$ are obtained by intersecting consecutive wall segments, where $v^{m,k} \in \mathbb{R}^2$ and K_m is the number of corners. Depending on visibility, a room layout may be closed or remain open under occlusion.

4.1 N-view Geometry Encoder Branch

The geometry branch is a multi-view encoder for dense geometric prediction, including camera pose, floor-boundary depth, and point maps. Architecturally, it is similar to *VGGT* Wang et al. [2025], while operating on panoramic column-wise features. Each panorama I_i is processed by an *EfficientNet-v2* backbone Tan and Le [2021], followed by a Height Compression Module (HCM) Sun et al. [2019] that produces column-patch features $f_i^G \in \mathbb{R}^{S\times C}$, where $S = 256$, $C = 1024$, and each patch spans $4 \times H$ pixels.

Patch encodings. We add angular column embeddings (ACE) and sinusoidal camera positional encodings (CPE) Vaswani [2017] to obtain geometry tokens F^G. The ACE are *Fourier features* Tancik et al. [2020] of per-patch viewing angles. For each column-patch, the viewing angle combines: (1) an intra-frame horizontal angle $\theta_{i,c}^h \in [0^\circ, 360^\circ)$ corresponding to the patch-center azimuth, and (2) a precomputed vanishing angle $\theta_i^v \in [-45^\circ, 45^\circ)$ for camera i Zhang et al. [2014]. Incorporating vanishing-angle embeddings exploits the high precision of indoor vanishing angles Li et al. [2025], Lambert et al. [2022] to axis-align the predicted floor layout. The global floor coordinate frame is centered at the first camera and oriented by its vanishing angle.

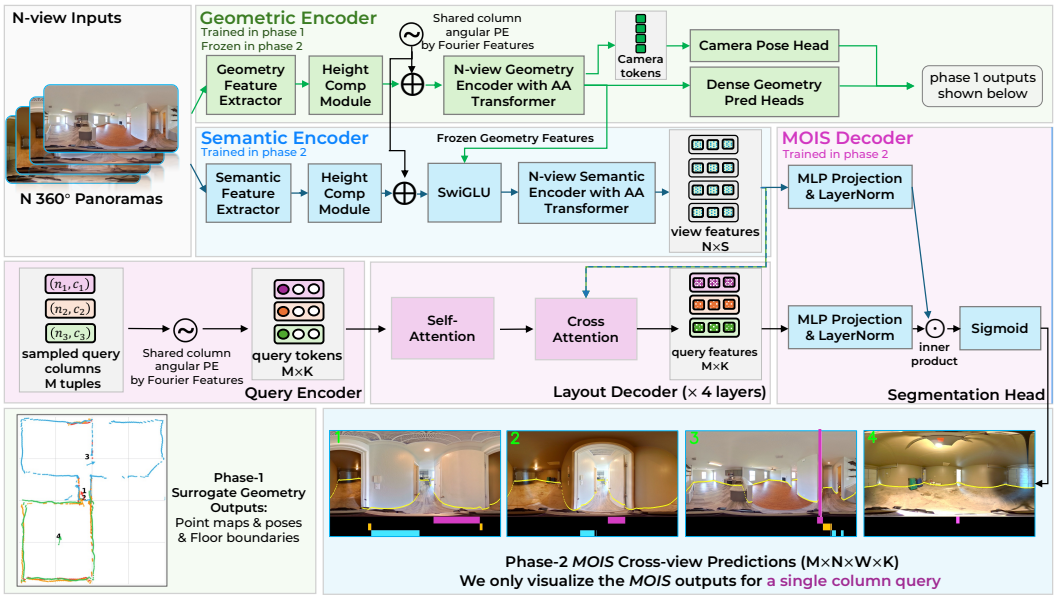


Figure 2: Architecture of *MvFFN* with two-stage training for *MOIS* prediction.

Alternating-attention transformer. We concatenate geometry tokens from all views, $t^G = \bigcup_{i=1}^{N} f_i^G$, with camera tokens $t_i^C \in \mathbb{R}^{1\times C}$ and register tokens $t_i^R \in \mathbb{R}^{4\times C'}$ Darcet et al. [2023], and process them with a 12-layer alternating-attention transformer Wang et al. [2025]. The transformer alternates between frame-wise and global attention layers implemented with *FlashAttention* Dao et al. [2022], and produces updated geometry, camera, and register tokens $(\hat{t}_i^G, \hat{t}_i^C, \hat{t}_i^R)_{i=1}^N$.

Dense predictions. We use separate multi-layer perceptron (MLP) heads to predict dense floor boundaries $\hat{\mathcal{B}}$, 2D floor point maps $\hat{\mathcal{P}}$, and 2D wall-line parameters $\hat{\Lambda}$. The point-map output contains both 2D floor coordinates and their 2D aleatoric uncertainties. Each supporting line is represented by a 2D orientation vector and a bias term. Floor boundaries and point maps are predicted with aleatoric uncertainty Kendall and Cipolla [2016]. Following prior panorama layout work Sun et al. [2019], Hutchcroft et al. [2022], Li et al. [2025], floor boundaries and point maps are supervised using derived column-wise targets that pass through open doorways and ignore occlusions from furniture and other non-structural objects, subject to dataset-specific annotation conventions. Camera poses are predicted from per-camera register and camera tokens Wang et al. [2025] using an MLP head with adaptive LayerNorm Perez et al. [2018].

4.2 Floor Layout Semantic Branch

Our floor-layout semantic branch predicts the dense *MOIS* probability field introduced in Sec. 4. Specifically, for each anchor column, it estimates which columns across all views belong to the current wall and to its next $K-1$ clockwise physically adjacent within-room neighbors. We use $K=3$, which provides enough neighboring-wall context for stable connectivity recovery in *Layout Chaining* (Sec. 4.5), while avoiding the degradation of longer prediction windows (Sec. 6). The branch consists of a semantic N-view encoder conditioned on geometry features, a cross-attention transformer decoder with anchor-conditioned wall-slot queries, and a segmentation head.

Semantic N-view encoder. The semantic encoder follows the geometry branch architecture from Sec. 4.1. Each panorama is processed by an *EfficientNet-v2* backbone and HCM to produce column-patch features $f_i^S \in \mathbb{R}^{S\times C}$. These features are fused with the frozen geometry tokens $\hat{t}_i^G$ through a SwiGLU module, and the fused features are then processed by a four-layer semantic alternating-attention transformer, producing semantic column tokens $\hat{t}_i^S$.

Anchor-conditioned wall-slot queries. We adopt a two-level query design with $M \times K$ learned query tokens, where M is the number of sampled anchor columns and K is the number of ordered *MOIS* wall slots. Each query group corresponds to one anchor column and contains K ordered

query tokens associated with the wall slots $r \in \{0, \ldots, K-1\}$. Given sampled anchor columns $\{(i_m, c_m)\}_{m=1}^{M}$, we initialize query-group features using the angular encoding of the anchor column together with a camera identity encoding.

Transformer decoder. The decoder has four stacked layers. Within each anchor group m, self-attention operates over the K ordered wall-slot queries, and cross-attention is applied to the semantic column tokens from all views. The decoder outputs updated query features $\hat{t}^q_{m,r}$, one for each anchor m and wall slot r.

***MOIS* probability prediction.** Let $\tilde{t}^q_{m,r} \in \mathbb{R}^{C'}$ and $\tilde{t}^I_{j,d} \in \mathbb{R}^{C'}$ denote the projected query and semantic features for anchor-slot (m, r) and column (j, d), respectively. We compute the dense *MOIS* probability field as

$$\hat{\mathcal{S}}_{m,j,d,r} = \sigma\big(\langle \tilde{t}^q_{m,r}, \tilde{t}^I_{j,d} \rangle\big)\,, \tag{1}$$

where $\hat{\mathcal{S}}_{m,j,d,r}$ is the probability that column (j, d) belongs to the r-th ordered wall slot of anchor m. During training, M anchor columns are randomly sampled subject to GPU memory limits. During inference, we set $M = NW$ and predict the dense *MOIS* field for all anchor columns in all views in a single forward pass; equivalently a 5D tensor indexed by view and anchor column, as in Algorithm 1.

4.3 Training

We train *MvFFN* in two stages because the geometry and semantic objectives compete under joint training. In the first stage, we train the geometry branch for dense geometric prediction, including floor boundaries, floor point maps, and camera poses, using a multi-task objective that combines geometric supervision with absolute and relative pose losses. In the second stage, we freeze the pre-trained geometry branch and train the floor-layout semantic branch from scratch under dense *MOIS* supervision, reusing the geometry features from stage one. The *MOIS* objective is factorized into a wall-presence term, which predicts whether a column belongs to any ordered wall slot of the anchor, and an ordered wall-slot classification term applied to positive assignments. In both stages, training samples are formed by drawing sparse multi-view panorama groups from a single pose graph with sufficient co-visibility. Full loss definitions, optimizer settings, and sampling schedules are provided in Appendix A.

4.4 Inductive Bias for Training Data Scarcity

Training sparse multi-view models is substantially harder than training 2-view models because they require more diverse multi-view samples, while the number of scenes and unique structured training examples remains limited. In our settings, this makes naive 3-view to 5-view training unstable even when the same data is sufficient for strong 2-view geometric learning. This trend is also consistent with recent geometric models such as *VGGT* Wang et al. [2025] and *MV-DUSt3R* Tang et al. [2025], which require substantially more training data than *DUSt3R* Wang et al. [2024] to improve 2-view pose accuracy. To improve convergence under this regime, we introduce two inductive biases based on reliable 2-view *CovisPose* Hutchcroft et al. [2022] predictions.

Camera rotation priors. Two-view geometric models already provide reliable camera rotations, and per-panorama vanishing points further improve indoor rotation accuracy Zhang et al. [2014], Lambert et al. [2022]. We first run the 2-view *CovisPose* model and recover global camera rotations with a minimum spanning tree, following *BADGR*'s initialization strategy Li et al. [2025]. These refined rotations are then injected as conditioning signals into the angular encoder (Sec. 4.1). This reduces the amount of scarce multi-view supervision spent on relearning camera rotation and lets the model focus on the harder cross-view geometric reasoning problem, while retaining near 2-view rotation accuracy without requiring substantially more N-view training data.

Column-patch correspondence priors. Under limited multi-view training data, we use precomputed 2-view *CovisPose* correspondences during both training and inference to restrict early cross-view attention to a local window around predicted matches, rather than relearning global correspondence search from scratch. We implement this by masking the first four global self-attention layers of the alternating-attention transformer: for each column-patch token in view I_a, tokens in view I_b more than 40 columns away from the predicted match are masked out, and fully non-covisible view pairs

are blocked entirely. This reduces correspondence ambiguity, leading to less noisy gradients and faster convergence.

4.5 Layout Chaining

Layout Chaining (*LC*) is a deterministic procedure that reduces dense local *MOIS* predictions into global column-to-wall assignments $\mathcal{M}$, room-level wall sequences $\{\mathcal{L}^m\}$, and room endpoints $\{V^m\}$. Since many anchor columns provide overlapping same-wall and neighboring-wall evidence, *LC* first consolidates wall identity and then recovers within-room adjacency. It clusters same-view current-wall predictions $\hat{\mathcal{S}}[n, :, n, :, 0]$ into per-view wall instances L_n, then merges these across views into global wall groups through a cross-view same-wall graph. Directed within-room adjacency is estimated from the neighboring *MOIS* slots, producing a room graph whose edges are filtered by geometric consistency and whose longest directed paths define partial room layouts $\mathcal{L}^m$. Finally, wall-line parameters are fit from supporting point-map predictions grouped by $\mathcal{M}$, and room endpoints are recovered by intersecting consecutive walls. Algorithm details are provided in Appendix B.

4.6 Layout Bundle Adjustment

Planar bundle adjustment (*BA*) and *BADGR* Li et al. [2025] are not contributions of this work; we evaluate them to show that *MvFFN*'s automatic correspondences make fully automatic downstream refinement from unposed panoramas possible for the first time. The layouts produced by *Layout Chaining* and the predicted camera poses may still violate multi-view consistency and structural layout constraints. The predicted pixel-to-wall correspondences provide the associations needed for refinement. We investigate whether these methods can use the correspondences to improve poses and wall geometry. Sec. 5.3 reports the results.

5 Experiments

We evaluate primarily on `ZInD-XV`, which extends `ZInD` Cruz et al. [2021] with globally indexed multi-room layouts represented as room-wise wall sequences, and per-panorama column-wise floor-boundary and wall-instance labels derived from those layouts. `ZInD-XV` contains 2,706 floors and 66,834 panoramas. We additionally use `HM3D-XV`, a smaller auxiliary benchmark derived from residential `HM3DSem` scenes Yadav et al. [2023], with 372 floors and 17,943 panoramas, to evaluate cross-dataset transfer to furnished, mesh-rendered indoor scenes. Dataset construction and annotation details are provided in the supplementary material.

We compare against modular baselines assembled from existing pose, geometry, and layout components. The main evaluation measures final structured floor-layout accuracy after *Layout Chaining*; we also report pose estimation and downstream refinement with planar bundle adjustment and *BADGR* Li et al. [2025]. To isolate the learned representation before post-processing, we additionally evaluate raw *MOIS* prediction quality in the supplementary material. Baseline implementation details, `HM3D-XV` transfer results, and additional ablations are also provided in the supplementary material.

5.1 Pose Estimation Evaluation on ZInD-XV

We compare *MvFFN* with *CovisPose+Greedy Spanning Tree* (*GST*) Nejatishahidin et al. [2023], Li et al. [2025] and *VGGT* Wang et al. [2025] on `ZInD-XV`. We adapt the pretrained *VGGT* without fine-tuning: each panorama is decomposed into 12 overlapping perspective crops (six yaw directions at two pitches), all crops from all panoramas are processed jointly, and their poses are robustly aggregated into one planar pose per panorama. Full projection, metric-scale recovery, and aggregation details are provided in Appendix G.1. We exclude *GraphCovis* Nejatishahidin et al. [2023] and *PanoPose* Tu et al. [2024] because public code is unavailable. For each view-count setting, we evaluate on 1,000 N-view test tuples using the pose-graph alignment protocol from prior work Nejatishahidin et al. [2023]; tuple construction details are provided in the supplementary material. Although pose estimation is not the primary target of *MvFFN*, Table 1 shows that it is comparable to *CovisPose+GST*: *CovisPose+GST* attains lower median errors, while *MvFFN* reduces mean error, variance, and hard-case failures. *VGGT* performs substantially worse in this sparse wide-baseline panorama setting. We did not train the HCM and *MOIS* heads on a frozen *VGGT* backbone; whether improved panorama adaptation can make this head-only configuration effective remains an open question.

Table 1: Camera pose evaluation on ZInD-XV dataset.

Methods	5v Pos (cm)↓			5v Rot (°)↓			8v Pos (cm)↓			8v Rot (°)↓		
	Mn	Std	Med	Mn	Std	Med	Mn	Std	Med	Mn	Std	Med
CovisPose+GST	12.6	23.3	**9.3**	1.3	0.9	**0.7**	17.3	37.3	**11.8**	1.9	1.6	**0.8**
VGGT	30.7	39.4	13.3	11.1	16.0	4.0	53.4	63.1	27.8	13.3	17.7	4.9
MvFFN	**12.5**	**10.8**	10.2	**1.2**	**0.6**	1.1	**17.0**	**15.6**	13.8	**1.3**	**0.8**	1.2

5.2 Floor Layout Estimation Evaluation on ZInD-XV

We compare *MvFFN* with modular floor-layout pipelines because no prior method directly predicts structured multi-room layouts from sparse, unposed N-view panoramas. As summarized in Table 2, each baseline composes pose estimation, geometric input construction, and layout vectorization: *CovisPose+GST* (*CP*) or ground-truth poses provide the pose input, while the final layout is produced by heuristic *LGT* post-processing (*LGT-PP*) or learning-based top-down vectorization with *RoomFormer* (*RF*) Yue et al. [2023] and *PolyRoom* (*PR*) Liu et al. [2024]. We additionally evaluate *DOP-Net* Shen et al. [2023] and *Bi-Layout* Tsai et al. [2024] as fixed-protocol *LGT-Net* front-end replacements, and a best-effort *NadirFloorNet* Pintore et al. [2025] reproduction with *CP* registrations and no view-count fine-tuning because no public ZInD checkpoint or training entry point is available. We exclude *VGGT* from layout evaluation because its pose accuracy is substantially worse than *CovisPose*; implementation and metric details are provided in the supplementary material.

Table 2: Floor-layout evaluation on ZInD-XV. Distance is in cm. Geom. denotes the intermediate geometric input and Layout denotes the vectorization backend. GT-pose replaces the modular pipeline's estimated poses with ground-truth poses, isolating pose error. GT masks replace MvFFN's CovisPose-induced attention masks with ground-truth masks, testing sensitivity to that geometric prior. Bold marks the best result without ground-truth inputs; – denotes an unreported junction distance.

Method / Pipeline			5-view wall				5-view junction				8-view wall				8-view junction			
pose	geom.	layouts	Prec.↑	Rec.↑	F1 ↑	Dist ↓	Prec.↑	Rec.↑	F1 ↑	Dist ↓	Prec.↑	Rec.↑	F1 ↑	Dist ↓	Prec.↑	Rec.↑	F1 ↑	Dist ↓
MvFFN			**88.5**	**87.7**	**88.1**	**13.9**	**79.2**	**68.2**	**73.3**	**13.8**	**89.4**	**87.5**	**88.4**	**19.0**	**81.0**	**69.8**	**75.0**	**19.1**
CP		*RF*	31.9	44.2	37.1	84.9	15.4	20.9	17.7	137.8	35.0	42.4	38.3	79.0	15.7	18.6	17.0	136.8
CP		*PR*	33.6	29.3	31.3	86.6	17.4	14.9	16.1	129.7	31.8	25.0	28.0	86.2	13.8	10.5	11.9	133.4
CP	*NadirFloorNet*		64.7	61.6	63.1	30.8	46.3	43.3	44.8	–	64.0	65.8	64.9	33.8	47.9	47.7	47.8	–
CP	*LGT*	*LGT-PP*	75.6	71.4	73.4	21.3	60.0	54.8	57.3	21.3	76.3	72.7	74.5	24.4	61.7	56.7	59.1	24.4
CP	*DOP-Net*	*LGT-PP*	80.0	69.0	74.1	19.7	62.9	52.9	57.5	–	77.9	72.3	75.0	22.8	63.2	56.7	59.8	–
CP	*Bi-Layout*	*LGT-PP*	79.6	69.4	74.2	19.6	62.7	53.2	57.6	–	78.4	72.6	75.4	22.5	63.7	57.0	60.2	–
GT-pose	*LGT*	*LGT-PP*	78.8	70.5	74.4	19.2	62.4	54.3	58.1	19.7	80.5	70.8	75.3	22.7	65.1	55.5	59.9	22.7
MvFFN (GT masks)			88.9	87.9	88.4	14.0	79.4	67.9	73.2	13.9	87.8	87.1	87.4	19.2	79.2	69.2	73.9	19.1

We evaluate wall and junction precision, recall, and $F1$ with instance-level matching, and report matched-wall localization error; details are in the supplementary material. These structure-aware metrics are better suited to our setting than region IoU, since high-quality floor plan reconstruction requires accurate wall instances, junctions, connectivity, and geometric alignment, and *MvFFN* may produce either closed polygons or open wall sequences under limited view coverage Cruz et al. [2021]. As shown in Table 2, *MvFFN* substantially outperforms all modular baselines across both 5-view and 8-view settings. The strongest modern front-end substitution, *Bi-Layout*, reaches 74.2/57.6 wall/junction $F1$ at 5 views and 75.4/60.2 at 8 views, compared with 88.1/73.3 and 88.4/75.0 for *MvFFN*. *MvFFN* also outperforms *LGT-PP* with ground-truth poses, while replacing its CovisPose-induced attention masks with ground-truth masks changes accuracy by at most one point. Together, these controls show that stronger single-view boundaries, poses, or attention masks do not explain the margin; the remaining advantage comes from cross-view wall identity and ordered connectivity predicted by *MOIS* and assembled by *Layout Chaining*.

5.3 Global Refinement

We evaluate whether *MvFFN*'s predicted wall instances and pixel-to-wall correspondences support automatic downstream refinement. Table 3 reports matched-wall distance before and after planar bundle adjustment (*BA*) and *BADGR* Li et al. [2025], both over all samples and over the subset improved by each method. Planar bundle adjustment improves a majority of samples in both 5-view and 8-view settings, with clear gains on the improved subset. Overall gains are modest, and *BADGR* is more sensitive to initialization because it assumes complete initial room layouts and clean correspondences. Details are provided in the supplementary material.

Table 3: Matched-wall distance before and after refinement on ZInD-XV. Distances are in cm.

View	Method	All samples				Improved samples				
		Mean dist		Med. dist		Improved	Mean dist		Med. dist	
		before	after	before	after	(%)	before	after	before	after
5-view	*MvFFN+BA*	13.9	**13.7**	6.9	**6.7**	**68.0%**	14.8	**12.3**	7.5	**5.9**
	MvFFN+BADGR	13.9	17.7	6.9	8.1	**45.9%**	16.2	**14.1**	8.2	**6.2**
8-view	*MvFFN+BA*	19.0	**18.8**	8.8	8.8	**66.0%**	20.6	**17.8**	9.7	**8.2**
	MvFFN+BADGR	19.0	21.0	10.1	10.4	**48.5%**	22.9	**20.0**	8.8	**8.8**

6 Ablation Studies

We evaluate three complementary ablations. This section studies **architecture**, testing whether the two-branch, two-stage *MvFFN* design is necessary for stable *MOIS* prediction, and **prediction window**, testing how longer ordered wall sequences affect accuracy. Appendix E.1 studies **inductive bias**, evaluating rotation and correspondence priors across input-view counts.

Architecture. We ablate two core design choices in *MvFFN*: a dedicated semantic branch for *MOIS* prediction and the two-stage geometry-to-semantics training strategy. Table 4 compares four variants: a shared-encoder model, a two-branch model without geometry-to-semantic fusion, a full architecture trained in one stage, and the full two-stage *MvFFN*. The results show that both geometry-conditioned semantic reasoning and staged training are critical: removing geometry-to-semantic fusion sharply reduces recall and $F1$, while shared features or single-stage training underperform and become unstable, indicating tension between dense geometry objectives and the semantic *MOIS* objective. The full two-stage design gives the best stability and final layout accuracy.

Table 4: Ablation study of *MvFFN* on ZInD-XV, comparing encoder design, geometry fusion, and staged training. – denotes pose metrics identical to *MvFFN*.

Method	Geom Enc	Sem Enc	Geom Fusion	Train Stages	5v pos (cm)↓			5v rot (°)↓			5v wall				5v junction			
					Mn	Std	Med	Mn	Std	Med	Prec↑	Rec↑	$F1$ ↑	Dist↓	Prec↑	Rec↑	$F1$ ↑	Dist↓
Shared-Enc	✓	✗	✗	2	35.8	50.1	26.4	2.1	1.3	2.0	86.4	69.8	77.2	31.2	72.9	50.4	59.6	39.9
No-Fusion	✓	✓	✗	2	–	–	–	–	–	–	**92.6**	22.7	36.5	28.0	**84.3**	14.4	24.6	21.4
1-stage	✓	✓	✓	1	46.8	50.5	30.1	2.5	1.5	1.9	87.7	71.9	79.0	33.9	75.8	53.1	62.5	44.1
MvFFN	✓	✓	✓	2	**12.5**	**10.8**	**10.2**	**1.2**	**0.6**	**1.1**	88.5	**87.7**	**88.1**	**13.9**	79.2	**68.2**	**73.3**	**13.8**

***MOIS* prediction window.** To select the prediction-window size, we train two additional *MvFFN* variants with 5 and 7 ordered wall slots (Table 5). *Layout Chaining* remains unchanged and uses only the first three slots, isolating the effect of the training target. Longer windows consistently reduce accuracy: 5-view wall $F1$ drops from 88.1 to 85.6 and 84.4, while 8-view junction $F1$ drops from 75.0 to 71.3 and 67.1; 8-view matched-wall distance also worsens. Thus, the 3-wall window is the best operating point: compact local prediction is easier under sparse views, whereas longer sequences accumulate ambiguity from occlusion, room boundaries, and distant structure.

Table 5: *MvFFN* layout accuracy reported on 5- and 8-view test set from ZInD-XV. The model is trained to predict different numbers of adjacent walls in *MOIS* prediction.

# walls	5-view wall				5-view junction				8-view wall				8-view junction			
	Prec↑	Rec↑	$F1$ ↑	Dist↓	Prec↑	Rec↑	$F1$ ↑	Dist↓	Prec↑	Rec↑	$F1$ ↑	Dist↓	Prec↑	Rec↑	$F1$ ↑	Dist↓
3-wall	88.5	**87.7**	**88.1**	**13.9**	**79.2**	**68.2**	**73.3**	**13.8**	**89.4**	**87.5**	**88.4**	**19.0**	**81.0**	**69.8**	**75.0**	**19.1**
5-wall	**88.7**	82.8	85.6	14.7	78.8	64.2	70.8	17.0	88.4	83.7	86.0	19.6	79.4	64.7	71.3	23.2
7-wall	87.6	81.4	84.4	16.2	76.3	58.7	66.4	18.7	87.7	78.9	83.1	19.2	77.9	59.0	67.1	21.7

7 Qualitative Results

Qualitative results in Fig. 3 show representative 8-view reconstructions, with additional examples provided in Appendix J. Across simple and complex multi-room scenes, *MvFFN* produces more complete and globally consistent layouts than the strongest modular baseline, *LGT-Net* with *CovisPose*. The predicted wall instances are color-coded to show cross-view consistency. Compared with the baseline, *MvFFN* better preserves room shape, avoids duplicate or fragmented walls, and recovers neighboring structure visible through doorways under partial occlusion.



Figure 3: Qualitative comparisons on ZInD-XV test set. We show the strongest modular baseline, *LGT-Net* with *CovisPose*.

8 Conclusion

We introduced *MvFFN*, which reconstructs structured floor plans and camera poses from sparse, unposed 360° panoramas by predicting cross-view wall instances and within-room connectivity before global assembly. Experiments show clear gains over modular baselines, demonstrate automatic downstream refinement from predicted wall correspondences, and introduce ZInD-XV for future structured multi-view layout research.

References

Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, et al. Scenescript: Reconstructing scenes with an autoregressive structured language model. In *European Conference on Computer Vision*, pages 247–263. Springer, 2024.

Robert Bridson. Fast Poisson disk sampling in arbitrary dimensions. In *ACM SIGGRAPH 2007 Sketches*, pages 22–es. ACM, 2007.

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*, pages 2661–2670, 2019.

Jiacheng Chen, Ruizhi Deng, and Yasutaka Furukawa. Polydiffuse: Polygonal shape reconstruction via guided set diffusion models. *arXiv preprint arXiv:2306.01461*, 2023.

Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in neural information processing systems*, 34:17864–17875, 2021.

Steve Cruz, Will Hutchcroft, Yuguang Li, Naji Khosravan, Ivaylo Boyadzhiev, and Sing Bing Kang. Zillow indoor dataset: Annotated floor plans with 360deg panoramas and 3d room layouts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2133–2143, 2021.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35: 16344–16359, 2022.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.

Gustav Hanning, Kalle Åström, and Viktor Larsson. Pixcuboid: Room layout estimation from multi-view featuremetric alignment. In *2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 7283–7292. IEEE, 2025.

Yaxuan Huang, Xili Dai, Jianan Wang, Xianbiao Qi, Yixing Yuan, and Xiangyu Yue. Unposed sparse views room layout reconstruction in the age of pretrain model. In *International Conference on Learning Representations*, volume 2025, pages 43478–43495, 2025.

Will Hutchcroft, Yuguang Li, Ivaylo Boyadzhiev, Zhiqiang Wan, Haiyan Wang, and Sing Bing Kang. Covispose: Co-visibility pose transformer for wide-baseline relative pose estimation in 360-degree indoor panoramas. In *European Conference on Computer Vision*, pages 615–633. Springer, 2022.

Zhigang Jiang, Zhongzheng Xiang, Jinhua Xu, and Ming Zhao. Lgt-net: Indoor panoramic room layout estimation with geometry-aware transformer network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1654–1663, 2022.

Dongki Jung, Jaehoon Choi, Yonghan Lee, and Dinesh Manocha. Im360: Large-scale indoor mapping with 360 cameras. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 29040–29050. IEEE, 2025.

Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *2016 IEEE international conference on Robotics and Automation (ICRA)*, pages 4762–4769. IEEE, 2016.

John Lambert, Yuguang Li, Ivaylo Boyadzhiev, Lambert Wixson, Manjunath Narayana, Will Hutchcroft, James Hays, Frank Dellaert, and Sing Bing Kang. Salve: Semantic alignment verification for floorplan reconstruction from sparse panoramas. In *European Conference on Computer Vision*, pages 647–664. Springer, 2022.

Yuguang Li, Ivaylo Boyadzhiev, Zixuan Liu, Linda Shapiro, and Alex Colburn. Badgr: Bundle adjustment diffusion conditioned by gradients for wide-baseline floor plan reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16785–16795, 2025.

Yuzhou Liu, Lingjie Zhu, Xiaodong Ma, Hanqiao Ye, Xiang Gao, Xianwei Zheng, and Shuhan Shen. Polyroom: Room-aware transformer for floorplan reconstruction. *arXiv preprint arXiv:2407.10439*, 2024.

Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. Openmvg: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016.

Negar Nejatishahidin, Will Hutchcroft, Manjunath Narayana, Ivaylo Boyadzhiev, Yuguang Li, Naji Khosravan, Jana Košecká, and Sing Bing Kang. Graph-covis: Gnn-based multi-view panorama global pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6459–6468, 2023.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Giovanni Pintore, Marco Agus, and Enrico Gobbetti. Atlantanet: inferring the 3d indoor layout from a single 360-degree image beyond the manhattan world assumption. In *European Conference on Computer Vision*, pages 432–448. Springer, 2020.

Giovanni Pintore, Uzair Shah, Marco Agus, and Enrico Gobbetti. Nadirfloornet: reconstructing multi-room floorplans from a small set of registered panoramic images. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1976–1985. IEEE, 2025.

Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

Zhijie Shen, Zishuo Zheng, Chunyu Lin, Lang Nie, Kang Liao, Shuai Zheng, and Yao Zhao. Disentangling orthogonal planes for indoor panoramic room layout estimation with cross-scale distortion awareness. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17337–17345. IEEE, 2023.

Jheng-Wei Su, Kuei-Yu Tung, Chi-Han Peng, Peter Wonka, and Hung-Kuo James Chu. Slibo-net: Floorplan reconstruction via slicing box representation with local geometry regularization. *Advances in Neural Information Processing Systems*, 36:48781–48792, 2023.

Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1047–1056, 2019.

Cheng Sun, Chi-Wei Hsiao, Ning-Hsu Wang, Min Sun, and Hwann-Tzong Chen. Indoor panorama planar 3d reconstruction via divide and conquer. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11333–11342. IEEE, 2021a.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. Hohonet: 360 indoor holistic understanding with latent horizontal features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2573–2582, 2021b.

Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, pages 10096–10106. PMLR, 2021.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5283–5293, 2025.

Yu-Ju Tsai, Jin-Cheng Jhang, Jingjing Zheng, Wei Wang, Albert YC Chen, Min Sun, Cheng-Hao Kuo, and Ming-Hsuan Yang. No more ambiguity in 360° room layout via bi-layout estimation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28056–28065. IEEE, 2024.

Diantao Tu, Hainan Cui, Xianwei Zheng, and Shuhan Shen. Panopose: Self-supervised relative pose estimation for panoramic images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20009–20018, 2024.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Led2-net: Monocular 360deg layout estimation via differentiable depth rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12956–12965, 2021.

Haiyan Wang, Will Hutchcroft, Yuguang Li, Zhiqiang Wan, Ivaylo Boyadzhiev, Yingli Tian, and Sing Bing Kang. Psmnet: Position-aware stereo merging network for room layout estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8616–8625, 2022.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, et al. Habitat-matterport 3d semantics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4927–4936, 2023.

Shang-Ta Yang, Fu-En Wang, Chi-Han Peng, Peter Wonka, Min Sun, and Hung-Kuo Chu. Dula-net: A dual-projection network for estimating room layouts from a single rgb panorama. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3363–3372, 2019.

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*, pages 845–854, 2023.

Yinda Zhang, Shuran Song, Ping Tan, and Jianxiong Xiao. Panocontext: A whole-room 3d context model for panoramic scene understanding. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 668–686. Springer, 2014.

Lipu Zhou. Efficient second-order plane adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13113–13121, 2023.

Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2051–2059, 2018.

Appendix

A Training Details

We train *MvFFN* in two stages.

N-view geometry pre-training. We first train the geometry branch with the multi-task loss

$$\mathcal{L}^G = \mathcal{L}_{\mathcal{P}} + \mathcal{L}_{\mathcal{B}} + \mathcal{L}_{rel} + \mathcal{L}_{abs}. \tag{2}$$

Here, $\mathcal{L}_{\mathcal{P}}$ and $\mathcal{L}_{\mathcal{B}}$ denote the point-map and floor boundary losses. Following *DUSt3R* Wang et al. [2024], both use aleatoric uncertainty weighting Kendall and Cipolla [2016] with predicted uncertainties $\Sigma_i^{\mathcal{P}}$ and $\Sigma_i^{\mathcal{B}}$. The pose losses $\mathcal{L}_{rel}$ and $\mathcal{L}_{abs}$ follow *CovisPose* Hutchcroft et al. [2022]; valid pairs for $\mathcal{L}_{rel}$ have minimum co-visibility 0.01. Specifically,

$$\mathcal{L}_{\mathcal{P}} = \sum_{i=1}^{N} \left\| \Sigma_i^{\mathcal{P}} (\mathcal{P}_i - \hat{\mathcal{P}}_i) \right\| - \alpha \log \Sigma_i^{\mathcal{P}},$$

$$\mathcal{L}_{\mathcal{B}} = \sum_{i=1}^{N} \left\| \Sigma_i^{\mathcal{B}} (\mathcal{B}_i - \hat{\mathcal{B}}_i) \right\| - \alpha \log \Sigma_i^{\mathcal{B}},$$

$$\mathcal{L}_{abs} = \sum_{i=1}^{N} \left(\|t_i - \hat{t}_i\| + \beta_r \|r_i - \hat{r}_i\| \right).$$

and $\mathcal{L}_{rel}$ is defined analogously on relative camera translations and rotations for valid image pairs. Here, r_i, t_i and $\hat{r}_i$, $\hat{t}_i$ denote the ground truth and predicted camera translation and normalized rotation vector for camera i, and $\beta_r = 0.1$.

We optimize Eq. (2) with AdamW for 80k iterations, a cosine learning-rate schedule with peak 10^{-4}, and 4k warmup iterations. Each batch samples one scene and 2, 3, 5, or 8 images per example such that the views form a single pose graph with co-visibility greater than 20%. A linear view-count scheduler shifts from 2 and 3-view batches early in training to 5 and 8-view batches later. We use random horizontal rotation augmentation and clip gradients to norm 1.0.

Layout semantic training. We train the floor-layout semantic branch from scratch while freezing the geometry branch and reusing its features. We use the same data-sampling schedule and a similar learning-rate schedule, with maximum learning rate 3.5×10^{-4}. The semantic branch uses dense *MOIS* supervision with a factorized objective: wall-presence prediction for each anchor-column pair, and ordered wall-slot classification on positives.

Let $m \in \{1, \ldots, M\}$ index sampled anchor columns, with anchor m corresponding to (i_m, c_m). For anchor m and column (j, d), the decoder produces dense *MOIS* probabilities $\hat{\mathcal{S}}_{m,j,d,r} = \sigma(z^{\text{cls}}_{m,j,d,r})$, where $r \in \{0, \ldots, K-1\}$ indexes the ordered wall slot relative to the anchor. The binary supervision is $\mathcal{S}^{\text{gt}}_{m,j,d,r} \in \{0, 1\}$.

We aggregate wall-slot logits into a presence logit $z^{\text{pres}}_{m,j,d} = \log \sum_{r=0}^{K-1} \exp(z^{\text{cls}}_{m,j,d,r})$, with target $y^{\text{pres}}_{m,j,d} \in \{0, 1\}$ indicating whether (j, d) is assigned to any ordered wall slot of anchor m. The wall-presence and ordered-slot classification losses are

$$\mathcal{L}_{\text{pres}} = \frac{1}{|\Omega_{\text{valid}}|} \sum_{(m,j,d) \in \Omega_{\text{valid}}} \text{BinaryCrossEntropy}\big(z^{\text{pres}}_{m,j,d},\, y^{\text{pres}}_{m,j,d}\big),$$

$$\mathcal{L}_{\text{cls}} = \frac{1}{|\Omega_{\text{pos}}|} \sum_{(m,j,d) \in \Omega_{\text{pos}}} \text{CrossEntropy}\Big(z^{\text{cls}}_{m,j,d,:},\, y^{\text{cls}}_{m,j,d}\Big),$$

$$\mathcal{L}_{\textit{MOIS}} = \mathcal{L}_{\text{pres}} + \lambda_{\text{cls}} \mathcal{L}_{\text{cls}}.$$

Here, Ω_{valid} is the set of valid projected panorama columns after masking invalid regions; $\Omega_{\text{pos}} \subseteq \Omega_{\text{valid}}$ contains columns assigned to one of the K ordered slots, with label $y^{\text{cls}}_{m,j,d} \in \{0, \ldots, K-1\}$.

B *Layout Chaining* Post Processing

Layout Chaining (*LC*) decodes the dense *MOIS* tensor $\hat{\mathcal{S}}$ into compact structured layout outputs: global column-to-wall assignments $\mathcal{M}$, room-level wall sequences $\{\mathcal{L}^m\}$, and metric wall endpoints $\{V^m\}$. The input is dense because every anchor column predicts cross-view masks for the current wall and neighboring wall slots, creating redundant local evidence for wall identity and adjacency.

LC reduces this evidence in four stages: (1) same-view current-wall masks are grouped into per-view wall instances L_n; (2) cross-view same-wall evidence merges them into global wall instances, defining $\mathcal{M}$; (3) neighboring-wall slots induce directed within-room adjacency and room-level wall sequences $\{\mathcal{L}^m\}$; and (4) predicted point maps and wall-line parameters provide metric support for wall-line fitting and endpoint recovery, yielding $\{V^m\}$.

Algorithm 1: *MvFFN* Post-processing: *Layout Chaining*

Input: Dense *MOIS* field $\hat{\mathcal{S}} \in [0,1]^{N\times W\times N\times W\times K}$, floor point maps $\{\hat{\mathcal{P}}_{i,c}\}$, and wall-line parameters $\{\hat{\lambda}_{i,c}\}$
Output: Column-to-wall assignments $\mathcal{M}$, within-room connectivity $\{\mathcal{L}^m\}$, and room endpoints $\{V^m\}$

`/* Stage 1: Per-view wall instances */`
for *each view* n **do**
  Define per-view current wall segmentation: $\hat{\mathcal{S}}_n^{\text{view}} = \hat{\mathcal{S}}[n,:,n,:,0]$;
  Define per-view column nodes: $\mathcal{V}_n^{\text{view}} = \{1,\dots,W\}$;
  for *column pair* $(c_1,c_2) \in \mathcal{V}_n^{\text{view}} \times \mathcal{V}_n^{\text{view}}$ **do**
    Define same-wall column edge: add (c_1,c_2) to $\mathcal{E}_n^{\text{view}}$ if $\mathrm{IoU}(\hat{\mathcal{S}}_n^{\text{view}}[c_1], \hat{\mathcal{S}}_n^{\text{view}}[c_2]) > \tau_{\text{iou}}$;
  end
  Build per-view same-wall column graph: $\mathcal{G}_n^{\text{view}} = (\mathcal{V}_n^{\text{view}}, \mathcal{E}_n^{\text{view}})$;
  Collect per-view wall instances: $L_n = \textbf{ConnectedComponents}(\mathcal{G}_n^{\text{view}})$;
  for *each connected component* $l \in L_n$ **do**
    Select anchor column: $c^* = \arg\max_{c\in l} \deg(c)$;
    Define instance mask: $l^* = \hat{\mathcal{S}}_n^{\text{view}}[c^*]$;
  end
end
`/* Stage 2: Cross-view global wall instances */`
Define cross-view wall nodes: $\mathcal{V}_{\text{cross}} = \bigcup_{n=1}^{N} L_n$;
for *instance pair* $(l_i,l_j) \in L_{n_1} \times L_{n_2},\ n_1 \neq n_2$ **do**
  Define cross-view same-wall instance edge: add (l_i,l_j) to $\mathcal{E}_{\text{cross}}$ if $\mathrm{IoU}(l_i^*, l_j^*) > \tau_{\text{iou}}$;
end
Build cross-view same-wall instance graph: $\mathcal{G}_{\text{cross}} = (\mathcal{V}_{\text{cross}}, \mathcal{E}_{\text{cross}})$;
Collect global wall instances via cross-view column-to-wall assignments: $\mathcal{M} = \textbf{ConnectedComponents}(\mathcal{G}_{\text{cross}})$;
`/* Stage 3: Room connectivity and partial layouts */`
Define room-wall nodes: $\{w^\ell\}$, induced by $\mathcal{M}$;
Define directed room-wall edges: $\mathcal{E}_{\text{room}} = \{(w_i,w_j) \mid \text{the } k\to 0 \text{ predictions in } \hat{\mathcal{S}}[:,:,:,:,1:K-1] \text{ support } w_i \to w_j\}$;
Build directed room graph: $\mathcal{G}_{\text{room}} = (\{w^\ell\}, \mathcal{E}_{\text{room}})$;
Filter invalid edges by geometry: $\mathcal{G}_{\text{room}} = \textbf{GeomFilter}(\mathcal{G}_{\text{room}})$;
for *each weakly connected component* $\mathcal{H}^m \subseteq \mathcal{G}_{\text{room}}$ **do**
  Collect within-room connectivity: $\mathcal{L}^m = \textbf{LongestPath}(\mathcal{H}^m)$;
end
`/* Stage 4: Final wall geometry */`
Define global wall parameters: $\{(\hat{b}^\ell, \hat{\mathbf{v}}^\ell)\} = \textbf{LineFit}(\{\hat{\mathcal{P}}_{i,c}\}, \mathcal{M})$;
for *each partial room layout* $\mathcal{L}^m$ **do**
  Define room endpoints: $V^m = (v^{m,1},\dots,v^{m,K_m}) = \textbf{LineIntersect}(\mathcal{L}^m, \mathcal{M})$;
end

C Intermediate *MOIS* Prediction Evaluation

***MOIS* evaluation.** We evaluate the intermediate *MOIS* representation before *Layout Chaining* with *Pooled Ordered Wall-Slot IoU*, an IoU metric computed separately for each ordered *MOIS* wall slot. For each N-view test tuple, we sample anchor columns from the conditioning panoramas and retain anchors with valid ordered slot targets. Each anchor instantiates the cross-view slot masks defined in Sec. 4 of the main paper. For each slot, we accumulate intersections and unions between the predicted and ground-truth masks over all valid anchors, views, and panorama columns before taking the ratio. This pixel-weighted pooling measures aggregate cross-view mask quality and avoids the high variance of query-averaged mIoU for small or weakly visible wall instances.

***LGT-Net + CovisPose* baseline.** No prior method directly predicts ordered cross-view wall instances, so we construct a composed baseline from *LGT-Net* Jiang et al. [2022] and *CovisPose* Hutchcroft et al. [2022]. The goal is to approximate *MOIS* by first assigning each anchor column to an ordered set of local wall slots in its source view, then propagating those slots across views. Given the single-view layout predicted by *LGT-Net*, we convert it into a floor-plan polygon whose edges define the local slots: the edge intersected by the anchor ray is slot 1, and the next two adjacent edges in clockwise order are slots 2 and 3. *CovisPose* then provides cross-view column correspondences, which propagate these source-view slots to other panoramas and produce cross-view slot masks. We evaluate these masks against the same ordered wall-instance targets using *Pooled Ordered Wall-Slot IoU*.

Table C.1: *Pooled Ordered Wall-Slot IoU*↑ (%) for raw *MOIS* predictions before *Layout Chaining*. Wall 1 denotes the anchor wall, and 2-3 denote the next ordered neighboring walls.

	5-view inputs			8-view inputs		
Method	wall 1	wall 2	wall 3	wall 1	wall 2	wall 3
MvFFN	**81.9**	**71.1**	**57.8**	**75.4**	**63.8**	**47.7**
LGT + CP	63.7	37.8	20.3	60.4	34.5	18.5

Table C.1 shows that *MvFFN* substantially improves raw *MOIS* prediction quality before any *Layout Chaining* post-processing. Wall 1 mainly reflects cross-view wall-instance recovery, while walls 2 and 3 evaluate the ordered neighboring-wall predictions that encode local within-room connectivity. The larger gains on walls 2 and 3 show that *MvFFN* learns not only same-wall correspondence but also adjacent-wall structure more reliably than the composed *LGT-Net+CovisPose* baseline. These stronger local predictions provide better inputs to *Layout Chaining*, supporting the downstream gains in final wall and junction accuracy.

D Experiments on HM3D-XV Dataset

We use `HM3D-XV` as an auxiliary cross-domain validation benchmark to evaluate whether *MvFFN* and the *MOIS* formulation remain effective beyond `ZInD-XV`. `HM3D-XV` is derived from residential `HM3DSem` scenes Yadav et al. [2023] with sufficient mesh coverage, and contains 17,943 panoramas and 372 floors from 180 scenes. Compared with `ZInD-XV`, which contains 2,706 floors, `HM3D-XV` is smaller and noisier, contains furnished mesh-rendered scenes with rendering artifacts, and has sparser column-level supervision, with 27.3% of image columns masked after annotation cleanup. Dataset construction details are provided in Sec. F.2. This makes `HM3D-XV` a challenging setting for both pose estimation and layout reconstruction. We therefore use `ZInD-XV` as an anchor dataset and finetune *MvFFN* with mixed `ZInD-XV`/`HM3D-XV` samples to test whether the learned geometry and *MOIS* representations remain useful under domain shift and noisy annotations.

For evaluation, we construct N-view test splits from per-floor tuples generated by covisibility-based sampling. Each floor receives a sample count proportional to its number of panoramas, capped by the number of valid tuples; remaining samples are redistributed according to residual capacity. This yields 951 tuples for the 5-view split and 991 tuples for the 8-view split, spanning all floors with valid N-view connected components while keeping floor-level representation roughly proportional to scene size. Both pose and layout experiments use these splits. Because some comparisons rely on *CovisPose*-derived priors, we additionally report those results on the *CovisPose*-feasible subset defined in Sec. D.1. Sec. D.1 evaluates pose estimation, and Sec. D.2 evaluates floor layout reconstruction.

D.1 Camera Pose Estimation

We follow the same pose-evaluation protocol as the `ZInD-XV` experiments and report pose accuracy on a *CovisPose*-feasible subset of `HM3D-XV`. Starting from the `ZInD-XV`-trained model, we finetune the geometry branch on `HM3D-XV` using the same losses, view-count schedule, and dataloader setup as Sec. A, with a 60k-iteration learning-rate schedule and the 40k-iteration checkpoint used for evaluation. Training batches mix `HM3D-XV` and `ZInD-XV` examples at a 1:1 ratio to adapt to `HM3D-XV` while retaining broader geometric variation from `ZInD-XV`.

Furnished, mesh-rendered `HM3D-XV` panoramas differ substantially from the `ZInD` panoramas used to train the available *CovisPose* checkpoint. As a result, some pairwise *CovisPose* estimates are

unreliable. We filter pairs with rotation error above 30° or translation error above 2 m, and retain only tuples whose remaining estimates form a connected *GST* pose graph. This removes invalid pairwise links that would otherwise dominate pose and layout metrics, retaining approximately 60% of the 5-view tuples and 65% of the 8-view tuples. We compare *CovisPose+GST*, *VGGT*, and *MvFFN*. For *MvFFN*, we use the same rotation-prior construction as in ZInD-XV, with *CovisPose* rotations refined by vanishing-angle correction. For the correspondence attention mask, we use ground-truth column-patch correspondences to provide a reliable inductive bias for sparse multi-view training on HM3D-XV.

Table D.1: Camera pose evaluation on the *CovisPose*-feasible subset of HM3D-XV. Mn/Std/Med denote mean, standard deviation, and median errors.

Methods	5v Pos (cm)↓			5v Rot (°)↓			8v Pos (cm)↓			8v Rot (°)↓		
	Mn	Std	Med	Mn	Std	Med	Mn	Std	Med	Mn	Std	Med
VGGT	35.2	109.3	**10.4**	6.1	16.3	2.1	62.3	71.8	37.4	13.7	6.4	12.3
CovisPose+GST	30.1	24.7	19.3	2.2	3.3	**1.1**	44.0	59.7	25.9	2.9	4.4	**1.4**
MvFFN	**26.8**	**21.2**	21.5	**1.8**	**0.6**	1.6	**32.0**	**23.5**	**26.6**	**1.9**	**0.5**	1.7

As shown in Table D.1, under reliable correspondence initialization, *MvFFN* produces stable pose estimates on the *CovisPose*-feasible subset of HM3D-XV. Compared with *CovisPose+GST*, it achieves lower mean errors and lower variance across both 5-view and 8-view settings, while *CovisPose+GST* and *VGGT* obtain lower medians in some cases. These results suggest that the *MvFFN* geometry branch, under this setting, can support stable multi-view pose estimation in the furnished HM3D-XV domain.

D.2 Layout Estimation

For layout estimation, our goal is to test whether the learned *MOIS* representation and layout decoder can support multi-room reconstruction in the furnished, mesh-rendered HM3D-XV domain. Following the pose evaluation, we use the same prior setup for *MvFFN* as in Sec. D.1. Starting from the HM3D-XV tuned geometry checkpoint, we retrain only the floor-layout semantic branch with mixed ZInD-XV and HM3D-XV samples, using a 2:1 ZInD-XV: HM3D-XV sampling ratio and otherwise keeping the training settings the same as Sec. A. We compare against *LGT-Net* based modular baselines using either *CovisPose* or ground truth camera poses. As in Sec. 5.2 of the main paper, the modular pipeline separates panorama-level boundary prediction from layout vectorization: *LGT-Net* (*LGT*) provides floor boundary predictions, and *LGT-Net* post-processing (*LGT-PP*) converts them into vectorized layouts. *RoomFormer* and *PolyRoom* are omitted because they are substantially weaker on the main ZInD-XV layout benchmark.

Table D.2: Floor-layout evaluation on the *CovisPose*-feasible subset of HM3D-XV. Distance is in cm. Geom. denotes the intermediate geometric input and Layout denotes the vectorization backend. GT-pose denotes a control group.

pose	pts	layouts	5-view wall				5-view junction				8-view wall				8-view junction			
			prec	recall	f1	dist	prec	recall	f1	dist	prec	recall	f1	dist	prec	recall	f1	dist
CP	*LGT*	*LGT*	57.0	62.6	59.7	51.8	18.2	33.3	23.5	53.1	59.4	60.2	59.8	52.1	19.0	30.5	23.4	53.8
GT	*LGT*	*LGT*	69.1	66.8	67.9	36.9	25.5	44.3	32.4	35.7	69.7	64.6	67.1	39.9	25.5	38.9	30.8	38.6
MvFFN			**77.8**	**69.6**	**73.5**	**22.1**	**55.9**	**38.8**	**45.8**	**24.1**	**75.0**	**71.6**	**73.3**	**28.9**	**52.9**	**35.0**	**42.1**	**29.3**

As shown in Table D.2, under the reliable prior setup, *MvFFN* substantially improves layout accuracy on HM3D-XV despite the smaller and noisier training set. It achieves much higher wall precision and $F1$ than the *LGT-Net* based baselines, and reduces matched wall distance in both 5-view and 8-view settings. The gains are especially large at the junction level, where the modular baselines remain weak even with ground truth poses. This suggests that *MOIS* remains an effective supervision target under sparse column-level annotations, and helps *MvFFN* recover wall connectivity and junction structure more reliably than geometry-first modular pipelines.

E More Ablation Studies

The main paper reports the architecture and *MOIS* prediction-window ablations in Sec. 6. Here, Sec. E.1 studies the rotation and correspondence priors used for sparse multi-view training, and Sec. E.2 provides additional architecture-ablation details.

E.1 Inductive Bias

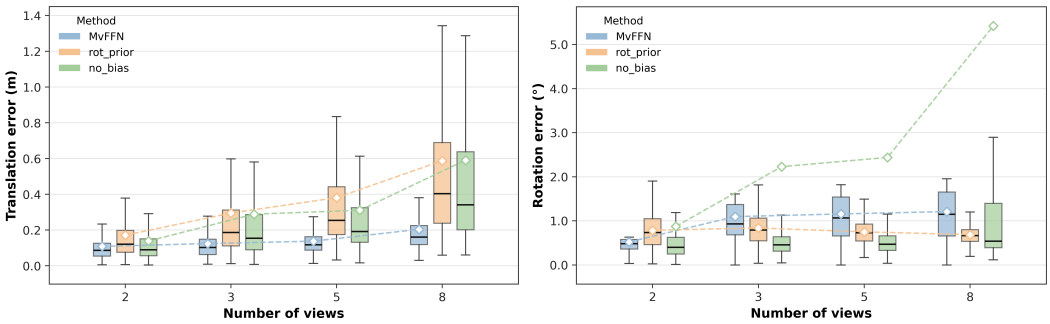


Figure E.1: Effect of inductive bias across input views. Box plots show translation (left) and rotation (right) errors under different inductive-bias variants. Dashed lines and diamond markers indicate mean errors. Stronger inductive bias improves translation accuracy and suppresses extreme rotation outliers.

We evaluate *MvFFN* across varying numbers of input panoramas, using three levels of inductive bias described in Sec. 4.4 of the main paper: no inductive bias, rotation priors only, and rotation priors with correspondence-based attention masks. As shown in Fig. E.1, stronger inductive bias has its clearest effect on translation accuracy: the full *MvFFN* maintains substantially lower translation error across all input-view settings, while the weaker-bias variants degrade much more rapidly as the number of views increases. For rotation, the main benefit is robustness rather than uniformly lower central statistics. In some settings, the no-bias variant can appear slightly better in the median, but this occurs only near the center of the distribution, where all methods already achieve low angular error. Its mean is nonetheless much higher because the error distribution develops a heavy tail: in the 8-view setting, the no-bias variant reaches 17.5° at p90 and 27.8° at p95, whereas the full *MvFFN* remains at 2.5° at p95. Thus, the proposed inductive bias mainly suppresses large rotation failures and produces much more stable estimates in hard cases. Since all variants already use vanishing-angle priors, this gain is not explained by rotation snapping alone, but by the additional regularization introduced by the multi-view inductive bias.

E.2 Architecture Ablation Details

We compare four architecture variants. The shared-encoder model removes the dedicated semantic feature encoder and predicts *MOIS* directly from geometry-branch features. The no-fusion model keeps separate geometry and semantic branches but removes geometry-to-semantic feature fusion, while retaining the same downstream *Layout Chaining* procedure. The one-stage model uses the full two-branch architecture but trains all objectives jointly from scratch. The full model uses separate geometry and semantic branches with the proposed two-stage schedule.

The no-fusion variant substantially degrades layout recall, $F1$, and matched-wall distance, indicating that semantic image features alone are insufficient and that geometry-conditioned semantic features are important for cross-view wall grouping and within-room connectivity. The shared-encoder and one-stage variants both become unstable during training: the shared-encoder model collapses after roughly 25k steps, while the one-stage model collapses after roughly 35k steps. This suggests conflicting optimization pressure between dense geometry prediction, which relies on precise one-to-one cross-view matching, and *MOIS* prediction, which groups columns by wall identity and room structure. Separating the geometry and semantic branches and training them in stages improves both stability and final layout accuracy.

F Dataset Preparation

F.1 ZInD-XV Dataset Preparation

ZInD-XV is derived from all scenes in the Zillow Indoor Dataset (ZInD) Cruz et al. [2021]. It contains 66,834 panoramas, 167,983 unique wall instances, 23,651 rooms, and 2,706 floors from 1,524 residential house scenes in North America. ZInD-XV retains the original panoramas and room layouts in the global frame, but augments them with globally indexed multi-room floor-plan annotations. Specifically, each floor is represented as a set of room layouts with doors and windows, encoded as closed room polygons over globally unique wall instances in the same format as FloorPlan60K used in Li et al. [2025]. Because ZInD often partitions large open spaces into smaller partial rooms, we recursively merge adjacent rooms connected by openings that do not resemble doorways. As a result, the room partition in ZInD-XV is less fragmented than the original ZInD partition in open-plan areas, while still preserving meaningful room-level structure rather than collapsing each floor into a single geometric region. These global wall instances are further projected into each panorama to produce column-wise floor boundaries and wall-instance semantic labels.

Because a single panorama can observe walls from multiple rooms, and visible room connectivity depends on doorway state and view position, these per-panorama labels cannot be obtained by projecting a single room polygon independently. Instead, for each panorama, we search the closest room graph connected through open doors and select the room configuration whose rendered floor boundary best matches the 1D floor boundary predicted by *CovisPose* Hutchcroft et al. [2022]. This graph is then used to generate the per-panorama floor-boundary and wall-instance labels. Finally, we apply a manual cleanup pass to mask panorama columns with inconsistent floor boundaries or incorrect wall-instance assignments. We plan to release the ZInD-XV annotations, including global floor-plans with unique wall instances, per-panorama floor boundaries, wall-instance semantic labels, and validity masks.

F.2 HM3D-XV Dataset Preparation

HM3D-XV is constructed from residential scenes in the Habitat-Matterport 3D Semantic Dataset (HM3DSem) Yadav et al. [2023], restricted to scenes with sufficient mesh coverage to better match the residential scene distribution of ZInD-XV. It contains 17,943 panoramas from 372 eligible floors across 180 scenes, split into 287 training floors and 85 validation floors following HM3DSem. Although HM3DSem provides textured house meshes with semantic labels, it does not provide the panorama-based annotations required by our model: camera-posed panoramas, globally indexed wall instances, room-level wall sequences, and per-panorama column-wise floor-boundary and wall-instance labels. We construct these annotations in three stages: automated floor preparation, room-level wall sequence annotation, and per-panorama column-wise annotation. We plan to release the HM3D-XV annotations, including global floor-plans with unique wall instances, per-panorama camera poses, floor boundaries, wall-instance semantic labels, and validity masks.

Automated floor preparation. This stage converts each multi-floor HM3DSem scene mesh into per-floor sub-scenes with panoramas of known camera poses and initialized global wall instances. We first segment each scene mesh by floor using the semantic labels to identify floor and ceiling geometry. The floor and ceiling heights used for this partition are estimated from the median vertical positions of these surfaces, and nearby split floor regions are merged with simple geometric heuristics. For each floor, we sample panorama centers from the Habitat navigation mesh with a density-controlled greedy Poisson-disk strategy Bridson [2007], targeting approximately 2 panoramas per square meter at a fixed height of 1.2 m. Using the *Habitat-Sim* renderer Ramakrishnan et al. [2021], we render 2048×1024 RGB equirectangular panoramas from the textured mesh with upright camera orientation. Candidate samples that fall on clutter are discarded, and panoramas with severe visible mesh defects are removed during manual inspection. We then initialize global wall instances at the floor level. Vertical wall meshes are grouped by semantic wall instance label from HM3DSem and projected to the floor plane, and a 2D line is fit to each projected wall instance with *RANSAC* to recover the wall parameters defined in Li et al. [2025].

Room-level wall sequence annotation. Starting from the initialized global wall instances, this step produces the room-level wall sequences used as the final multi-room floor-plan layout for each

floor. We designed a custom graphical interface that lets annotators reorder, merge, and split wall instances, and group them into room-wise ordered sequences with globally consistent wall identities. Each room sequence may be closed or open, since incomplete mesh coverage in HM3DSem often prevents some rooms from being represented as closed polygons.

Per-panorama column-wise annotation. Given the annotated multi-room wall layout and panorama camera poses, we project floor geometry and wall instances into each panorama to generate column-wise floor boundaries and wall-instance semantic labels. A separate manual annotation pass is then used to mask out columns with subpar projected assets, for example when floor boundaries are misaligned with visible structure or wall-instance assignments are incorrect.

G Baseline Details

We evaluate modular baselines under a common three-stage pipeline: (1) pose estimation in a shared floor-plane frame, (2) construction of pose-aligned floor boundaries or point-maps, and (3) layout vectorization. This setup lets us compare existing pose, boundary, and layout components under the same sparse multi-view input setting. Unlike *MvFFN*, these baselines do not perform joint multi-view reasoning directly from panorama images in the final layout stage; the layout backend receives only pose-aligned geometric inputs, so upstream pose or boundary errors are already fixed before layout prediction.

G.1 Pose Evaluation

Pose estimation baselines. We compare against two pose baselines: *CovisPose+Greedy Spanning Tree* (*GST*) and *VGGT*. For *CovisPose*, pairwise relative panorama poses are predicted for all panorama pairs, and global 3-DoF poses (2D position and 1D rotation) are recovered by chaining relative transforms along a co-visibility-ranked greedy spanning tree, following prior panoramic pose-estimation work Nejatishahidin et al. [2023], Li et al. [2025]. *VGGT* Wang et al. [2025] does not operate directly on 360° panoramas, so we adapt it through perspective decomposition. We exclude *GraphCovis* Nejatishahidin et al. [2023] and *PanoPose* Tu et al. [2024] because public code is unavailable.

For each view-count setting, we evaluate on 1,000 N-view test tuples from ZInD-XV. Tuples are generated following the prior pose-evaluation protocol Nejatishahidin et al. [2023]: views are sampled from the same floor with minimum pairwise co-visibility of 0.3, and floors are sampled to maintain broad floor coverage. Predicted pose graphs are aligned to ground truth before evaluation following Nejatishahidin et al. [2023], Li et al. [2025]. We report per-panorama position error in centimeters and yaw error in degrees, summarized by mean, standard deviation, and median over all evaluated panoramas.

VGGT panorama adaptation. Since *VGGT* operates on perspective images, each 360° panorama is decomposed into overlapping rectilinear crops with 120° field of view, 60° horizontal overlap, resolution 560×560, and pitch angles 0° and -30°. All crops from all panoramas are processed jointly in a single forward pass. We recover metric scale by aligning the vertical histogram peak of the predicted point cloud to the camera height provided by ZInD Cruz et al. [2021]. We then estimate one planar pose per panorama by compensating each crop pose for its known azimuthal offset and robustly fitting a consensus position and yaw across crops.

G.2 Layout Baseline Pipelines

Layout baseline pipelines. Given Stage-1 poses, all modular layout baselines reconstruct multi-view layouts by first building geometric inputs in the global floor frame and then applying a layout vectorization backend. Unless otherwise noted, we use *CovisPose* poses; GT-pose variants replace them with ground-truth poses to isolate layout error from pose error. *LGT-Net* provides panorama-level floor-wall boundary cues, while *RoomFormer* and *PolyRoom* use fused top-down geometric inputs. We do not use *VGGT* poses for layout baselines because its pose accuracy is substantially worse than *CovisPose* on ZInD-XV.

LGT-Net backend. *LGT-Net* Jiang et al. [2022] is applied independently to each panorama using the pretrained `ZInD` checkpoint without fine-tuning. Its predicted floor-wall boundary is converted into a local floor-plan polygon using Manhattan-world post-processing with *Douglas–Peucker* simplification ($\epsilon = 0.1$), then projected into the global floor plane using the input poses. Polygons whose overlap exceeds 50% of the smaller polygon area are grouped and merged by Shapely union. We also report a GT-pose variant that replaces predicted poses with ground-truth poses.

Topdown training data construction. *RoomFormer* Yue et al. [2023] and *PolyRoom* Liu et al. [2024] operate on topdown geometric inputs rather than panoramas, so we finetune both models on `ZInD-XV` training data constructed to mimic sparse multi-view observations. Starting from lifted floor-plans with fixed ceiling height and room connectivity preserved through open doors, we project per-panorama floor wall boundary points into the global frame using groundtruth poses and rasterize them into 256×256 orthographic floor plane density maps. The corresponding room polygons from `ZInD-XV` are used as supervision. To match test-time conditions, we train with 2, 3, 5, and 8-view samples and add Gaussian noise with $\sigma = 0.001$ on boundary locations, $\sigma = 0.1$ m on panorama positions, and $\sigma = 0.05$ rad on panorama yaw.

RoomFormer. *RoomFormer* Yue et al. [2023] is initialized from its `Structured3D` checkpoint with a *ResNet-50* backbone, 800 vertex queries, and up to 20 room polygons per scene. We fine-tune *RoomFormer* on the rendered `ZInD-XV` density maps and room polygon supervision described above, using AdamW with learning rate 2×10^{-4} (2×10^{-5} for the backbone), weight decay 10^{-4}, batch size 10, and gradient clipping at max norm 0.1. Fine-tuning runs for approximately 70 epochs from the `Structured3D` initialization within a *MultiStepLR* schedule whose learning-rate decay is configured at epoch 400 (not reached). At inference, we render the topdown density map from the evaluated input poses and predicted *LGT-Net* floor wall boundaries, then apply the same vectorization pipeline: corners with foreground probability > 0.5 are retained, polygons with area $< 100\,\mathrm{px}^2$ are discarded, and surviving corners are mapped back to metric floor plane coordinates using a precomputed homography.

PolyRoom. *PolyRoom* Liu et al. [2024] uses the same rendered input representation, but adds a *Mask2Former* Cheng et al. [2021] room-instance segmentation stage whose predicted room contours initialize a *RoomFormer* style polygon decoder instead of fixed learned queries. We train *PolyRoom* in two stages: first fine-tuning a *Mask2Former* with *Swin-L* backbone initialized from `ImageNet-22K` pretraining, then fine-tuning the *PolyRoom* decoder from its `Structured3D` checkpoint on the same rendered `ZInD-XV` data. Stage 1 trains *Mask2Former* at 256×256 input resolution with AdamW (learning rate 10^{-4}, weight decay 0.05, batch size 2) for 400 epochs, with a backbone learning-rate multiplier of 0.1 and standard random-flip, random-resize, and random-crop augmentations. Stage 2 resumes from the `Structured3D`-pretrained *PolyRoom* checkpoint, including its optimizer and scheduler state, and continues fine-tuning on `ZInD-XV` for the remainder of the original *MultiStepLR* schedule (configured for 500 total epochs with a learning-rate decay at epoch 400) using AdamW (learning rate 2×10^{-4}, 2×10^{-5} for the backbone), weight decay 10^{-4}, batch size 10, gradient clipping at max norm 0.1, 800 vertex queries, and up to 20 room polygons per scene. At inference, *Mask2Former* instances with score > 0.6 are converted to contours using *Douglas–Peucker* simplification ($\epsilon = 5\,\mathrm{px}$) and used to seed the decoder. From the decoded polygons, we retain corners with foreground probability > 0.1 and interior angles within 30° of a right angle, merge nearby corners within 10 pixels, discard polygons with area $< 100\,\mathrm{px}^2$, and map the surviving polygons back to metric floor-plane coordinates.

G.3 Layout Evaluation Metrics

We evaluate wall and junction precision, recall, and $F1$ using instance-level matching. A predicted wall is considered a candidate match if its orientation differs by at most 10° and its perpendicular distance is within 2.0 m of a ground-truth wall. Candidate wall pairs are then matched by *Hungarian* assignment based on line overlap. Junction precision, recall, and $F1$ are defined analogously: a junction is counted as matched when its two incident walls are both matched. We also report mean matched-wall localization error. We use these structure-aware metrics because *MvFFN* predicts both closed polygons and open wall sequences, making region-based IoU not well defined for all outputs.

G.4 Downstream Refinement Protocol

We evaluate planar bundle adjustment (*BA*) Zhou [2023] and *BADGR* Li et al. [2025] as downstream refinement methods applied to the *MvFFN* layouts from Sec. 5.3 of the main paper. For each method, we report mean and median matched wall distances before and after refinement on the full evaluation set, and on the subset of samples whose mean matched wall distance improves; the latter also defines the improvement rate. Before refinement, partial room shapes are closed with a simple rule-based procedure to avoid self-intersections. This can introduce a small number of completed walls without image correspondences, which is especially challenging for *BADGR* because it assumes the number of room walls is known. This explains why *BADGR* is more sensitive to initialization in our setting.

H Failure Cases



Figure H.1: Failure case caused by incorrect *MOIS* wall-instance prediction, where *MvFFN* fails to distinguish different wall instances across views.

A first failure mode of *MvFFN* occurs when the predicted *MOIS* field does not consistently separate wall instances across views, producing incorrect wall grouping. As shown in Fig. H.1, these errors propagate through layout chaining and lead to both connectivity and geometric mistakes in the final layout. The failure likely stems from limited training coverage for non-Manhattan structures such as bay windows, where cross-view wall-instance prediction is more ambiguous. Although layout chaining suppresses weak wall hypotheses supported by only a few columns, noisy predictions from distant regions can still introduce incorrect connectivity. Inconsistent annotations for structures such as kitchen islands further worsen this problem under limited training data. More generally, because layout chaining is heuristic-based, it has limited ability to recover from noisy *MOIS* predictions, highlighting the value of a learned post-processing module for robust global layout assembly.



Figure H.2: Failure case caused by geometric errors in the initial *MvFFN* predictions, where inaccurate poses and point maps lead to distorted layout geometry.

A second failure mode arises from geometric errors in the initial *MvFFN* predictions, including inaccurate camera poses and noisy point maps, as shown in Fig. H.2. Because layout chaining builds the global layout from these predicted geometric cues, such errors can propagate to the final reconstruction and produce misaligned walls or distorted room geometry. Although the wall-instance predictions of *MvFFN* enable downstream bundle adjustment, the quality of the final refinement still depends on the accuracy of the initial geometry and correspondences.

I Limitations and Broader Impact

Limitations. The primary system limitation is the scarcity of structured multi-view panoramic training data. To compensate, the system uses two-view *CovisPose* rotation and correspondence priors and vanishing-angle priors during training and inference. These are data-scarcity remedies, not architectural requirements; sufficient multi-view data could remove them. Because the *CovisPose* training pipeline is not public, we cannot adapt it to `HM3D-XV`, and failures under domain shift can propagate to pose and layout predictions. The fundamental bottleneck is the absence of large-scale datasets combining annotated multi-room layouts, dense geometry, multiple overlapping panoramas per room, and views into adjacent rooms through open doorways. Such data would enable end-to-end geometry learning and improve cross-domain generalization. *MvFFN* also assumes straightened equirectangular RGB panoramas from one connected indoor pose graph, with sufficient overlap and known relative camera heights. Sparse coverage, severe occlusion, inaccurate height metadata, non-Manhattan or curved architecture, poor imagery, or out-of-domain scenes may produce inaccurate, partial, or open layouts.

Future directions. The current *Layout Chaining* procedure is deterministic and has no self-correction, so local *MOIS* errors near open doorways or bay windows can propagate into global topology (Fig. H.1). A learned global assembly module is a natural extension; the pre-assembly results in Table C.1 show that the structured local representation already supports this direction. The *MOIS* target—cross-view wall identity and ordered connectivity—could also pretrain panoramic scene-understanding models. Finally, although *MvFFN* supplies the initialization and correspondences required by planar bundle adjustment and *BADGR*, their mixed per-scene results motivate training refinement on *MvFFN* outputs and developing a criterion for deciding when to apply it.

Broader impact and responsible use. This work can support useful applications such as lower-cost indoor digital twins, AR/VR walkthroughs, assistive indoor navigation, and robotics mapping. At the same time, accurate reconstruction of private interior layouts can create privacy and security risks if used without consent, and erroneous reconstructions may be harmful in downstream navigation, robotics, or planning systems if treated as safety-critical. We therefore intend the method and released assets for research and consented indoor reconstruction, not for covert mapping, access-control, insurance, credit, law-enforcement, or other consequential decisions. Released annotations and code should follow the licenses and access restrictions of the underlying datasets, document intended use and limitations, and avoid distributing raw indoor imagery beyond what the original dataset terms permit.

J More Qualitative Results

J.1 ZInD-XV Results

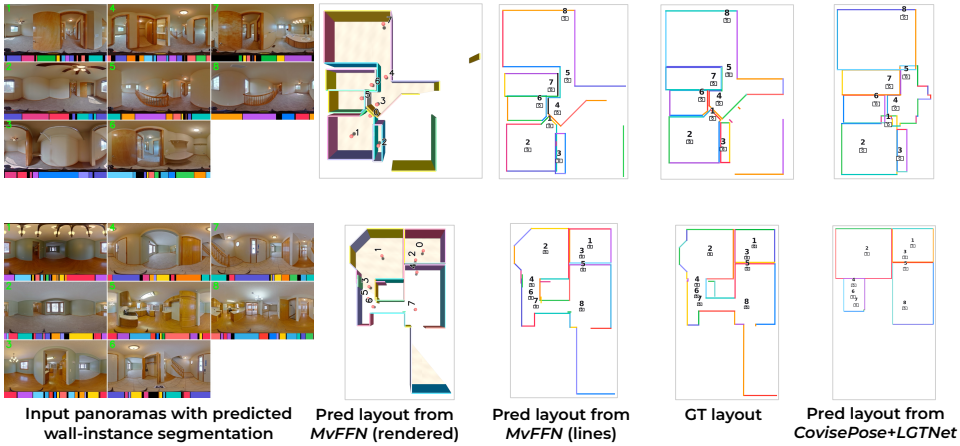


Figure J.1: Qualitative comparisons on `ZInD-XV` test set.

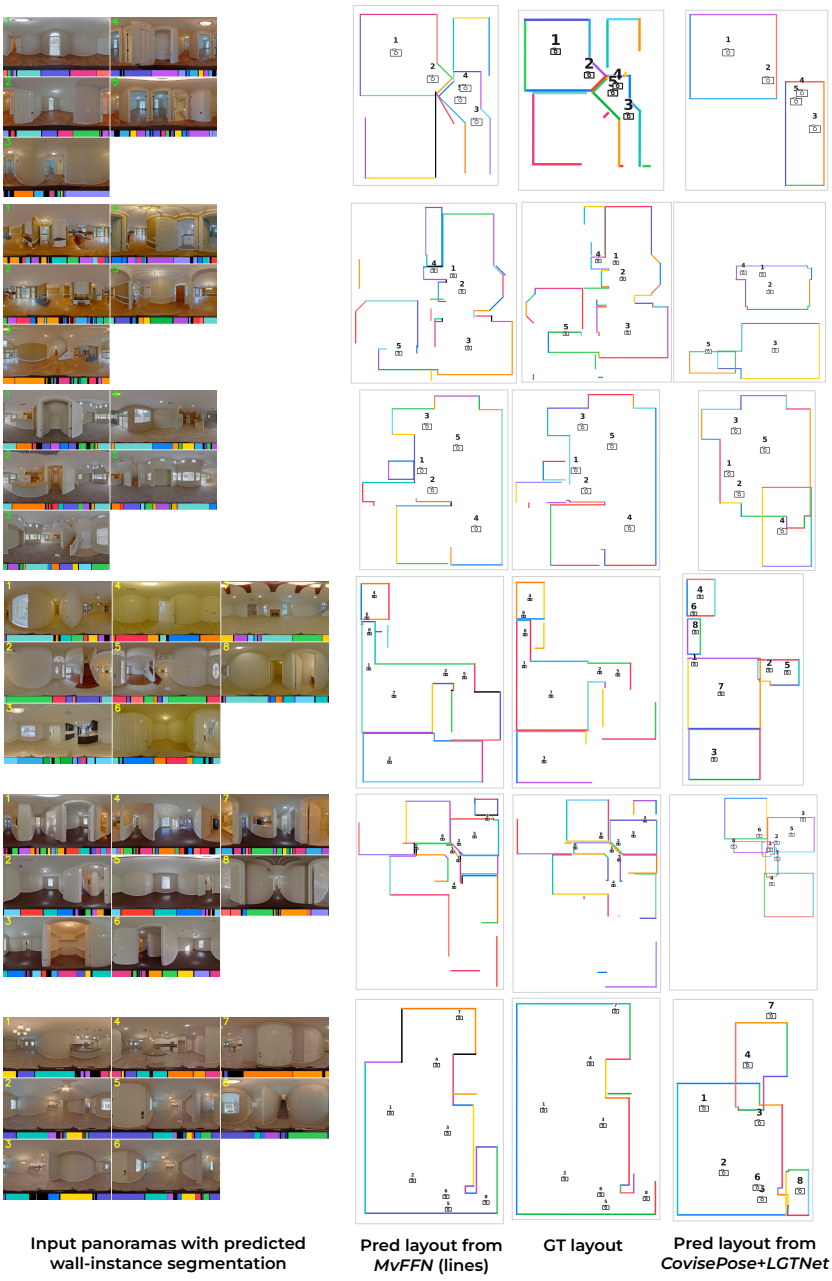


Figure J.2: More Qualitative comparisons on ZInD-XV test set.

Figs. J.1 and J.2 show additional qualitative examples on ZInD-XV from 5-view and 8-view samples. The first columns visualize the predicted multi-view wall instances with consistent colors across panoramas, followed by the predicted top-down layout, the ground-truth layout, and the strongest modular baseline, *LGT-Net* with *CovisPose*. Across both simple rooms and larger multi-room scenes, *MvFFN* more reliably reconstructs complete room shapes, remains accurate under partial occlusion, and recovers distant wall instances and neighboring room structure visible through doorways. Consistent with the quantitative pose and layout results, the predicted camera poses and wall instances show fewer duplications and omissions than the baseline. In contrast, the baseline often predicts partial room shapes, splits large rooms into smaller fragments, and fails to reason jointly across views, leading to incorrect boundaries, missing wall structure, and redundant contours.

J.2 HM3D-XV Results

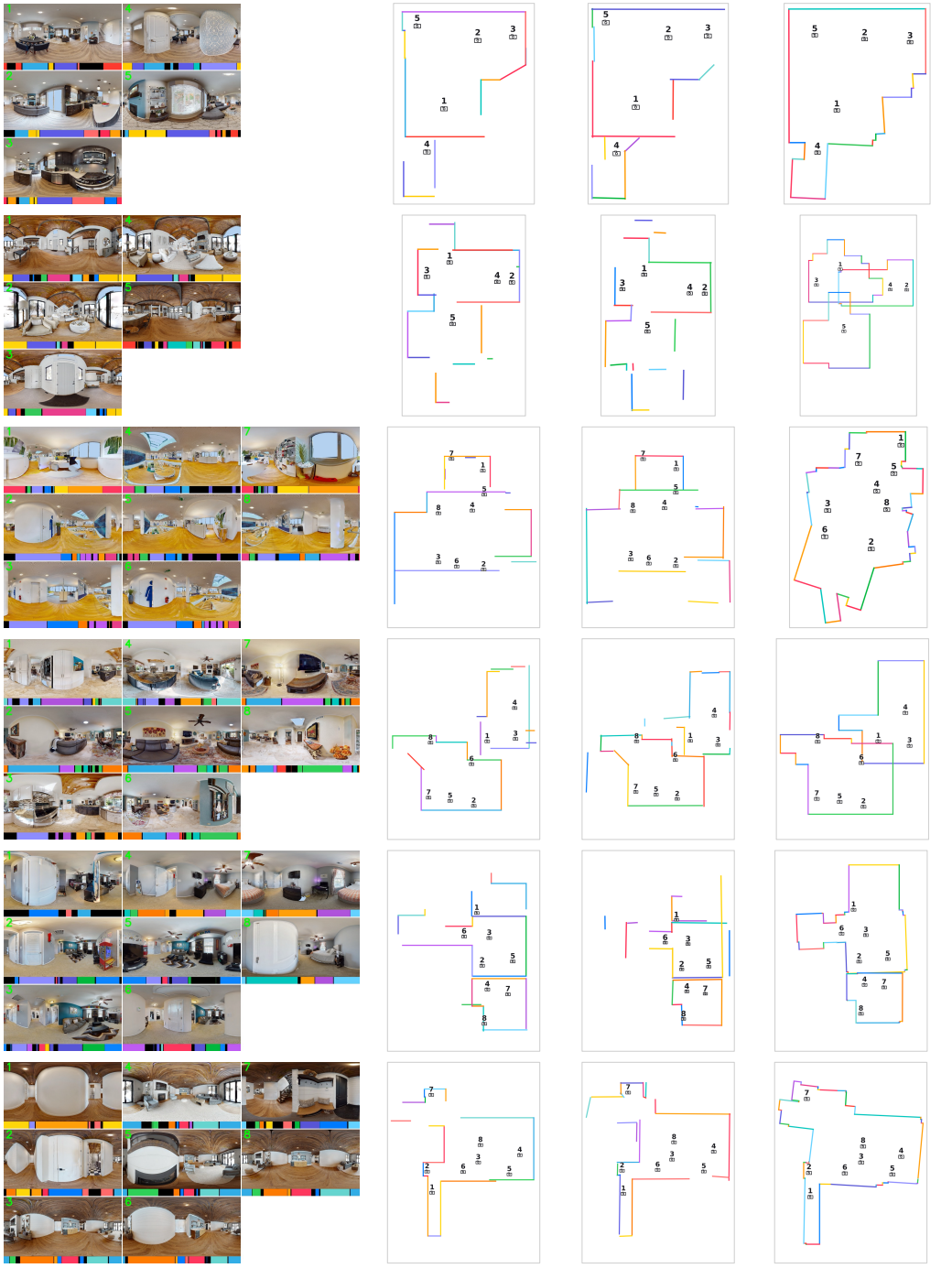


Figure J.3: Qualitative comparisons on HM3D−XV test set.

Figure J.3 shows HM3D−XV reconstruction examples under the furnished, mesh-rendered domain shift. As in ZInD−XV, *MvFFN* can recover globally consistent wall instances and coherent room-level structure, reducing fragmented or duplicate walls relative to modular geometry-first baselines.

Because HM3D−XV is smaller, noisier, and more sparsely annotated, these qualitative results illustrate representative successful cases rather than the full distribution of outcomes.

K Assets, Licenses, and Responsible Release

Assets, licenses, and data release. This work uses existing datasets and research artifacts, including ZInD Cruz et al. [2021], HM3D/HM3DSem Yadav et al. [2023], and the baseline or dependency code used in our comparisons. Table K.1 lists the version or access date, citation, role, license or access terms, and release treatment for each asset, including pretrained weights and baseline implementations where applicable. ZInD−XV and HM3D−XV are derived annotation assets. Unless further restricted by the original dataset terms, we release our derived annotations, splits, metadata, and evaluation files under CC BY-NC 4.0. This license applies only to the annotations and metadata created in this work, not to the underlying raw panoramas, meshes, textures, renderings, or other source assets. We do not redistribute raw restricted source data; users must obtain ZInD and HM3D/HM3DSem from the original providers and comply with their respective licenses, terms of use, and access restrictions. We also provide conversion and evaluation scripts, together with a dataset card or README documenting label definitions, annotation and conversion pipelines, statistics, splits, file formats, known limitations, intended use, out-of-scope use, and maintenance policy.

Table K.1: Existing and newly derived assets used in this work. We release only assets created by this work under the stated terms and do not redistribute raw restricted source data.

Asset	Role in this work	Version / access date	License or terms	Release treatment
ZInD-XV annotations	New derived annotation asset	v1.0	CC BY-NC 4.0, subject to original ZInD terms	We release derived wall-instance labels, splits, metadata, conversion scripts, and evaluation files only.
HM3D-XV annotations	New derived annotation asset	v1.0	CC BY-NC 4.0, subject to original HM3D and HM3DSem terms	We release derived annotations, splits, metadata, conversion scripts, and evaluation files only.

Dataset card summary. ZInD−XV contains derived multi-view floor-plan annotations, globally indexed wall instances, room-wise wall sequences, and per-panorama column-wise floor-boundary and wall-instance labels. HM3D−XV contains analogous derived annotations for residential HM3DSem scenes. The released files include train/validation/test splits, per-floor metadata, per-panorama labels, wall-instance identifiers, room connectivity, and evaluation tuples. The README specifies the file format, coordinate convention, label definitions, preprocessing steps, conversion scripts, and evaluation protocol. The intended use is research on indoor layout reconstruction, camera pose estimation, and structured multi-view scene understanding. Out-of-scope uses include covert mapping of private spaces, security-sensitive reconstruction without consent, and consequential decision-making based on predicted layouts.

Responsible release. Indoor floor-plans can reveal sensitive information about private spaces. To reduce privacy, security, and license risks, we do not redistribute raw restricted panoramas, meshes, textures, renderings, or floor-plan source files. The released package contains only compatible derived annotations, metadata, splits, and scripts. Users are required to obtain the original datasets from their providers and comply with the corresponding terms. The release documentation states intended uses, out-of-scope uses, limitations, and access restrictions. We will release all code, trained checkpoints, and utilities upon acceptance.

Annotation provenance. Manual annotation and cleanup for ZInD−XV and HM3D−XV were performed by the authors/research team using custom annotation and quality-control tools. No crowdworkers, contractors, or human-subject study participants were recruited. The released documentation describes the annotation interface, label definitions, quality-control procedure, and cleanup criteria.

L Reproducibility and Compute

Compute resources. All *MvFFN* experiments were run on 8 or 16 A100 GPUs with 40GB memory per GPU, using 6 12 cpus per gpu. Note that the experiments can also be conducted on lower-tier gpus. Stage-1 geometry pre-training used approximately 2 days for 80k iterations, and stage-2 layout semantic training used approximately 2.5 days for 80k iterations. We used a batch size of [10,7,4,2] for 2, 3, 5, 8 views with 6 12 dataloader workers per process.

L.1 Inference Scalability

The logical *MOIS* output has shape $N \times W \times N \times W \times K$, but it is computed through query–patch dot products rather than materialized as a dense 5D tensor; tiled FlashAttention similarly avoids materializing the full attention matrix. Table L.1 shows approximately linear growth in memory and runtime over the tested range, with 16 panoramas requiring 3.93 GB.

Table L.1: Batch-size-1 inference in FP32 at 512×1024 resolution on one 48 GB NVIDIA RTX A6000. The 12- and 16-view settings were profiled for compute only, not evaluated for accuracy.

N	Peak mem. (GB)	Runtime (ms)	N	Peak mem. (GB)	Runtime (ms)
2	1.30	45.1	8	2.45	149.2
3	1.49	58.7	12	3.18	226.9
5	1.87	93.2	16	3.93	320.9

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 state the paper's main contributions: the *MOIS* formulation, the *MvFFN* model for predicting local cross-view wall structures, and the `ZInD-XV` dataset. They also describe how local predictions are assembled into global layouts and how the resulting wall correspondences are evaluated for downstream refinement. The experiments support these claims on sparse, unposed indoor panorama layout reconstruction. The scope is limited to the stated indoor 360° panorama setting, including straightened panoramas, connected pose graphs, sufficient co-visibility, and known relative camera heights, as discussed in the method and the supplementary limitations.

   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: The supplementary limitations section discusses the main assumptions and failure modes, including straightened equirectangular panoramas, a connected indoor pose graph, sufficient overlap, known relative camera heights, low co-visibility, occlusion, non-Manhattan or curved architecture, low-light or textureless images, partial/open layouts, refinement sensitivity, and scaling with the number of views and image columns.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results, theorems, or proofs; the mathematical notation defines the representation, architecture, losses, and optimization procedures.

   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: The method and the supplementary material describe the architecture, MOIS targets, two-stage training, Layout Chaining, refinement, dataset construction, tuple construction, metrics, baselines, losses, optimizer settings, sampling schedules, batch sizes, and compute setup needed to reproduce or verify the main empirical claims.

   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: [No]

   Justification: We do not provide open-access code with the NeurIPS submission. The paper and the supplementary material provide the method, training, evaluation, dataset-construction, and baseline details needed to understand and verify the main results. We commit to releasing code upon acceptance. Raw `ZInD` and `HM3D`/`HM3DSem` source data are not redistributed due to the original dataset terms. We also plan to release derived `ZInD-XV` and `HM3D-XV` annotations, metadata, splits, conversion scripts, evaluation files, and potentially the annotation GUI under terms compatible with the original datasets.

   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: The experimental sections specify the evaluated datasets, view-count settings, baselines, metrics, tuple construction, and main ablations. The supplementary material provides training losses, optimizer settings, sampling schedules, dataset preparation, Layout Chaining details, baseline implementations, evaluation metrics, batch sizes, and compute resources.

   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: [Yes]

   Justification: The paper reports distributional summaries for the pose experiments, including mean, standard deviation, and median over evaluated panoramas, and reports layout and refinement results as aggregate metrics over fixed test tuple sets. These summaries characterize variability across evaluated examples, although the paper does not report multi-seed training variability or formal hypothesis tests.

   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: The supplementary material reports the compute resources used for the main MvFFN experiments, including GPU type, GPU memory, number of GPUs, CPU/dataloader resources, training time for both stages, iteration counts, and view-dependent batch sizes. These details provide the main hardware and runtime requirements needed to reproduce the reported training setup.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics to the best of the authors' knowledge: it uses existing indoor-scene datasets and author-created derived annotations, does not involve deception or weaponization, documents privacy and security risks, respects original dataset access restrictions, and uses a responsible-release strategy for derived assets.

   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: The supplementary broader-impact discussion covers positive impacts such as indoor digital twins, AR/VR walkthroughs, assistive indoor navigation, and robotics mapping, as well as negative impacts such as privacy and security risks from interior reconstruction, unauthorized mapping, and errors in downstream navigation, robotics, or planning systems.

    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: [Yes]

Justification: Although the model is not a high-risk generative model, indoor layout reconstruction can raise privacy and security concerns. The supplementary material describes safeguards including derived-annotation-only distribution where required, no redistribution of raw restricted panoramas, meshes, textures, renderings, or floor plan source files, user compliance with original dataset terms, intended-use documentation, out-of-scope-use documentation, and release limitations.

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: The supplementary assets and licenses section documents the use of existing datasets and research artifacts, including ZInD, HM3D/HM3DSem, and baseline or dependency code, and describes the license or access restrictions that govern the released derived assets. The derived ZInD-XV and HM3D-XV annotations are released under CC BY-NC 4.0 only where compatible with the original dataset terms, and raw restricted source data are not redistributed.

Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: The paper introduces ZInD-XV and HM3D-XV derived annotation assets, and the supplementary material documents their construction, label definitions, released files, statistics, splits, file formats, access restrictions, intended use, out-of-scope use, known limitations, license, responsible-release policy, and annotation provenance.

- The answer [N/A] means that the paper does not release new assets.

- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [N/A]

    Justification: The paper does not report crowdsourcing experiments or human-subject studies. Manual annotation and cleanup for ZInD-XV and HM3D-XV were performed by the authors/research team using custom annotation and quality-control tools; no crowdworkers, contractors, or human-subject study participants were recruited.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The study does not involve direct interaction with human participants or collection of new human-subject data; it uses existing indoor-scene datasets and author-created derived layout annotations.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

Justification: LLMs are not an important, original, or non-standard component of MvFFN, dataset construction, or the experimental methodology; any use for writing, editing, or formatting does not affect the core scientific contribution.

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.