

Structured Indoor Reconstruction from Sparse Unposed Panoramas: Coarse Prediction to Generative Geometric Refinement

Yuguang Li

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington
2026

Reading Committee:
Linda Shapiro, Chair
Alex Colburn
Sing Bing Kang
Jenq-Neng Hwang

Program Authorized to Offer Degree:
Electrical and Computer Engineering

©Copyright 2026

Yuguang Li

University of Washington

Abstract

Structured Indoor Reconstruction from Sparse Unposed Panoramas: Coarse Prediction to Generative Geometric Refinement

Yuguang Li

Chair of the Supervisory Committee:

Linda Shapiro

Computer Science and Engineering

Accurate indoor floor plans and camera poses are important for applications such as digital twins, augmented reality, real estate visualization, and immersive walkthrough generation. However, reconstructing structured indoor layouts from sparse, unposed, wide-baseline panoramic images remains difficult due to pose uncertainty, limited cross-view overlap, occlusion, and the inherently global nature of floor-plan structure. This thesis studies learning-based methods for structured indoor reconstruction in this setting, with the goal of recovering precise camera pose, geometry, and multi-room floor-plan layouts from sparse panoramic observations.

The thesis begins by establishing this problem setting through practical reconstruction pipelines, public datasets, and targeted studies of wide-baseline pose and layout estimation. Early human-in-the-loop systems show that accurate cross-view alignment is essential for constructing precise global floor plans, while also revealing the limitations of pipelines that primarily stitch together single-view layouts. To support systematic study, the thesis develops public benchmarks and cross-view annotations, and investigates targeted subproblems such as wide-baseline pose estimation and joint multi-view layout prediction. These early efforts establish two key insights: reliable indoor reconstruction depends critically on cross-view reasoning, and direct prediction of full multi-view, multi-room layouts is difficult to learn under sparse data, pose ambiguity, and noisy supervision.

Motivated by these observations, the thesis reformulates the problem through *Multi-view Ordered Wall Instance Segmentation (MOIS)*, which represents layouts using cross-view wall instances and within-room connectivity rather than direct global wall-sequence prediction. Under this formulation, the thesis introduces *MvFFN*, a feed-forward multi-view model that jointly predicts coarse camera poses, dense global geometry, and wall-instance-level structural outputs, which are assembled into globally consistent multi-room floor plans.

This makes the reconstruction problem substantially more tractable and enables automated recovery of structured layouts from sparse panoramas.

The thesis then studies refinement beyond feed-forward prediction. While *MvFFN* provides strong coarse reconstructions, challenging scenes with wide-baseline ambiguity, sparse observations, and noisy boundary evidence still require scene-level correction. This motivates *BADGR*, whose central insight is that learned generative priors and iterative geometric optimization should not be treated as separate stages, but as complementary processes that can be learned to improve one another. *BADGR* combines structural generation and bundle-adjustment-style refinement to jointly improve pose and layout consistency under global scene constraints. Experiments show improved wall, junction, and pose accuracy over prior baselines on challenging indoor benchmarks. Together, these results show that accurate indoor reconstruction from sparse panoramas requires the integration of cross-view reasoning, structured intermediate representations, learned structural priors, and global optimization.

Acknowledgements

Completing this dissertation has been a long and deeply meaningful journey. My PhD path began in 2014. I paused after receiving my master’s degree in 2017 to explore industry, understand real-world problems, and find the research direction that I truly wanted to pursue. Returning to the PhD in 2024 gave me the opportunity to come back to a problem that had stayed with me for many years, and to approach it with a clearer sense of purpose, maturity, and independence.

I am deeply grateful to Professor Linda Shapiro for her support, understanding, and guidance throughout this journey. Thank you for taking me in as my advisor, giving me the freedom to pursue this research direction, and offering advice that helped me navigate both the academic and personal sides of the PhD. Your support made it possible for me to complete this work.

I owe special thanks to Dr. Alex Colburn, who has been much more than a collaborator and mentor throughout this thesis. Our long Friday meetings became the place where many of the ideas in this dissertation were shaped, challenged, and refined. Even though much of this work was pursued as a personal research project outside a formal project structure, you consistently made time to discuss ideas, review directions, and help me think more clearly. I learned a great deal from you about research taste, research methods, technical writing, and how to turn an intuition into a coherent scientific contribution. I am also grateful that, through this process, we became friends.

I would also like to thank Dr. Sing Bing Kang for his generous mentorship and advice. Our conversations helped me think more clearly about research, career decisions, and the broader direction of my work. Many of those lessons will stay with me well beyond this dissertation. I am also grateful to Professor Jenq-Neng Hwang and Professor Sawyer B. Fuller for serving on my committee and for supporting my PhD.

I am grateful to my labmates and friends who provided a supportive environment and a sense of community. In particular, I would like to thank Zixuan Liu and Yichuan Deng for two years of collaboration, many in-depth discussions, and countless experiments that helped shape this work. I would also like to thank my friend and collaborator, Dr. Ivaylo Boyadzhiev, for his support in strengthening our experimental design and improving the storytelling of our papers. Without the support and contributions of all three of you, this work would not have reached the level of a CVPR Highlight paper and a strong recent NeurIPS submission.

I am also grateful to everyone from Linda’s lab, including Dr. Saygin Seyfioglu, Wisdom Ikezogwo, Mahtab Bigverdi, Rustin Soraki, Simon Zou, and Huayu Wang, for the research environment, conversations, and support along the way. I would also like to thank my

friends from the GRAIL lab, including Baback Elmieh, Bowei Chen, and Jingwei Ma, for their friendship, conversations, and encouragement throughout this journey.

I would also like to thank the sponsors and supporters who made this research path possible, especially Dr. Chris Buehler and Josh Weisberg from Zillow. Thank you for supporting this PhD with your trust, resources, and encouragement. This work would have been much harder without your support, from computing infrastructure and data access to financial support that gave me the freedom to pursue this research seriously.

Finally, I would like to thank my family for their tremendous support during these years, especially during the final stretch of the PhD. Their encouragement, patience, and belief in me gave me the stability and motivation to finish this thesis.

TABLE OF CONTENTS

Section Number	Page
Chapter 1: Introduction	1
Chapter 2: Preliminary Human-in-the-Loop Multi-View Floor Plan Reconstruction Pipeline	5
2.1 Introduction	5
2.2 Zillow Indoor Dataset: Annotated Floor Plans With 360° Panoramas and 3D Room Layouts	5
2.2.1 Zillow Indoor Dataset (ZInD) Pipeline	6
2.2.2 Key insights from the ZInD pipeline	7
2.3 SALVe: Semantic Alignment Verification for Floorplan Reconstruction from Sparse Panoramas	8
2.3.1 Overview	8
2.3.2 Methods	9
2.3.3 Results	10
2.3.4 Key insights on the ZInD pipeline	12
2.4 PSMNet: Position-aware Stereo Merging Network for Room Layout Estimation	13
2.4.1 Overview	13
2.4.2 Methods	13
2.4.3 Results	15
2.4.4 Key insights from the PSMNet	16
2.5 Conclusion	17
Chapter 3: MvFFN: Multi-view Floor-plan Feed-Forward Network for Unposed Wide-Baseline Panorama Layout Reconstruction	18
3.1 CoVisPose: An Intermediate Step Toward MvFFN	18
3.2 Introduction	21
3.3 Related Work	22
3.4 Overview	23
3.5 Method	24
3.6 N-view Geometry Encoder Branch	25
3.7 Floor Layout Semantic Branch	25
3.8 Training	27
3.9 Inductive Bias for Training Data Scarcity	28
3.10 Layout Chaining	29
3.11 Experiments	31

3.12 Experiments on ZInD-XV Dataset	31
3.12.1 Pose Estimation Evaluation	31
3.12.2 Floor Layout Estimation Evaluation	31
3.12.3 MOIS Prediction Window	32
3.13 Experiments on HM3D-XV dataset	33
3.14 Camera Pose Estimation	34
3.15 Layout Estimation	35
3.16 Ablation Study	36
3.16.1 Architecture.	36
3.16.2 Inductive Bias	36
3.16.3 MOIS Prediction Window	37
3.17 Qualitative Results	38
3.17.1 ZInD-XV Results	38
3.17.2 HM3D-XV Results	41
3.18 Failure Cases	42
3.19 Dataset Preparation	43
3.19.1 ZInD-XV Dataset Preparation	43
3.19.2 HM3D-XV Dataset Preparation	43
3.20 Conclusion	45

Chapter 4: BADGR: Bundle Adjustment Diffusion Conditioned by GRadients for Wide-Baseline Floor Plan Reconstruction	46
4.1 Introduction	46
4.2 Related Work	48
4.3 Problem Statement	49
4.4 Pipeline Overview	50
4.5 Coarse Scene Initialization	50
4.6 Scene Representation for BA Optimization	51
4.7 Bundle Adjustment Diffusion	52
4.8 Experiments	56
4.9 Experiments with FloorPlan-60K Data	56
4.9.1 Baseline Models	57
4.9.2 Results	58
4.10 Reprojection Errors	59
4.11 Experiments with RPLAN	60
4.12 Cross dataset training and validation	63
4.13 Compatibility with <i>MvFFN</i> Initialization	63
4.14 Ablation Studies	64
4.15 Qualitative Results	64
4.16 Discussion	65
4.17 Failure Cases	66

4.18 Conclusion	67
Chapter 5: Conclusion	68
5.1 Summary	68
5.2 Future Work	69
A Appendix	77
A.1 Publication Notes	77
A.2 Implementation Details of <i>MvFFN</i> Experiments.	78
A.3 Pose Evaluation	78
A.4 Layout Baseline Pipelines	79
A.5 More Qualitative Results from <i>BADGR</i>	80

LIST OF FIGURES

Figure Number	Page
1.1 Pipeline overview and methods studied in this thesis.	4
2.1 ZInD dataset overview	5
2.2 ZInD annotation pipeline	6
2.3 Complete room layout and secondary panorama localization	7
2.4 <i>SALVe</i> pipeline overview.	8
2.5 <i>SALVe</i> pipeline architecture	9
2.6 <i>SALVe</i> Alignment-verifier input	10
2.7 Qualitative comparison of floor-plan reconstruction	12
2.8 Overview of <i>PSMNet</i>	13
2.9 Architecture of <i>PSMNet</i>	14
2.10 Qualitative results on ZInD with noisy pose	16
3.1 CoVisPose prediction targets	18
3.2 CoVisPose architecture	19
3.3 CoVisPose qualitative results	20
3.4 Overview of <i>MvFFN</i>	23
3.5 Architecture of <i>MvFFN</i>	26
3.6 Effect of inductive bias.	37
3.7 Qualitative comparisons on ZInD-XV test set.	38
3.8 More qualitative comparisons on ZInD-XV test set.	39
3.9 More qualitative comparisons on ZInD-XV test set.	40
3.10 Qualitative comparisons on HM3D-XV test set.	41
3.11 Failure case caused by incorrect <i>MOIS</i> wall-instance prediction.	42
3.12 Failure case caused by geometric errors in the initial <i>MvFFN</i> predictions. .	42
4.1 BADGR overview	47
4.2 BADGR architecture	49
4.3 Overview of coarse scene initialization.	50
4.4 Column-wise planar BA module	54
4.5 Qualitative layout and pose refinement results	58
4.6 Qualitative RPLAN results	62
4.7 Failure case: coarse initialization	66
4.8 Failure case: incorrect wall length	67
4.9 Failure case: inconsistent floor heights	67
A.1 More qualitative BADGR results, page 1	81
A.2 More qualitative BADGR results, page 2	82
A.3 More qualitative BADGR results, page 3	83

LIST OF TABLES

Table Number		Page
2.1	Results of global pose estimation on the ZInD test set.	11
2.2	Floor-plan reconstruction results on the ZInD test set.	11
2.3	Quantitative evaluation by spatial overlap	15
3.1	Camera pose evaluation on the ZInD-XV dataset.	31
3.2	<i>MvFFN</i> layout evaluation on ZInD-XV.	32
3.3	<i>Pooled Ordered Wall-Slot IoU$\uparrow$</i> (%) for raw <i>MOIS</i> predictions before <i>Layout Chaining</i>	33
3.4	Camera pose evaluation on the <i>CovisPose</i> -feasible subset of HM3D-XV. . . .	35
3.5	<i>MvFFN</i> layout accuracy on HM3D-XV.	35
3.6	Ablation study of <i>MvFFN</i> on ZInD-XV.	36
3.7	<i>MvFFN</i> layout accuracy on ZInD-XV.	37
4.1	Pose and layout errors on ZInD	59
4.2	Reprojection errors for wall-assigned columns	60
4.3	Pose and layout errors on RPLAN	61
4.4	Errors on RPLAN with simulated floor-boundary noise	61
4.5	Cross-dataset BADGR evaluation on ZInD	63
4.6	Matched-wall distance before and after refinement on ZInD-XV. Distances are in cm.	64
4.7	Ablation analysis of BADGR variants	65
4.8	GraphCovis vs BADGR on panorama groups	66

Chapter 1

INTRODUCTION

Indoor environments are structurally rich, visually complex, and central to many real-world applications. Accurate reconstruction of indoor spaces enables not only geometric modeling and visualization, but also navigation, robotics, digital twins, augmented reality, and emerging workflows such as floor-plan-guided virtual home editing and reimagining with generative models. Among the many ways to represent an indoor environment, floor plans occupy a particularly important role: they provide a compact, interpretable, and spatially grounded description of walls, rooms, openings, and their organization. Unlike dense geometric reconstructions, which preserve fine detail but often remain unstructured, floor plans distill the architectural structure of a home into a form that is both semantically meaningful and operationally useful.

Despite their simplicity as an output representation, floor plans are difficult to recover from images. Indoor scenes are filled with clutter, occlusion, repeated patterns, reflective surfaces, and large appearance variation. More fundamentally, structural reconstruction depends not only on recognizing local visual evidence, but also on resolving global geometric ambiguity: walls must align across views, rooms must connect consistently, and the recovered structure must remain coherent under uncertainty in camera pose and scene geometry. These challenges become especially pronounced when reconstruction must be performed from sparse visual inputs rather than densely scanned observations.

The sensing modality strongly shapes the tradeoff between capture practicality and reconstruction difficulty. Laser scanners produce dense point clouds with highly accurate geometry and precise cross-view registration, making large-scale structural reconstruction relatively reliable, but they are slow and require specialized hardware. RGB video is far more accessible and benefits from temporal continuity, which often makes pose estimation easier, yet its limited field of view makes full structural coverage labor-intensive. In contrast, 360° panoramas offer a compelling middle ground for practical indoor capture: each image provides broad spatial coverage, reducing the number of viewpoints required while preserving rich visual information. At the same time, sparse panoramic capture introduces a challenging reconstruction setting, since wide baselines, limited overlap, and the absence of dense sequential motion make camera registration and structural recovery substantially more ambiguous than in video-based pipelines.

This thesis studies floor-plan reconstruction from sparse, unposed 360° panoramas. The goal is to recover a structured layout, in which a single-room or multi-room floor plan is represented as an ordered set of wall or structural boundary instances rather than dense geometry. While single-view indoor geometry and single-room layout estimation have been studied extensively [89, 66, 28, 76], reconstruction in this setting is substantially more

difficult. In real homes, panoramic captures are often sparse for practical and economic reasons, resulting in wide baselines, limited overlap, and unknown camera pose [30, 17]. These environments also frequently contain featureless regions and strong visual similarity across rooms, making reliable registration difficult and limiting the effectiveness of classical feature-based SfM. As a result, floor-plan reconstruction from sparse panoramas is challenging because floor plans are inherently global representations: their correctness depends not only on local image evidence, but also on consistent spatial relationships across views. This difficulty is further amplified by uncertain camera pose, since pose errors directly degrade cross-view alignment and floor-plan accuracy, while coherent global structure can in turn provide strong constraints for pose estimation. Robust reconstruction in this setting therefore requires treating pose, geometry, and layout as tightly coupled aspects of the same problem rather than as independent stages.

The thesis first examines floor-plan reconstruction in a practical sparse-panorama setting derived from an early human-in-the-loop production pipeline and formalized in ZInD, a public dataset [17]. In this setting, accurate cross-view alignment is essential for constructing precise global floor plans. It highlights both the value of high-spatial-accuracy pose and layout estimation and the limitations of building a global floor plan primarily by stitching together single-view layouts. As a result, it exposes several coupled subproblems, including wide-baseline pose estimation, structural correspondence across views, and joint reasoning over layouts spanning multiple rooms. The earliest part of the thesis addresses these challenges through two early pre-transformer methods: *SALVe* [30], which studies wide-baseline indoor pose estimation and floor-plan assembly from sparse panoramas, and *PSMNet* [77], which studies joint multi-view layout estimation and begins to couple pose and structure more directly. Although these methods do not yet provide a unified solution, they establish the central role of cross-view reasoning in indoor reconstruction and motivate the more integrated feed-forward and refinement-based models developed in later chapters.

CoVisPose [27] provides additional context for the next stage of the thesis. As a two-view pose estimation method for wide-baseline indoor panoramas, it demonstrates that robust pose estimation benefits from dense cross-view reasoning over visual overlap, correspondence, and layout geometry, especially under low overlap, strong occlusion, and repetitive indoor structure. This observation motivates the transition from loosely coupled pose and layout estimation to a unified multi-view reconstruction formulation.

The next stage of the thesis introduces *MvFFN*, a feed-forward multi-view model for reconstructing camera pose and structured floor-plan layouts from sparse, unposed, wide-baseline RGB panoramas. Direct prediction of a multi-view, multi-room floor plan is difficult: real-world scenes contain ambiguous room partitions, incomplete observations, and noisy or inconsistent annotations, making local errors easy to amplify into global structural mistakes. Moreover, reconstruction requires not only geometric reasoning, but

also cross-view semantic reasoning to resolve wall identities, infer within-room connectivity, and reduce pose ambiguity. To make this problem tractable, the thesis introduces *Multi-view Ordered Wall Instance Segmentation (MOIS)*, a formulation that replaces direct global wall-sequence prediction with local cross-view wall instances and within-room connectivity. Under this formulation, *MvFFN* predicts coarse poses, dense global geometry, and wall-instance-level structural outputs with globally consistent identities and local connectivity. These predictions are assembled into multi-room floor plans through a Layout Chaining (LC) pipeline, significantly improving wall and junction level accuracy and yielding a wall-centered representation that is compatible with downstream refinement.

The final technical stage of the thesis introduces *BADGR* [36], which addresses a key limitation of feed-forward reconstruction: a single forward pass is often insufficient to enforce global cross-view consistency in challenging scenes. While *MvFFN* provides strong coarse predictions of pose and layout, sparse observations, noisy boundary evidence, and wide-baseline ambiguity can still lead to globally inconsistent structure. At the same time, floor plans exhibit strong geometric regularities and structural priors, and large amounts of training data are naturally available in floor-plan form. This raises a broader question: how can learned knowledge of plausible floor-plan structure be used to improve both pose and layout reconstruction, while remaining compatible with iterative global optimization? *BADGR* answers this by combining bundle adjustment with a learned generative refinement process, allowing structural priors and reprojection-based consistency to work together. In this way, it refines poses and layouts jointly using global context, extending the thesis from feed-forward multi-view prediction to learned geometric and structural refinement.

This thesis presents three interconnected projects that progressively address the challenge of indoor reconstruction from sparse panoramic imagery. At a high level, the reconstruction pipeline studied in this dissertation proceeds from sparse unposed panoramas through single-view layout estimation, coarse pose and global layout assembly, globally unique layout fusion, and final geometric refinement. Chapter 2 presents earlier targeted methods for wide-baseline pose estimation and joint multi-view layout prediction, which help establish the importance of cross-view consistency for both localization and structure recovery while also revealing the limitations of pipeline-based reconstruction. The main focus of this thesis lies in the more unified formulations introduced in Chapter 3 and Chapter 4. Chapter 3 introduces *MvFFN*, a unified feed-forward multi-view model that reformulates floor-plan recovery through ordered wall-instance prediction and jointly estimates pose, geometry, and structured layout. Chapter 4 introduces *BADGR*, a generative refinement framework that combines bundle adjustment with learned structural priors to improve poses and layouts from coarse multi-view reconstructions. Finally, Chapter 5 concludes the dissertation and discusses future directions. Together, these projects trace a progression from early partially coupled solutions to more unified models that

integrate pose estimation, geometry, layout reasoning, and refinement within a common reconstruction framework.

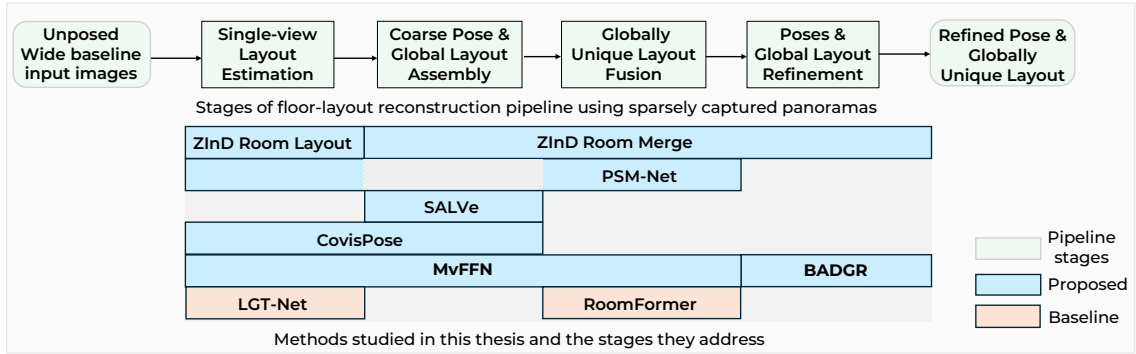


Figure 1.1: Top: Floor layout reconstruction pipeline from wide-baseline panoramas studied in this thesis. Bottom: The proposed methods from Chapter 2 – 4 and the corresponding stages they address in the reconstruction pipeline above.

Fig. 1.1 provides a stage-level overview of this reconstruction pipeline and situates each method within it. This view highlights how the earlier methods in Chapter 2 address individual or partially coupled subproblems, while the main methods in Chapter 3 and Chapter 4 increasingly unify multiple stages of the reconstruction process. Such unification is important because sparse panoramic reconstruction is fundamentally shaped by cross-view ambiguity: pose, geometry, and layout errors interact strongly, and treating these stages more jointly allows structural and geometric constraints to reinforce one another throughout reconstruction.

Chapter 2

Preliminary Human-in-the-loop Multi-view Floor Plan Reconstruction Pipeline

2.1 Introduction

Chapter 2 examines early pipeline-based approaches to floor-plan reconstruction from sparse wide-baseline panoramas. These methods address the problem through a sequence of partially coupled components, including dataset construction, wide-baseline pose estimation, and multi-view layout prediction. While effective in practice, this decomposition exposes fundamental limitations: cross-view consistency is difficult to maintain, and errors in pose and layout propagate across stages. The systems presented in this chapter, including *SALVe* and *PSMNet*, therefore serve not as final solutions but as diagnostic steps that reveal the need for more unified multi-view reasoning, which is developed in Chapter 3.

2.2 Zillow Indoor Dataset: Annotated Floor Plans With 360° Panoramas and 3D Room Layouts

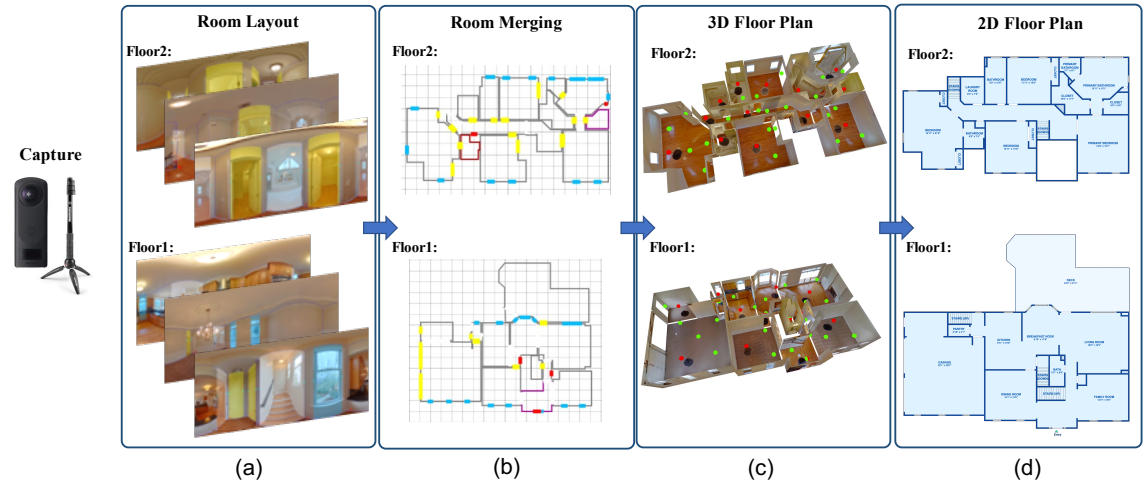


Figure 2.1: The dataset captures real unfurnished homes using sparse panoramic imagery and provides annotations at multiple levels, including room layouts, windows and doors, merged rooms, localized secondary panoramas, textured 3D meshes, and final 2D floor plans. From left to right: (a) annotated RGB panoramas, (b) merged room layouts, (c) textured 3D mesh with primary panoramas in red and secondary panoramas in green, and (d) final cleaned 2D floor plan.

2.2.1 Zillow Indoor Dataset (ZInD) Pipeline

Overview. The goal of the ZInD pipeline is to reconstruct precise floor plans of real homes from sparse 360° panoramic captures at production scale [17]. More than a dataset, it is a human-in-the-loop annotation pipeline designed to recover globally consistent room layouts, camera poses, and final 2D and 3D floor plans across diverse homes, ranging from spaces represented by a single panorama to more complex multi-view configurations. In this setting, robustness and precision are the key design principles: they determine whether the system can scale while maintaining correct global structure across large numbers of homes. As illustrated in Figures 2.1 and 2.2, the pipeline achieves this through a staged process of capture, local room annotation, cross-view room merging, secondary panorama localization, and final floor-plan cleanup.

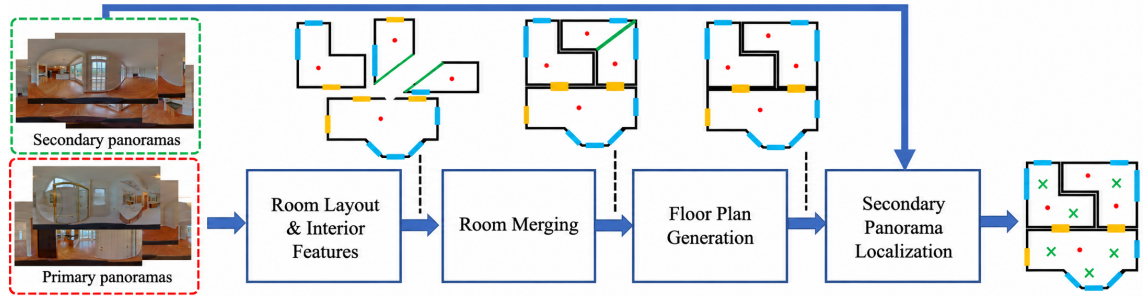


Figure 2.2: The pipeline operates in 2D floor-plan space and produces room layouts, merged floor plans, and localized secondary panoramas; 3D structure is inferred from the 2D annotations under the flat-ceiling assumption. Primary panoramas (·) are used for room layout annotation, while secondary panoramas (×) are localized after the floor plan is assembled. During room merging, **doors**, **windows** and **openings** provide initial snapping cues for stitching layouts before cross-view refinement.

Local structure first. The first stage focuses on recovering local room structure from selected *primary* panoramas, using the primary/secondary panorama terminology and annotation protocol introduced in ZInD [17]. Sparse 360° panoramas provide broad visual coverage and sufficient detail for room-level reconstruction and virtual-tour visualization, making them an efficient acquisition modality for full homes. At this stage, room layouts and interior features are specified from individual views through an annotation UI initialized by automatic layout estimation and feature prediction, including room layout proposals from *HorizonNet* [65]. Large spaces may be partitioned into multiple primary panoramas when needed. This single-view-first strategy makes local annotation practical, but it places a strong burden on consistency: annotators must keep room boundaries and room divisions aligned across related views to avoid substantial reconciliation later. At the same time, this design does not fully exploit complementary evidence from neighboring panoramas, even when another view provides a clearer observation of the same structural boundary.

Global consistency next. The second stage turns these local layouts into a global floor plan. Here, the central requirement is cross-view consistency: the merging and localization workflow is built around checking whether layouts, poses, and semantic structure agree when one view is projected into another, as shown in Figure 2.3. Annotators refine pose alignment through an interactive UI that allows projected layouts to be dragged directly in image space. In practice, this only works when both layout estimates and camera poses are sufficiently accurate. If either is wrong, cross-view agreement degrades and the annotation process becomes unstable. The same recovered structure is then used to localize *secondary* panoramas, again through cross-view verification and UI-assisted refinement, followed by final cleanup to produce the completed floor plan annotations.

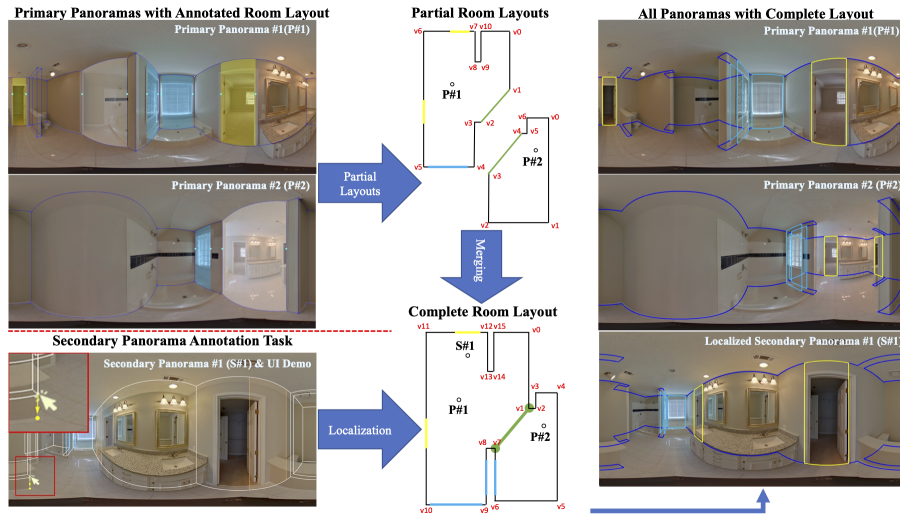


Figure 2.3: Top and center: complete layouts of complex spaces obtained by merging partial layouts from multiple panoramas. Both room merging and secondary panorama localization use the same projection-based UI, where annotators refine alignment by dragging an inherited layout to match image corners. Right: the resulting shared layout from the viewpoints of three localized panoramas.

2.2.2 Key insights from the ZInD pipeline

Beyond serving as a dataset and annotation system, ZInD exposes several practical insights that directly shape the automated methods developed later in this thesis.

Pose and layout are the two core components of floor-plan reconstruction. A key design choice in ZInD is to separate floor-plan reconstruction into local layout estimation from selected *primary* panoramas and cross-view pose estimation for global stitching. This split makes the annotation pipeline practical: room geometry can be specified from one view at a time and later assembled into a floor plan. At the same time, it exposes an important limitation of single-view geometry. Structural details that are weak, occluded, or far away in one panorama may be clearer in another, yet the annotation must commit to a single-view layout and then manually maintain consistent room boundaries and room

divisions across related views. This requires substantial human effort and reveals that cross-view understanding is already needed even before global merging. In other words, if cross-view reasoning is necessary to correct and reconcile single-view layouts, it naturally motivates automated methods that use cross-view information earlier to address these limitations directly.

Cross-view consistency is the key requirement for robust large-scale reconstruction.

The pipeline also reveals that annotation is only correct when layouts align consistently across views. This makes cross-view consistency the practical criterion for whether a reconstruction is acceptable. As a result, pose and layout become tightly coupled: good poses are needed to verify layout consistency, while consistent layout structure provides strong cues for refining pose. More broadly, the ZInD pipeline shows that practical large-scale floor-plan generation depends on solving a small set of tightly related problems well: accurate cross-view pose estimation, reliable room layout estimation, and consistent merging across rooms. These insights directly inform the design of the automated systems developed in the later chapters of this thesis.

2.3 SALVe: Semantic Alignment Verification for Floorplan Reconstruction from Sparse Panoramas

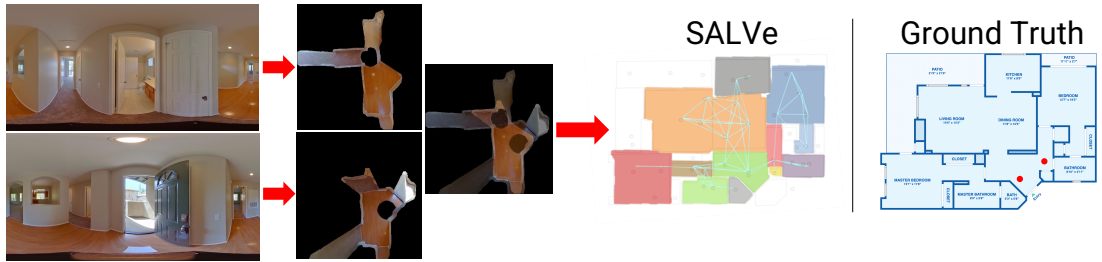


Figure 2.4: A challenging wide-baseline indoor example where traditional keypoint-based *SfM* fails, but *SALVe* succeeds by exploiting semantic features such as windows, doors, and openings (W/D/O). Pairwise pose hypotheses are generated from these cues and verified using top-down aligned renderings, producing more complete global pose estimation and improved floor-plan reconstruction than *OpenMVG* [42] and *OpenSfM* [23]. Here, *SALVe* validates a pose hypothesis grounded on a hallway opening.

2.3.1 Overview

SALVe [30] studies an early subproblem in sparse indoor reconstruction: how to recover coarse global poses for unordered, wide-baseline 360° panoramas when traditional feature-based *SfM* is too incomplete to support floor-plan reconstruction. This step is essential because multi-view floor-plan generation depends on first placing panoramas into a common coordinate system so that per-view layouts can be fused consistently, yet that initialization is especially difficult in residential interiors with weak texture, repeated structure, limited overlap, and abrupt room-to-room transitions. Before this work, *SfM* was

the standard baseline for multi-view reconstruction [54, 41, 68], but it often failed in sparse wide-baseline indoor settings. Semantic alternatives such as *ExtremeSfM* [57] had begun to appear, but were still limited, while learning-based 360° geometry prediction, including layout and depth cues, was only starting to emerge [65, 66, 17]. In this context, *SALVe* was the first work to make floor-plan reconstruction from unposed, wide-baseline panoramas practical by treating pose estimation as a discrete semantic alignment problem, showing that structural correspondences can replace unreliable low-level feature matching.

Each panorama is processed to predict room layout geometry together with windows, doors, and openings. These structural cues are used to generate discrete pairwise pose hypotheses, which are then verified by a learned alignment model using bird’s-eye-view renderings derived from predicted depth. The verified pairwise relations are aggregated into a globally consistent panorama arrangement, after which the per-image layouts are placed into a common frame and stitched into a rasterized floor plan.

2.3.2 Methods

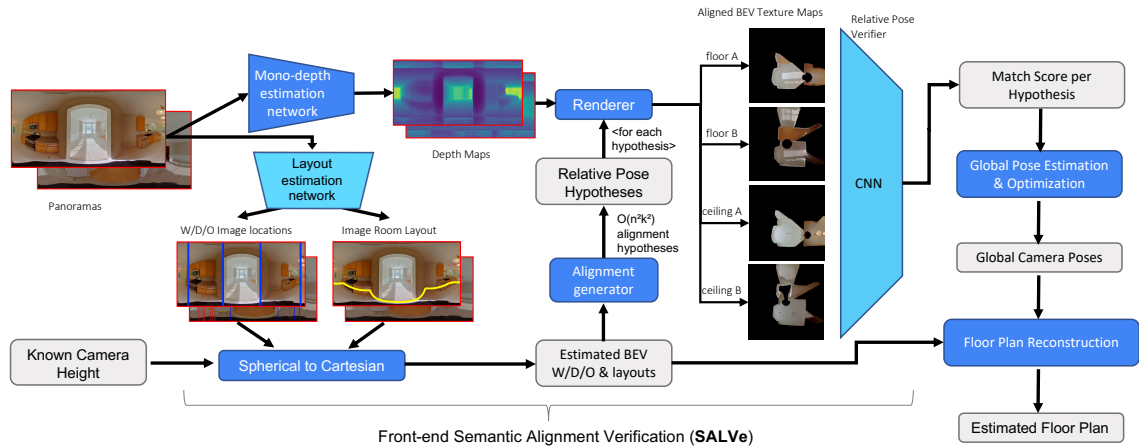


Figure 2.5: Overview of our floorplan reconstruction system. *BEV* = *bird’s eye view*. Blue boxes are processing components, gray boxes are data. Slanted blocks denote components based on deep networks; lighter blue networks are trained by us. ‘Image Room Layout’ the predicted floor boundaries (at each panorama column). n is the number of panoramas and k is the average number of detected windows/doors/openings per panorama. We show rendered floor and ceiling texture maps for a consistently-aligned pair of panoramas.

SALVe addresses the above challenge with a two-stage pipeline that first recovers globally consistent panorama poses and then uses those poses to assemble a floor plan. Given an unordered set of indoor 360° panoramas, each view is first processed to predict structural elements, including room layout boundaries together with windows, doors, and openings. These predicted cues are then used to form discrete pairwise relative pose candidates in $SE(2)$ by matching semantically compatible elements across panorama pairs. Each candidate alignment is evaluated by a learned verifier, and the verified pairwise relations are finally combined in a global pose graph, which is optimized with *GTSAM* [20] to

recover a consistent set of panorama poses. The full system assumes known camera height and upright orientation, and operates in a 2D *BEV* coordinate system.

At the pairwise level, *SALVe* uses a modified *HorizonNet* to predict room layout geometry together with windows, doors, and openings for each panorama. These structural cues generate discrete relative pose hypotheses by matching semantically compatible elements across views and refining their relative orientation using the dominant axes of the predicted layouts. To determine which hypotheses are plausible, *SALVe* renders floor and ceiling bird’s-eye-view maps from predicted depth under each candidate pose and scores the resulting overlap with a ConvNet-based semantic alignment verifier.

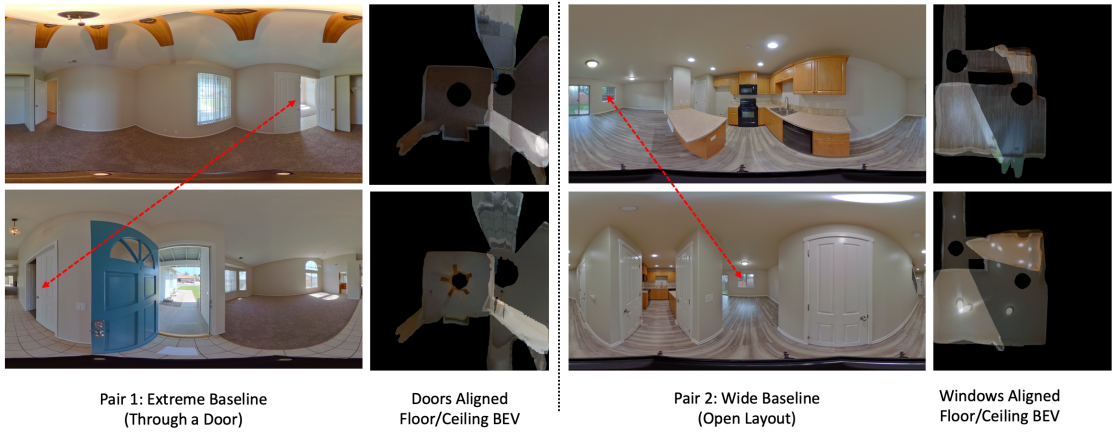


Figure 2.6: **Alignment-verifier input.** Orthographic BEV renderings of panorama pairs after semantic alignment proposal. Red arrows indicate matched windows, doors, or openings used to form pose hypotheses. Overlaid floor and ceiling maps are shown for an extreme-baseline pair and a wide-baseline pair.

The verified pairwise relations are assembled into a global panorama graph. Global poses are then recovered either by spanning-tree (*ST*) aggregation of pairwise transforms or by pose-graph optimization (*PGO*) with *GTSAM* [20], which refines pose accuracy by enforcing global consistency of rotations and translations over cycles in the graph. An optional axis-alignment (*AA*) step further rotates the reconstruction to the dominant indoor frame by aligning vanishing directions, following *PanoContext* [86], before floor-plan stitching. The recovered poses are finally used to place the predicted room layouts into a common frame, where they are grouped and confidence-weighted to form a stitched raster floor plan.

2.3.3 Results

Table 2.1 provides the main quantitative evaluation of *SALVe*. The learned semantic verifier, together with global pose aggregation, yields much higher localization completeness than traditional *SfM* baselines on sparse indoor panoramas, while preserving low rotation error and comparable translation accuracy. This is especially important because floor-plan reconstruction depends not only on local pose accuracy, but also on how many

panoramas can be placed into a single globally consistent frame. On this metric, *SALVe* substantially outperforms *OpenMVG* and *OpenSfM*, showing that semantic snapping and learned pairwise verification are much more effective than keypoint-based matching in the wide-baseline indoor regime. The same trend appears in downstream floor-plan reconstruction from Table 2.2: *SALVe* reaches 0.56 mean intersection over union (IoU) [89], compared with 0.29 for *OpenSfM* and 0.16 for *OpenMVG*, confirming that improved global pose recovery leads to substantially better stitched floor plans.

Table 2.1: **Results of global pose estimation on the ZInD test set.** We compare two global aggregation methods, spanning tree (*ST*) and pose graph optimization (*PGO*), with optional axis alignment (*AA*). Both *ST* and *PGO* operate on the same largest connected component of $\mathcal{G}$ and therefore localize the same number of panoramas.

METHOD	LOCALIZATION %		TOUR AVG. ROTATION ERROR (DEG.)		TOUR AVG. TRANSLATION ERROR (METERS)	
	MEAN	MEDIAN	MEAN	MEDIAN	MEAN	MEDIAN
<i>OpenSfM</i> [23]	27.62	22.22	9.52	0.36	1.88	0.12
<i>OpenMVG</i> [41, 42]	13.94	8.70	3.84	0.37	0.41	0.10
OURS (w/ <i>ST</i> + <i>AA</i>)	60.70	57.10	3.69	0.03	0.81	0.26
OURS (w/ <i>PGO</i> + <i>AA</i>)	60.70	57.10	3.73	0.17	0.80	0.25

Table 2.2: **Floor-plan reconstruction results on the ZInD test set.** Floor-plan 2D IoU is measured in bird’s-eye view against the manually annotated ground-truth floor plan, on the largest connected component. Rows differ in whether global poses and layouts are estimated or provided as oracle inputs. *AA* denotes axis alignment.

METHOD	GLOBAL POSES		LAYOUT		FLOORPLAN IOU	
	ORACLE	ESTIMATED	ORACLE	ESTIMATED	MEAN	MEDIAN
OPENSfM		✓	✓		0.29	0.26
OPENMVG		✓	✓		0.16	0.07
OURS(CONTROL)	✓			✓	0.94	0.95
OURS (PGO + AA) (CONTROL)		✓	✓		0.56	0.53
OURS (PGO + AA)		✓		✓	0.49	0.45

Figure 2.7 shows the same trend qualitatively. Relative to the baselines, *SALVe* produces pose graphs that cover larger portions of the home and therefore support more complete floor-plan reconstruction. Successful cases show that semantically verified alignments can connect panoramas across rooms and through bottlenecks where traditional *SfM* often fails. Failure cases suggest that the main limitation is less severe pose drift within a connected component than fragmentation of the global graph into disconnected pieces. Overall, these results show that the main strength of *SALVe* is as a reliable coarse initializer for downstream indoor reconstruction rather than as a final high-precision pose estimation system.



Figure 2.7: **Qualitative comparison of floor-plan reconstruction.** Columns show *OpenSfM*, *OpenMVG*, *SALVe*, and ground truth, all overlaid on the ground-truth floor plan. Colored regions indicate reconstructed coverage, showing that *SALVe* recovers more complete layouts. Colored lines denote **doors**, **openings**, and **windows**; cyan edges indicate verified W/D/O alignments.

2.3.4 Key insights on the ZInD pipeline

For the ZInD pipeline, *SALVe* provided a highly useful level of automation by supplying a strong first guess of panorama connectivity and coarse layout arrangement. Although the recovered poses were not accurate enough for final human-grade annotation, they substantially reduced the amount of manual image matching required. A key novelty was the shift from treating pose estimation as continuous feature matching to treating it as discrete semantic hypothesis generation and verification, which proved much more effective in sparse residential interiors. In practice, the system was most useful when deployed as pairwise connection signals rather than as a fully recovered floor plan: it gave annotators plausible room connections to refine, instead of requiring them to

establish panorama relations from scratch. This partial deployment also lowered the bar relative to full pose-graph recovery, which requires much higher precision and recall. More broadly, *SALVe* highlighted a modeling lesson used later in this thesis: ConvNet-based verification is effective for scoring discrete alignment hypotheses, but precise pose recovery benefits from additional geometric refinement, such as *PGO*, and eventually from stronger architectures beyond this formulation.

2.4 *PSMNet*: Position-aware Stereo Merging Network for Room Layout Estimation

2.4.1 Overview

Recovering room layout from images is an important step toward floor-plan generation and indoor scene understanding. While prior methods perform well for simple Manhattan or Atlanta-world rooms, they are much less reliable for large and complex interiors with many corners, merged functional areas, and severe self-occlusion. This is common in real homes, where a single panorama may capture only part of an open-plan space, leaving room boundaries incomplete or weakly observed.

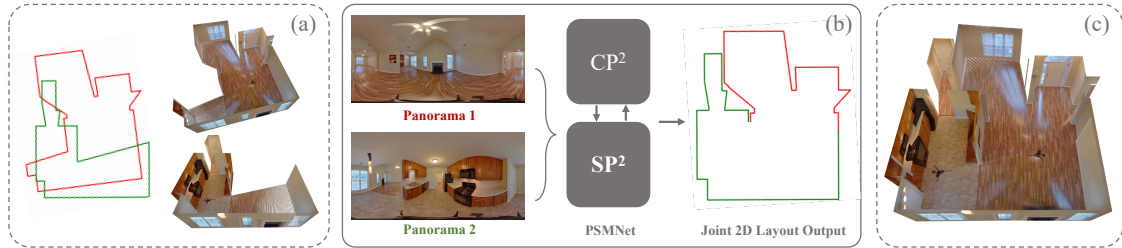


Figure 2.8: **Overview of *PSMNet*.** Two 360° panoramas capture complementary but partial observations of a complex room. Given a noisy relative pose, *PSMNet* jointly refines the pose and predicts the full visible 2D layout. Panels (a) and (c) are ground-truth visualizations; panel (b) shows the actual input and output of the system.

PSMNet [77] addresses this limitation by extending layout estimation from a single panorama to a pair of 360° panoramas. The key idea is that two views can provide complementary coverage of the same room, but only if their relative pose is estimated accurately enough for cross-view fusion. *PSMNet* therefore formulates layout prediction and relative pose refinement as a coupled two-view problem, where improved alignment leads to better layout recovery. Within this thesis, *PSMNet* serves as an early step from single-view room understanding toward multi-view floor-plan reconstruction.

2.4.2 Methods

PSMNet is built around a simple idea: two panoramas can recover a more complete room layout than either view alone, but only if they are first brought into a common geometric frame. The pipeline is therefore designed around joint pose refinement and layout fusion.

One panorama is treated as the anchor view, and the second is aligned to it under a 3-DOF relative pose consisting of planar translation and yaw rotation. Rather than fusing the two views directly in equirectangular image space, *PSMNet* first converts them into aligned top-down representations, where room structure can be compared more naturally across views.

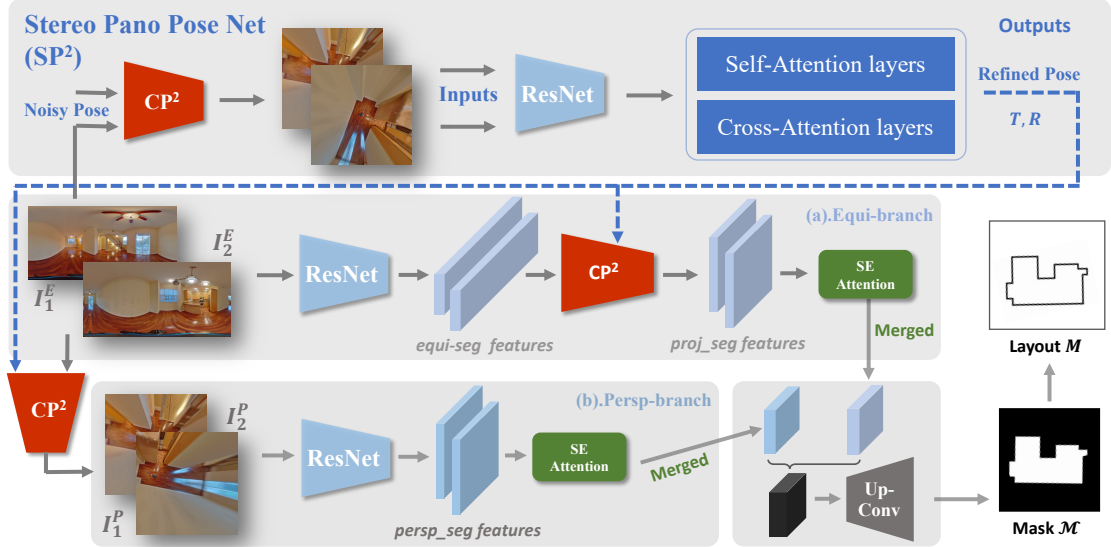


Figure 2.9: **Architecture of *PSMNet*.** The input is a pair of equirectangular panoramas with a noisy relative pose. The Cross-Perspective Projection layer (CP^2) projects them into aligned perspective views, while the Stereo Pano Pose network (SP^2Net) refines the relative pose. *PSMNet* predicts a room-layout mask $\mathcal{M}$, which is then post-processed into the final layout polygon M .

This alignment is performed by the Cross-Perspective Projection layer, denoted CP^2 , which projects both panoramas into a shared floor-coordinate system centered at the anchor view. Under the standard assumptions of upright capture and approximately equal camera height, the anchor view is projected directly, while the secondary view is projected using its current relative pose estimate. The result is a pair of top-down perspective views that make cross-view structure more comparable and provide a convenient representation for both pose refinement and downstream layout fusion.

Because the input relative pose is only approximately known, *PSMNet* includes a Stereo Pano Pose network, denoted SP^2Net , to refine it. SP^2Net takes the aligned top-down views, extracts multi-scale features with a shared backbone, and reasons jointly across the two views using transformer-based self- and cross-attention. It then predicts a residual correction to the input translation and rotation, rather than estimating pose from scratch. This refined pose is fed back into the projection stage so that the two panoramas can be better aligned before final layout prediction. The fused features are used to predict a 2D room-shape mask, which is then vectorized into a polygon by extracting its contour, fitting line segments, and applying an Atlanta post-processing step [55] that still allows non-Manhattan walls when sufficiently supported by the prediction.

2.4.3 Results

Quantitative results in Table 2.3 show that *PSMNet* achieves the best performance across all evaluation settings on the ZInD stereo-view benchmark. As an early 2021 multi-view layout method, *PSMNet* was evaluated against strong single-view baselines adapted to the stereo setting through cross-view merging, since prior direct multi-view layout baselines were not available. While this is not a perfectly matched comparison setting, it was the natural baseline setup at the time and still illustrates the value of multi-view reasoning. *PSMNet* improves over the strongest baseline by 4.62 points in 2D IoU when ground-truth pose is given, and the improvement grows further in the more difficult Overlap-Medium and Overlap-Low splits. With noisy input pose, *PSMNet* maintains a clear margin over all baselines, showing that joint pose refinement and layout estimation improve robustness in more realistic settings. In particular, *PSMNet* under noisy pose remains competitive with, and often better than, baseline methods given ground-truth pose.

Table 2.3: **Quantitative evaluation stratified by spatial overlap.** Results are grouped by overlap level, where *Overlap-Low* corresponds to higher room complexity with stronger occlusion.

Pose	Methods	Overall		Overlap-High		Overlap-Medium		Overlap-Low	
		2D IoU (%)	δ_i	2D IoU (%)	δ_i	2D IoU (%)	δ_i	2D IoU (%)	δ_i
w/ GT	<i>DulaNet</i> [83]	64.03	0.8043	65.21	0.8185	62.14	0.8031	60.27	0.7980
	<i>HorizonNet</i> [65]	73.35	0.8663	82.08	0.8801	71.47	0.8678	69.20	0.8535
	<i>HoHoNet</i> [66]	74.25	0.8649	82.55	0.8816	72.43	0.8672	70.35	0.8486
	<i>LED²Net</i> [76]	76.39	0.9056	83.68	0.9243	73.73	0.8736	72.08	0.8697
	<i>PSMNet</i> (Ours)	81.01	0.9238	85.71	0.9349	80.13	0.9253	76.93	0.9074
w/o GT	<i>DulaNet</i> [83]	59.30	0.7828	62.06	0.7699	57.97	0.7855	51.21	0.7936
	<i>HorizonNet</i> [65]	62.79	0.8354	70.98	0.8437	61.51	0.8355	58.24	0.8288
	<i>HoHoNet</i> [66]	63.31	0.8324	70.59	0.8390	62.03	0.8339	59.47	0.8253
	<i>LED²Net</i> [76]	65.81	0.8566	71.06	0.8493	64.81	0.8574	63.14	0.8611
	<i>PSMNet</i> (Ours)	75.77	0.9217	84.80	0.9371	74.73	0.9210	66.73	0.9040

Qualitative examples in Figure 2.10 show the same overall trend. Compared with the single-view baseline *LED²Net*, *PSMNet* recovers more complete room structure across views, especially in large spaces with strong occlusion and partial visibility. The predicted layouts are generally more coherent in top-down view and better aligned with the underlying room extent, confirming that multi-view fusion can improve room layout estimation beyond what is visible from any single panorama. These examples also make clear that 2D IoU, although a standard metric in this period, is a limited measure of structured layout quality: it captures coarse floor-area agreement, but can miss important geometric failures such as wrong wall topology, inaccurate corner placement, or oversimplified structure. For this reason, the qualitative results are especially important for understanding the strengths and limitations of the method.

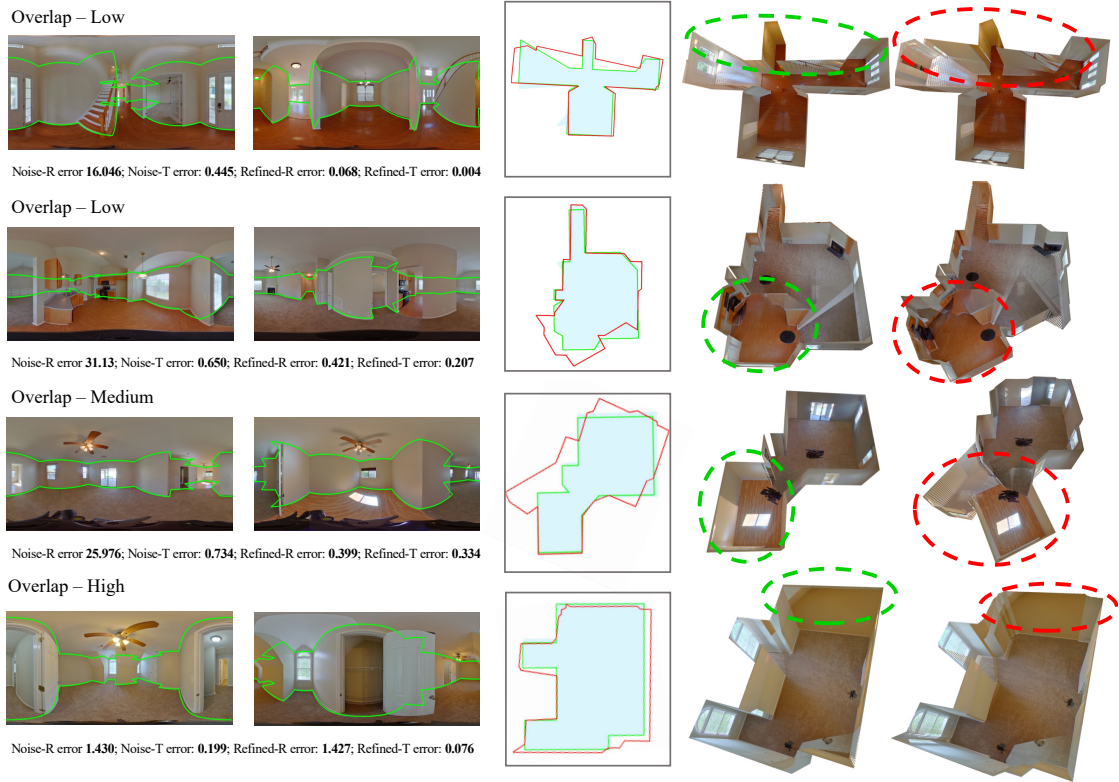


Figure 2.10: **Qualitative results on ZInD with noisy pose.** The first two columns show the estimated visible layout on each panorama in green. The middle plot overlays *PSMNet* in green, ground truth in cyan, and *LED²Net* in red. The last two columns show the corresponding 3D layouts.

2.4.4 Key insights from the *PSMNet*

PSMNet highlighted several important lessons that directly motivate later multi-view work in this thesis. First, multi-view reasoning is often necessary for large and complex rooms, where a single panorama may observe only a partial layout. Second, pose and layout are tightly coupled: the final fused layout depends strongly on relative pose accuracy, and even small alignment errors can produce visible artifacts in the recovered room shape. In this sense, pose is not merely an auxiliary variable, but a core factor that determines whether cross-view layout evidence can be combined correctly.

At the same time, *PSMNet* exposed the limitations of this early formulation. Pose refinement from image features remained difficult to make precise, especially with a largely convolutional residual-correction design. Even when pose error was modest, the subsequent reprojection and fusion process had limited ability to correct misalignment, since it relied mostly on geometric overlap rather than richer wall-level semantic cues. More fundamentally, the representation lacked the right intermediate semantics: the layout was first predicted as a fused segmentation mask and only later converted into a polygon, without an explicit wall-level representation linking image evidence, cross-view correspondence, and final structure. As a result, the vectorization stage was brittle, and noise in

the fused floor mask could cause the fitted layout to either oversimplify the room structure or overestimate the number and placement of walls.

Taken together, these observations pointed to the need for a stronger multi-view network in which pose estimation, cross-view fusion, and layout prediction are more tightly integrated. In particular, layout vectorization should be informed by wall-level correspondences and image semantics, while layout fusion should also exploit structural cues to remain robust to residual pose error. These lessons directly motivate the development of *MvFFN* later in this thesis.

2.5 Conclusion

The results from *PSMNet* and *SALVe* reveal complementary aspects of the same underlying problem. *PSMNet* shows that joint multi-image layout estimation is important for complex room layouts, especially when a single panorama provides incomplete or ambiguous structural evidence. However, its dependence on accurate relative pose limits its robustness in sparse wide-baseline settings. Conversely, *SALVe* and the *ZInD* pipeline show that reliable camera pose estimation benefits strongly from accurate room layout and semantic structure. Together, these observations suggest that pose and layout should not be treated as independent stages: robust reconstruction requires cross-view reasoning that couples geometry, camera pose, and room structure. Chapter 3 addresses this need by reformulating floor-plan reconstruction around cross-view wall instances and within-room connectivity, enabling pose and layout to be estimated within a more unified multi-view framework.

Chapter 3

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

Before introducing *MvFFN*, this chapter briefly revisits *CoVisPose*, an earlier method for wide-baseline indoor panorama pose estimation that informed several ideas developed more fully here. *CoVisPose* showed that robust relative pose estimation can benefit from moving beyond sparse feature matching and instead learning dense column-wise representations of co-visibility, angular correspondence, and dense layout geometry over the full 360° field of view. This established an important lesson for the present chapter: structured geometric intermediates make wide-baseline indoor reasoning more reliable under low overlap, repeated structure, and weak texture. Building on this insight, *MvFFN* extends the problem from pairwise pose estimation to unified multi-view prediction of pose, dense geometry, and structured floor layout. To make this difficult setting more tractable, it reformulates floor-plan reconstruction through *Multi-view Ordered Wall Instance Segmentation* (MOIS), replacing direct global wall-sequence prediction with cross-view wall instances and within-room connectivity, which are then assembled into globally consistent multi-room layouts.

3.1 *CoVisPose*: An Intermediate Step Toward *MvFFN*

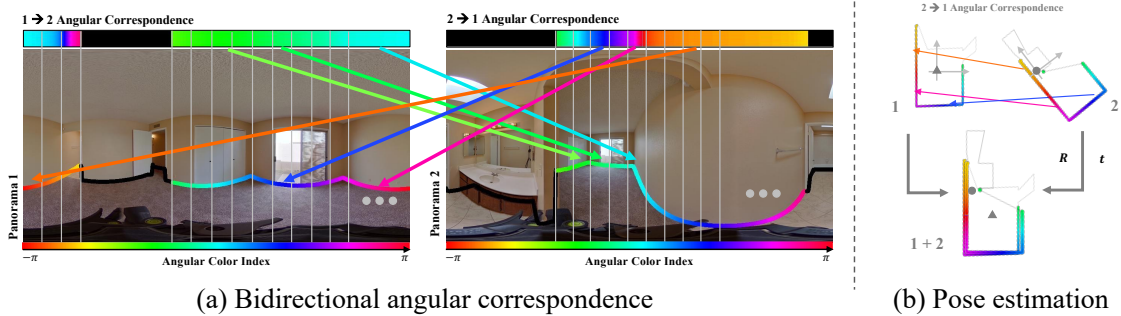


Figure 3.1: *CoVisPose* prediction targets: dense per-column correspondence, co-visibility, and floor-boundary geometry, which together support relative camera pose estimation.

Overview. *CoVisPose* [27] marks an important step beyond *SALVe* in wide-baseline indoor panorama reconstruction. Instead of treating pose estimation mainly as semantic verification over discrete hypotheses, it reformulates the problem through dense, pixel-grounded geometric prediction, decomposing relative pose estimation into local cross-view reasoning tasks that can be learned more robustly. This shift, conceptually similar to later dense matching approaches such as *LoFTR* [67, 54], improves both robustness and spatial

precision under limited overlap, occlusion, and repeated indoor structure. More broadly, it moved the pipeline from classification-style verification toward direct pose recovery, bringing it one step closer to end-to-end multi-view layout estimation.

Architecture. *CoVisPose* implements this idea by encoding each panorama with a shared *ResNet-50* backbone, converting the resulting features into column-wise tokens, and jointly reasoning over both views with a self-attention transformer. A lightweight decoder predicts dense geometric quantities for each column, including co-visibility, angular correspondence, and floor-wall boundary geometry, while a separate pose head regresses the relative camera pose. The same dense predictions can also support *RANSAC*-based pose recovery at inference time. The model is trained end-to-end with supervision on both the dense geometric outputs and the final pose: ℓ_1 losses are applied to co-visibility, angular correspondence, and boundary prediction, while an ℓ_2 regression loss is used for the relative pose parameters.

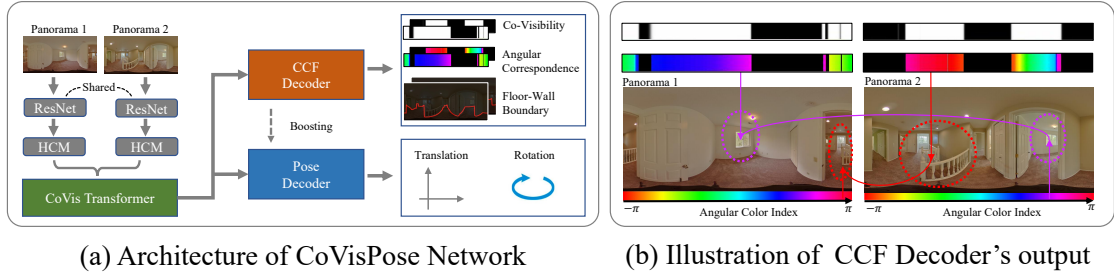


Figure 3.2: ***CoVisPose* architecture.** (a) The architecture consists of *CoVis* transformer, the Co-visibility, angular Correspondence and Floor-wall boundary (CCF) decoder, and pose decoder. (b) An example to illustrate the CCF decoder's outputs.

A key insight from *CoVisPose* is that dense surrogate tasks, combined with attention-based cross-view reasoning, can substantially improve the robustness and precision of wide-baseline indoor pose regression, even when panorama pairs share extremely low co-visibility, as illustrated in Figure 3.3. In the original study, *CoVisPose* was shown to outperform both classical *SfM* baselines and dense matching approaches such as *LoFTR*[67] on ZInD, especially under low overlap. Rather than predicting pose directly from weak global image evidence, the model learns structured intermediate geometry that is easier to supervise and more stable under occlusion and repeated indoor structure. More importantly for this thesis, the learned attention mechanism points beyond pairwise pose estimation: by jointly reasoning over floor-boundary geometry and cross-view relations, it suggests that the same representation could support richer multi-view layout reasoning. That potential is only implicit in *CoVisPose*, but it becomes a central idea behind *MvFFN*.



Figure 3.3: **Qualitative results** on ZInD. (a) *SIFT* feature point inliers generated by 2-view *SfM* with *OpenMVG* [42], (b) *LoFTR* [67] feature point inliers generated by the same solver as (a), (c) *CoVisPose* prediction, (d) predicted floor-wall boundaries for Panorama 1 and Panorama 2 aligned by *CoVisPose* direct regression. (e) Pose errors for each method. Color scheme corresponds to maximum of rotation and translation errors: Failure to recover pose or error $> 10^\circ$, error in $[2.5^\circ, 10^\circ]$, error $< 2.5^\circ$.

We now turn to *MvFFN*, which is the focus of the remainder of this chapter.

3.2 Introduction

MvFFN targets the fully automatic sparse wide-baseline setting: given unposed panoramas, the model must recover camera poses, establish cross-view wall identity, and assemble connected multi-room layouts in a shared frame from incomplete observations. Unlike earlier pairwise pose methods such as *CoVisPose*, *MvFFN* is designed as an N -view feed-forward reconstruction model that reasons over multiple panoramas jointly.

A floor plan encodes not only wall geometry, but also local connectivity and room-level organization. Recovering this structure from sparse views requires associating wall instances across panoramas, inferring their connectivity despite occlusion, and separating structures that belong to different rooms. Geometry alone is often insufficient under pose uncertainty: noisy fused point maps can lead to duplicate walls, missing walls, and incorrect junctions. *MvFFN* therefore combines geometric prediction with image-based semantic reasoning, using cross-view wall instances and within-room connectivity as intermediate structure rather than predicting the entire global floor plan directly. Existing methods typically simplify one or more parts of this setting: they assume known camera poses [85, 3, 38], require dense or small-baseline captures [3], or rely on human-in-the-loop adjustment [17].

Directly predicting a complete multi-room floor-plan as a single global output is also difficult to learn. Large layouts contain ambiguous room partitions, limited observations can obscure structural details, and small local sequence errors can cascade into global drift. The problem is further compounded by scarce datasets with structured multi-room layouts and camera-registered multi-view panoramas. This motivates the central insight of our paper: rather than predicting a global floor-plan directly, we formulate reconstruction as dense local, anchor-conditioned prediction of cross-view wall identity and within-room connectivity, which is 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 *Multi-view floor-plan Feed-Forward Network (MvFFN)*, which predicts pixel-grounded wall instances and within-room connectivity from cross-view geometric and semantic cues. A deterministic *Layout Chaining (LC)* procedure then 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 correspondences can support downstream layout refinement.

Together, these design choices make multi-view layout estimation more learnable under scarce data and noisy supervision.

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, and 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 the ZInD-CrossView (ZInD-XV) dataset with globally unique wall instances and cross-view segmentation annotations.

3.3 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. In the single-view setting, methods predict monocular 1D floor boundaries [83, 76, 28, 66] and recover wall and corner sequences through post-processing [89, 65, 51], 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. Under imperfect geometry, connectivity is typically recovered with heuristic optimization [6, 77] or learned polygon and graph decoders [38, 85, 8, 64]. By contrast, cross-view semantic cues remain underused for resolving wall identity and connectivity under pose uncertainty. *SceneScript* [3] 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* [56, 43, 30]. Recent learning-based models improve dense 3D point map and camera pose estimation from wide baseline inputs using strong foundation model encoders [80, 79, 72]. For indoor 360° panoramas, limited training data often constrains geometry estimation to 1D floor boundary representations rather than dense per-pixel 3D geometry [27, 73, 47]. *BADGR* [36] 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.

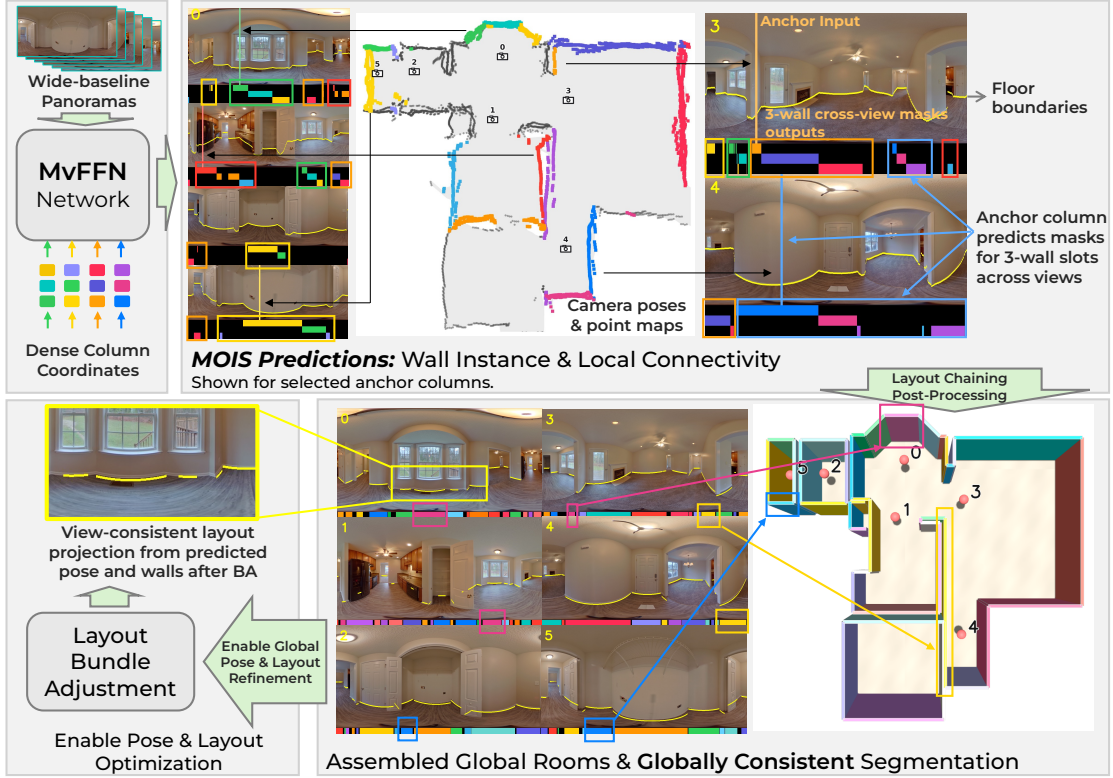


Figure 3.4: Overview of *MvFFN*, a cross-modality transformer for estimating consecutive wall instances from multi-view per-column segmentation and dense global geometry. Predicted wall instances and local within-room connectivity are assembled into multi-room global floor layouts. Our *MOIS* formulation also enables fully automatic downstream pose and layout refinement.

3.4 Overview

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 (Fig. 3.4) 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* [79], 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. 3.6 and Sec. 3.7, respectively. Finally, the dense local *MOIS* predictions are assembled into global room layouts via *Layout Chaining* (Sec. 3.10).

3.5 Method

We formulate multi-room layout reconstruction from sparse panoramas [86] 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 $\{I_i\}_{i=1}^N$ [86], 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 [17, 27, 36]. 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. 3.4 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

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

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 $\mathcal{S}^{\text{gt}}$, 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. 3.7 and is assembled into global layouts by *Layout Chaining* in Sec. 3.10.

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 $\mathcal{M} : \mathcal{C}_{i,c} \rightarrow \mathcal{W}^\ell$, which associates each image column with at most one global wall instance. The room corners $V^m = (v^{m,1}, \dots, v^{m,K_m})$ 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.

3.6 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 [79], while operating on panoramic column-wise features. Each panorama I_i is processed by an *EfficientNet-v2* backbone [69], followed by a Height Compression Module (HCM) [65] 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) [75] to obtain geometry tokens F^G . The ACE are *Fourier features* [70] 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 [86]. Incorporating vanishing-angle embeddings exploits the high precision of indoor vanishing angles [36, 30] to axis-align the predicted floor layout. The global floor coordinate frame is centered at the first camera and oriented by its vanishing angle.

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 $t_i^R \in \mathbb{R}^{4 \times C'}$ [19], and process them with a 12-layer alternating-attention transformer [79]. The transformer alternates between frame-wise and global attention layers implemented with *FlashAttention* [18], 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 stochastic uncertainties. Each supporting line is represented by a 2D orientation vector and a bias term. Floor boundaries and point maps are predicted with stochastic uncertainty [29]. Following prior panorama layout work [65, 27, 36], 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 [79] using an MLP head with adaptive LayerNorm [49].

3.7 Floor Layout Semantic Branch

Our floor-layout semantic branch predicts the dense *MOIS* probability field introduced in Sec. 3.5. Specifically, for each anchor column, it estimates which columns across all views

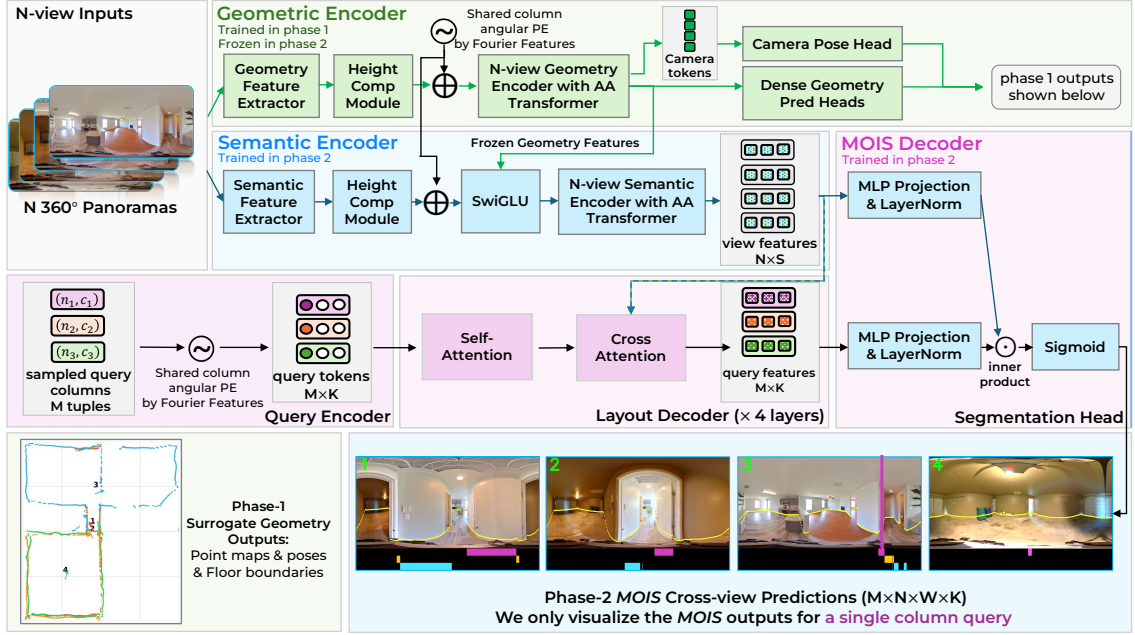


Figure 3.5: Architecture of *MvFFN* with two-stage training for *MOIS* prediction.

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. 3.10), while avoiding the degradation of longer prediction windows (see Sec. 3.16.3). 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. 3.6. 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 $r \in \{0, \dots, K - 1\}$. 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 $\hat{t}_{m,r}^q$, one for each anchor m and wall slot r .

MOIS probability prediction. Let $\tilde{t}_{m,r}^q \in \mathbb{R}^{C'}$ and $\tilde{t}_{j,d}^l \in \mathbb{R}^{C'}$ 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

$$\hat{\mathcal{S}}_{m,j,d,r} = \sigma\left(\left\langle \tilde{t}_{m,r}^q, \tilde{t}_{j,d}^l \right\rangle\right), \quad (1)$$

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.

3.8 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.

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

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

Here, $\mathcal{L}_{\mathcal{P}}$ and $\mathcal{L}_{\mathcal{B}}$ denote the point-map and floor boundary losses. Following *DUST3R* [80], both use stochastic uncertainty weighting [29] with predicted uncertainties $\Sigma_i^{\mathcal{P}}$ and $\Sigma_i^{\mathcal{B}}$. The pose losses $\mathcal{L}_{rel}$ and $\mathcal{L}_{abs}$ follow *CovisPose* [27]; valid pairs for $\mathcal{L}_{rel}$ have minimum co-visibility 0.01. Specifically,

$$\begin{aligned} \mathcal{L}_{\mathcal{P}} &= \sum_{i=1}^N \left\| \Sigma_i^{\mathcal{P}} (\mathcal{P}_i - \hat{\mathcal{P}}_i) \right\| - \alpha \log \Sigma_i^{\mathcal{P}}, \\ \mathcal{L}_{\mathcal{B}} &= \sum_{i=1}^N \left\| \Sigma_i^{\mathcal{B}} (\mathcal{B}_i - \hat{\mathcal{B}}_i) \right\| - \alpha \log \Sigma_i^{\mathcal{B}}, \\ \mathcal{L}_{abs} &= \sum_{i=1}^N (\|t_i - \hat{t}_i\| + \beta_r \|r_i - \hat{r}_i\|). \end{aligned}$$

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 $m \in \{1, \dots, M\}$ 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 $\hat{\mathcal{S}}_{m,j,d,r} = \sigma(z_{m,j,d,r}^{\text{cls}})$, where $r \in \{0, \dots, K-1\}$ indexes the ordered wall slot relative to the anchor. The binary supervision is $\mathcal{S}_{m,j,d,r}^{\text{gt}} \in \{0, 1\}$.

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

$$\begin{aligned}\mathcal{L}_{\text{pres}} &= \frac{1}{|\Omega_{\text{valid}}|} \sum_{(m,j,d) \in \Omega_{\text{valid}}} \text{BinaryCrossEntropy}\left(z_{m,j,d}^{\text{pres}}, y_{m,j,d}^{\text{pres}}\right), \\ \mathcal{L}_{\text{cls}} &= \frac{1}{|\Omega_{\text{pos}}|} \sum_{(m,j,d) \in \Omega_{\text{pos}}} \text{CrossEntropy}\left(z_{m,j,d,:}^{\text{cls}}, y_{m,j,d}^{\text{cls}}\right), \\ \mathcal{L}_{\text{MOIS}} &= \mathcal{L}_{\text{pres}} + \lambda_{\text{cls}} \mathcal{L}_{\text{cls}}.\end{aligned}$$

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 $y_{m,j,d}^{\text{cls}} \in \{0, \dots, K-1\}$.

3.9 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* [79] and *MV-DUST3R* [72], which require substantially more training data than *DUST3R* [80] to improve 2-view pose accuracy. To improve convergence under this regime, we introduce two inductive biases based on reliable 2-view *CovisPose* [27] predictions.

Camera rotation priors. Two-view geometric models already provide reliable camera rotations, and per-panorama vanishing points further improve indoor rotation accuracy [86, 30]. We first run the 2-view *CovisPose* model and recover global camera rotations with a minimum spanning tree, following *BADGR*’s initialization strategy [36]. These refined rotations are then injected as conditioning signals into the angular encoder (Sec. 3.6). 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.

3.10 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 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 */
1 for each view  $n$  do
2   Define per-view current wall segmentation:  $\hat{\mathcal{S}}_n^{\text{view}} = \hat{\mathcal{S}}[n, :, n, :, 0]$ ;
3   Define per-view column nodes:  $\mathcal{V}_n^{\text{view}} = \{1, \dots, W\}$ ;
4   for column pair  $(c_1, c_2) \in \mathcal{V}_n^{\text{view}} \times \mathcal{V}_n^{\text{view}}$  do
5     Define same-wall column edge: add  $(c_1, c_2)$  to  $\mathcal{E}_n^{\text{view}}$  if
      IoU( $\hat{\mathcal{S}}_n^{\text{view}}[c_1], \hat{\mathcal{S}}_n^{\text{view}}[c_2]$ )  $> \tau_{\text{iou}}$ ;
6   Build per-view same-wall column graph:  $\mathcal{G}_n^{\text{view}} = (\mathcal{V}_n^{\text{view}}, \mathcal{E}_n^{\text{view}})$ ;
7   Collect per-view wall instances:  $L_n = \text{ConnectedComponents}(\mathcal{G}_n^{\text{view}})$ ;
8   for each connected component  $l \in L_n$  do
9     Select anchor column:  $c^* = \arg \max_{c \in l} \deg(c)$ ;
10    Define instance mask:  $l^* = \hat{\mathcal{S}}_n^{\text{view}}[c^*]$ ;

/* Stage 2: Cross-view global wall instances */
11 Define cross-view wall nodes:  $\mathcal{V}_{\text{cross}} = \bigcup_{n=1}^N L_n$ ;
12 for instance pair  $(l_i, l_j) \in L_{n_1} \times L_{n_2}$ ,  $n_1 \neq n_2$  do
13   Define cross-view same-wall instance edge: add  $(l_i, l_j)$  to  $\mathcal{E}_{\text{cross}}$  if IoU( $l_i^*, l_j^*$ )  $> \tau_{\text{iou}}$ ;
14 Build cross-view same-wall instance graph:  $\mathcal{G}_{\text{cross}} = (\mathcal{V}_{\text{cross}}, \mathcal{E}_{\text{cross}})$ ;
15 Collect global wall instances via cross-view column-to-wall assignments:
     $\mathcal{M} = \text{ConnectedComponents}(\mathcal{G}_{\text{cross}})$ ;

/* Stage 3: Room connectivity and partial layouts */
16 Define room-wall nodes:  $\{w^\ell\}$ , induced by  $\mathcal{M}$ ;
17 Define directed room-wall edges:
     $\mathcal{E}_{\text{room}} = \{(w_i, w_j) \mid \text{the } k \rightarrow 0 \text{ predictions in } \hat{\mathcal{S}}[:, :, :, 1 : K-1] \text{ support } w_i \rightarrow w_j\}$ ;
18 Build directed room graph:  $\mathcal{G}_{\text{room}} = (\{w^\ell\}, \mathcal{E}_{\text{room}})$ ;
19 Filter invalid edges by geometry:  $\mathcal{G}_{\text{room}} = \text{GeomFilter}(\mathcal{G}_{\text{room}})$ ;
20 for each weakly connected component  $\mathcal{H}^m \subseteq \mathcal{G}_{\text{room}}$  do
21   Collect within-room connectivity:  $\mathcal{L}^m = \text{LongestPath}(\mathcal{H}^m)$ ;

/* Stage 4: Final wall geometry */
22 Define global wall parameters:  $\{(\hat{b}^\ell, \hat{v}^\ell)\} = \text{LineFit}(\{\hat{\mathcal{P}}_{i,c}\}, \mathcal{M})$ ;
23 for each partial room layout  $\mathcal{L}^m$  do
24   Define room endpoints:  $V^m = (v^{m,1}, \dots, v^{m,K_m}) = \text{LineIntersect}(\mathcal{L}^m, \mathcal{M})$ ;

```

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 $\{V^m\}$.

3.11 Experiments

We evaluate primarily on ZInD-XV, which extends ZInD [17] 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-CrossView (HM3D-XV), a smaller auxiliary benchmark derived from residential HM3DSem scenes [82], 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 Sec. 3.19.

3.12 Experiments on ZInD-XV Dataset

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 [36]. To isolate the learned representation before post-processing, Sec. 3.12.3 evaluates raw *MOIS* prediction quality, Sec. 3.13 reports HM3D-XV transfer results, and Appendix A.2 details baseline implementations.

3.12.1 Pose Estimation Evaluation

We compare *MvFFN* with *CovisPose+Greedy Spanning Tree (GST)* [47, 36] and *VGGT* [79] on ZInD-XV, adapting *VGGT* to panoramas via perspective decomposition. We exclude *GraphCovis* [47] and *PanoPose* [73] 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 [47]; tuple construction and implementation details are provided in Appendix A.2. Although pose estimation is not the main goal of *MvFFN*, Table 3.1 shows that it is comparable to *CovisPose+GST*: the latter has lower median errors, while *MvFFN* lowers the mean, variance, and hard-case failures. *VGGT* performs much worse in this sparse wide-baseline panorama setting.

3.12.2 Floor Layout Estimation Evaluation

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 3.2, each baseline composes pose estimation, geometric input construction, and layout vectorization: *CovisPose+GST* (CP) or ground-truth poses provide the

Table 3.1: Camera pose evaluation on the 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
<i>CovisPose+GST</i>	12.6	23.3	9.3	1.3	0.9	0.7	17.3	37.3	11.8	1.9	1.6	0.8
<i>VGGT</i>	30.7	39.4	13.3	11.1	16.0	4.0	53.4	63.1	27.8	13.3	17.7	4.9
<i>MVFFN</i>	12.5	10.8	10.2	1.2	0.6	1.1	17.0	15.6	13.8	1.3	0.8	1.2

pose input, *LGT-Net* (*LGT*) provides panorama-level boundary cues, and the final layout is produced by either heuristic *LGT* post-processing (*LGT-PP*) or learning-based top-down vectorization with *RoomFormer* (*RF*) [85] and *PolyRoom* (*PR*) [38]. We exclude *VGGT* from layout evaluation because its pose accuracy is substantially worse than *CovisPose* poses.

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.

Table 3.2: Floor-layout evaluation on ZInD-XV. Distance is in cm. Geom. denotes the intermediate geometric input and Layout denotes the vectorization backend. GT-pose denotes a control group.

Modular 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↓
<i>MvFFN</i>			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
<i>CP</i>		<i>RF</i>	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
<i>CP</i>		<i>PR</i>	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
<i>CP</i>	<i>LGT</i>	<i>LGT-PP</i>	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
GT	<i>LGT</i>	<i>LGT-PP</i>	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

We evaluate wall and junction precision, recall, and *F1* with instance-level matching, and report matched-wall localization error; details are in Sec. 3.12.2. 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 [17]. As shown in Table 3.2, *MvFFN* substantially outperforms all modular baselines across both 5-view and 8-view settings, with the largest gains over learned top-down vectorization backends and consistent improvements over the strongest *LGT-PP* baseline. Notably, *MvFFN* also outperforms *LGT-PP* with ground-truth poses, indicating that the gains are not explained merely by better pose estimation. Instead, the wall and junction improvements show that cross-view *MOIS* prediction and *Layout Chaining* recover wall identities, within-room connectivity, and global layout structure more reliably than modular geometry-first pipelines.

3.12.3 *MOIS* Prediction Window

***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 con-

ditioning panoramas and retain anchors with valid ordered slot targets. Each anchor instantiates the cross-view slot masks defined in Sec. 3.5 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* [28] and *CovisPose* [27]. 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 3.3: *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.

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

Table 3.3 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.

3.13 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 [82] 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. 3.19.2. 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. 3.14. Sec. 3.14 evaluates pose estimation, and Sec. 3.15 evaluates floor layout reconstruction.

3.14 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 fine-tune the geometry branch on HM3D-XV using the same losses, view-count schedule, and dataloader setup as Sec. 3.8, 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.

As shown in Table 3.4, 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.

Table 3.4: 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
<i>VGGT</i>	35.2	109.3	10.4	6.1	16.3	2.1	62.3	71.8	37.4	13.7	6.4	12.3
<i>CovisPose+GST</i>	30.1	24.7	19.3	2.2	3.3	1.1	44.0	59.7	25.9	2.9	4.4	1.4
<i>MvFFN</i>	26.8	21.2	21.5	1.8	0.6	1.6	32.0	23.5	26.6	1.9	0.5	1.7

These results suggest that the *MvFFN* geometry branch, under this setting, can support stable multi-view pose estimation in the furnished HM3D-XV domain.

3.15 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. 3.14. 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. 3.8. We compare against *LGT-Net* based modular baselines using either *CovisPose* or ground truth camera poses. As in Sec. 3.12.2 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 3.5: Floor-layout evaluation on 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
<i>CP</i>	<i>LGT</i>	<i>LGT</i>	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
<i>GT</i>	<i>LGT</i>	<i>LGT</i>	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
<i>MvFFN</i>			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 3.5, 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.

3.16 Ablation Study

We evaluate three complementary ablations. The main paper studies **architecture**, testing whether the two-branch, two-stage *MvFFN* design is necessary for stable *MOIS* prediction. Sec. 3.16.2 studies **inductive bias**, evaluating rotation and correspondence priors across input-view counts, and Sec. 3.16.3 studies **prediction window**, where longer *MOIS* wall sequences become harder to learn.

3.16.1 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 3.6 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 3.6: Ablation study of *MvFFN* on ZInD-XV, comparing encoder design, geometry fusion, and staged training. – denotes pose metrics identical to *MvFFN*.

Method	Geom	Sem	Geom	Train	5v pos (cm)↓			5v rot (°)↓			5v wall				5v junction			
	Enc	Enc	Fusion	Stages	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
<i>MvFFN</i>	✓	✓	✓	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

3.16.2 Inductive Bias

We evaluate *MvFFN* across varying numbers of input panoramas, using three levels of inductive bias described in Sec. 3.9 of the main paper: no inductive bias, rotation priors only, and rotation priors with correspondence-based attention masks. As shown in Fig. 3.6, 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.

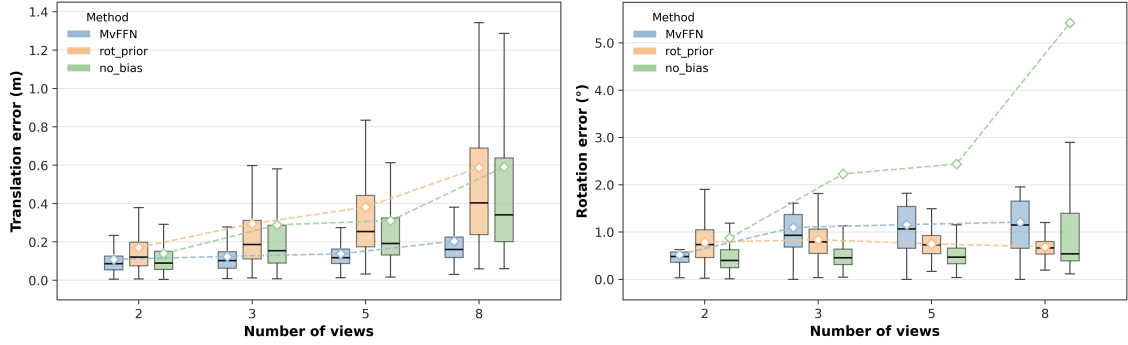


Figure 3.6: Effect of inductive bias across input views. Box plots show translation (left) and rotation (right) errors under different inductive-bias variants. Dashed lines and diamond markers indicate mean errors. Stronger inductive bias improves translation accuracy and suppresses extreme rotation outliers.

3.16.3 MOIS Prediction Window

We train two additional *MvFFN* variants with larger *MOIS* prediction windows of 5 and 7 ordered walls, and report the results in Table 3.7. For a controlled comparison, *Layout Chaining* is kept unchanged and uses only the first three predicted wall slots for all variants, so the ablation isolates the effect of training *MvFFN* to predict longer *MOIS* windows. Expanding the prediction window leads to a consistent drop in layout accuracy, with 5-view wall *F1* decreasing from 88.1 to 85.6 and 84.4, and 8-view junction *F1* decreasing from 75.0 to 71.3 and 67.1 for the 5-wall and 7-wall variants, respectively. Matched-wall distance also worsens in the 8-view setting. These results support our choice of a 3-wall *MOIS* window: a compact local prediction target is easier to learn under sparse wide-baseline observations, while longer ordered wall sequences accumulate more ambiguity from occlusion, room-boundary uncertainty, and distant structural details. The degradation is gradual rather than catastrophic: increasing the *MOIS* window makes the task harder, but does not break the formulation. This suggests that locality is helpful, with 3 walls giving the best operating point in our setting, while the broader advantage comes from replacing direct global prediction with an anchor-conditioned local representation.

Table 3.7: *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

3.17 Qualitative Results

We show qualitative results from both *ZInD-XV* and *HM3D-XV* datasets with 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.

3.17.1 ZInD-XV Results

Figs. 3.7 and 3.8 show 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.

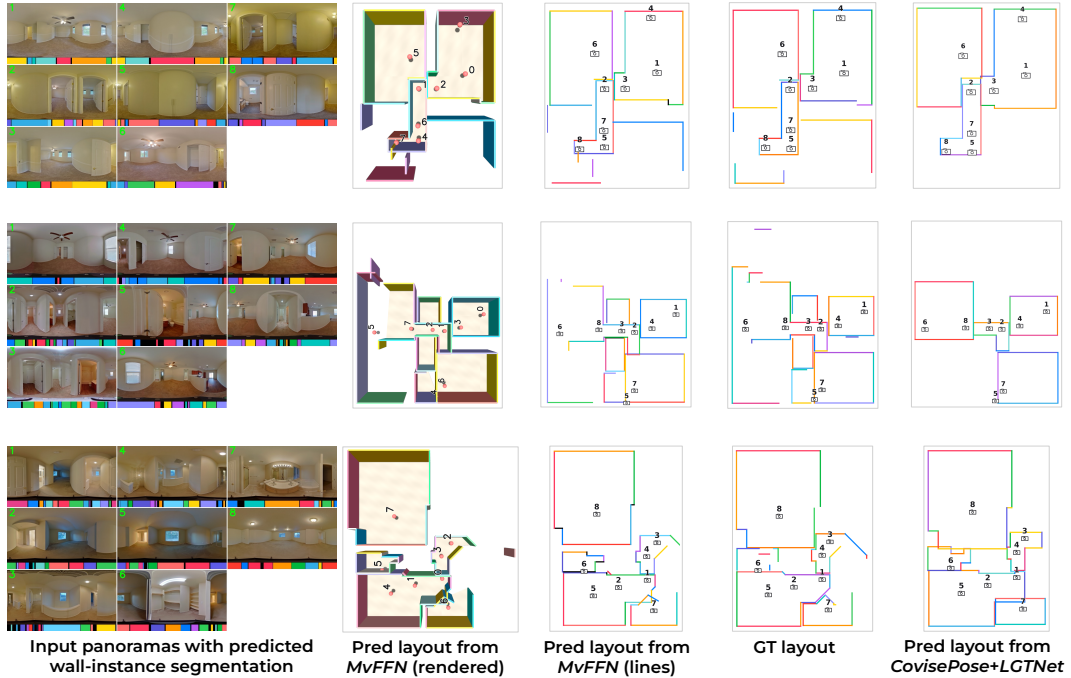


Figure 3.7: Qualitative comparisons on ZInD-XV test set.



Figure 3.8: More qualitative comparisons on ZInD-XV test set.



Figure 3.9: More qualitative comparisons on ZInD-XV test set.

3.17.2 HM3D-XV Results

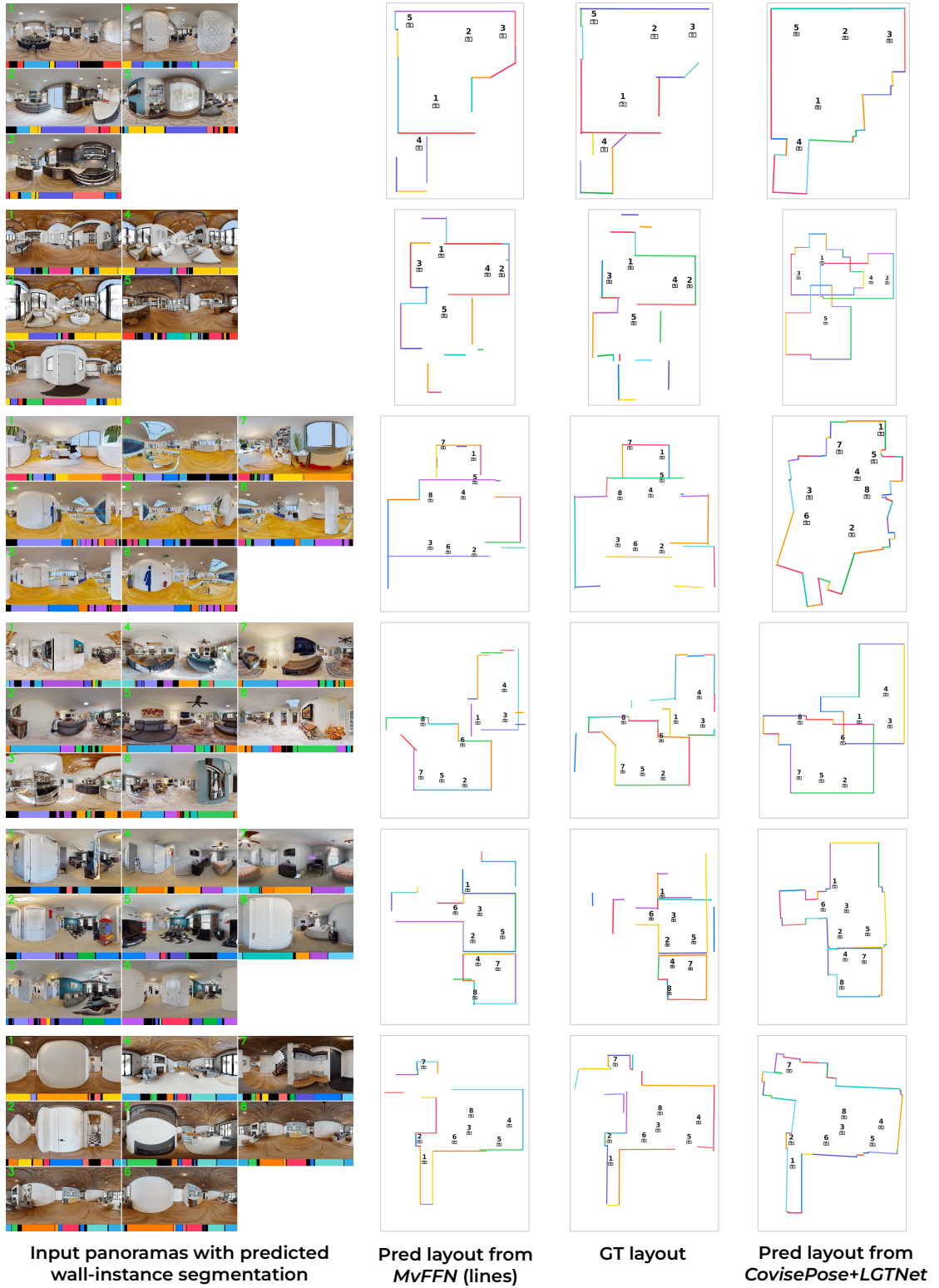


Figure 3.10: Qualitative comparisons on HM3D-XV test set.

Figure 3.10 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.

3.18 Failure Cases



Figure 3.11: 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. 3.11, 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 3.12: 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. 3.12. 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.

3.19 Dataset Preparation

3.19.1 ZInD-XV Dataset Preparation

ZInD-XV is derived from all scenes in the Zillow Indoor Dataset (ZInD) [17]. 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 [36]. 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* [27]. 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.

3.19.2 HM3D-XV Dataset Preparation

HM3D-XV is constructed from residential scenes in the Habitat-Matterport 3D Semantic Dataset (HM3DSem) [82], 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 [5], targeting approximately 2 panoramas per square meter at a fixed height of 1.2 m. Using the *Habitat-Sim* renderer [53], 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 [36].

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.

3.20 Conclusion

This chapter introduced *MvFFN*, a unified feed-forward model for joint pose and structured layout reconstruction from sparse, unposed, wide-baseline panoramas. Its key contribution is the *MOIS* reformulation, which replaces direct multi-room floor-plan prediction with cross-view wall instances and within-room connectivity, yielding a structured prediction target that is substantially easier to learn in this sparse setting. Under this formulation, *MvFFN* predicts coarse camera poses, dense global geometry, and wall-instance-level semantic outputs with globally consistent identities and local connectivity, which are assembled into globally consistent multi-room layouts through *Layout Chaining*. The results show that this formulation significantly improves pose and layout accuracy over prior modular baselines. Together, these findings show that reliable sparse indoor reconstruction requires tighter coupling between pose, geometry, and layout within a unified multi-view framework. The resulting wall-centered representation also provides a natural interface for downstream refinement, which is developed in Chapter 4.

Chapter 4

***BADGR*: Bundle Adjustment Diffusion Conditioned by Gradients for Wide-baseline Floor Plan Reconstruction**

Chapter 4 moves from feed-forward reconstruction to scene-level refinement by combining learned structural priors with iterative geometric optimization. Unlike *MvFFN*, whose goal is to predict pose and layout directly from sparse panoramas in a single forward pass, the focus here is to improve a coarse scene estimate by enforcing cross-view consistency under global structural constraints. This chapter is motivated by two observations. First, floor-plan data is substantially more accessible and structurally cleaner than fully annotated multi-view panoramic reconstructions, making it an attractive source of prior knowledge about plausible indoor layouts. Second, challenging scenes with wide baselines, sparse observations, and noisy boundary evidence often require iterative correction beyond feed-forward prediction. The method developed in this chapter, *BADGR*, addresses this setting by treating refinement as a learned optimization process over pose and layout, where reprojection-based consistency and priors learned from floor-plan data jointly guide global reconstruction.

4.1 Introduction

Reconstructing floor plan layouts and camera poses has become an important task with many applications such as virtual touring, interior design, and autonomous navigation. High spatial accuracy in both objectives is essential for high-fidelity downstream tasks, such as cross-view scene editing and dense reconstruction. Existing solutions for image-based layout reconstruction are either coarse, limited to a single room, or, while more accurate, require either densely captured image inputs or sparser capture with RGB-D cameras, which can be costly in terms of equipment, data bandwidth and capture efforts.

This work aims to accurately reconstruct camera extrinsics and floor plan layouts from sparsely captured 360° panoramas without prior pose information [17]. Specifically, our goals are to: (1) reconstruct the floor plan as a unique set of closed-loop polygons defining rooms and doors [85, 17]; (2) estimate each camera pose for view-consistency [22]; (3) accommodate diverse capture densities, down to one image per room; and (4) ensure that generated floor plans remain plausible within the natural distribution of training data, even when certain walls are occluded.

Accurate reconstruction of camera poses and spatial layouts in wide-baseline indoor environments is challenged by limited co-visibility and sparse features. This process demands not only view-consistency but also layout-structural constraints, such as *Manhattan* or

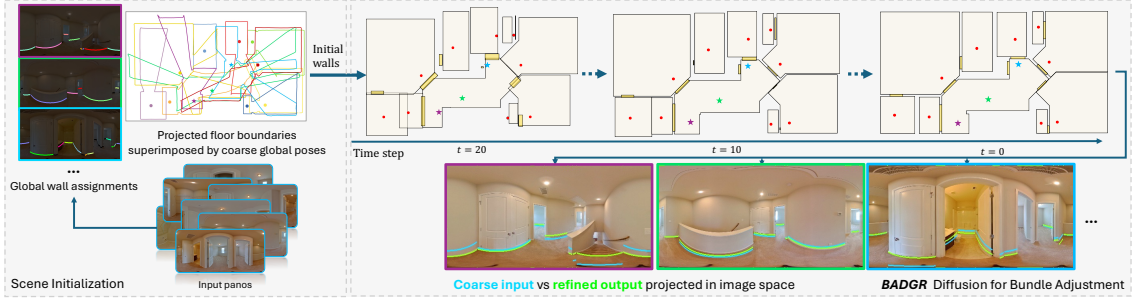


Figure 4.1: Overview of *BADGR*, a diffusion-based bundle adjustment (BA) model for generating precise, view-consistent camera poses and floor plan layouts. *BADGR* uses per-image floor boundaries and image column-to-wall assignments (upper left) as coarse input, refining poses and layouts through a gradient-conditioned denoising process (upper right). The bottom right shows view consistency by projecting the output layouts with the estimated poses.

Atlanta frameworks [55], wall thickness, collinearity, and prior knowledge of room layouts [88, 17, 30]. We propose *BADGR*, a conditional denoising diffusion probabilistic model (DDPM) trained to reconstruct and bundle adjust camera poses and layouts. *BADGR* employs a planar bundle adjustment (BA) module, to provide geometric guidance for conditioning the DDPM to maximize view-consistency from angle-constrained layouts and poses. The DDPM is also trained for the floor plan generation task. Combined with a reprojection loss, *BADGR* performs bundle adjustment through posterior sampling, with learned layout-structural constraints and the ability to handle noise from input features. The generative ability of the DDPM allows *BADGR* to predict plausible shapes of occluded layout sections based on training data. The *non-Markovian* inference process, i.e., predicting x_{t-1} from x_t by predicting x_0 , allows *BADGR* to combine a score-based generative model with a nonlinear optimization process, without specifying step size during training. *BADGR* differs from guided-diffusion style models [21, 13, 14, 78], where the gradients from differentiable objective functions are used only during inference to guide a pre-trained diffusion model, leading to issues like slow convergence and deviation from data manifold. Our experiments show that *BADGR* is more accurate in performing BA tasks compared to guided-diffusion style models.

The contributions of this work are: 1) *BADGR* is the first learning-based approach to jointly refine deformable room layouts from polygon-based floor plans and camera poses from sparsely captured RGB panoramas, guided by visually-derived features; 2) *BADGR* contains a novel approach to train a diffusion model for both nonlinear optimization (i.e., BA) and generation tasks (i.e., floor plan generation), allowing it to reconstruct poses and layouts from visual inputs and make reasonable guesses from learned layout-structural constraints; 3) *BADGR* obtains state-of-the-art accuracy in wide-baseline camera pose estimation and layout estimation with multi-view 360° panoramas.

4.2 Related Work

Our work bridges floor plan reconstruction and wide-baseline pose estimation, jointly reasoning over layout constraints and cross-view consistency.

Image-based Floor Plan Reconstruction involves estimating camera poses and creating a unique set of layout polygons. Research has shown good accuracy in producing floor plans in such formats from registered RGB-D point cloud scans [6, 62, 7, 85, 8]. Generative models [45, 46, 59] also demonstrate deep learning’s ability to model layout constraints, enabling floor plan generation from abstract inputs, like bubble diagrams [46]. Over the years, single-view room layout estimation has been extensively studied [89, 65, 51, 66, 76, 28]. However, noisy or unknown poses remain a key challenge for multi-view, wide-baseline layout reconstruction, as it requires 1) generating a single fused layout, 2) high accuracy in both camera poses and layouts for consistent views. Prior approaches attempted to regress poses and layouts from panorama pairs [77, 63, 30], yet often lack view-consistency without joint optimization [77].

Wide-Baseline Pose Estimation Analysis has shown [30, 27] that wide-baseline indoor pose estimation is challenging to traditional *Structure from Motion* (SfM), as images often contain featureless regions, repetitive textures, or narrow passageways, which limit co-visibility and cause drastic appearance changes across images [17]. Semantic constraints can help SfM establish anchors and loop-closures, but existing approaches often rely on heuristics and lack understanding of structural layouts [15, 16]. [58, 30] explored wide-baseline reconstruction as discrete problems, piecing rooms together like puzzles [26] and using vanishing points [86] to improve camera rotation estimates. However, these approaches can produce coarse poses and layouts with missing rooms from a larger floor plan. Recently, direct pose regression achieved good retrieval accuracy in solving coarse poses from a pair of [9, 27, 73] or up-to-5 [47] panorama inputs, along with predictions of dense correspondences among image columns and dense room layouts [27, 63]. A global pose graph can be later formed by merging local poses [47, 30], but this often lacks sufficient accuracy for good view-consistency. Furthermore, structural-layout constraints are frequently violated when superimposing projected single-view layouts on estimated poses [77], as shown in the starting floor plan of Figure 4.2.

Learned Optimization In traditional SfM, robust optimization often integrates sensor data and physical constraints iteratively to manage uncertainties in measurements and features. Recent data-driven approaches improved "front-end" feature extraction and matching [54, 48], adapting to noise and variability more effectively than classical methods. Differentiable optimizer, like differentiable LM [50, 71], have been used to train feature extraction with uncertainty prediction [44]. Uncertainty is also modeled from the parameter posterior distributions given initial measurements. G3R [10] iteratively optimizes 3D Gaussians using a gradient-conditioned 3D U-Net. PoseDiffusion [78], PhysDiff [84] apply

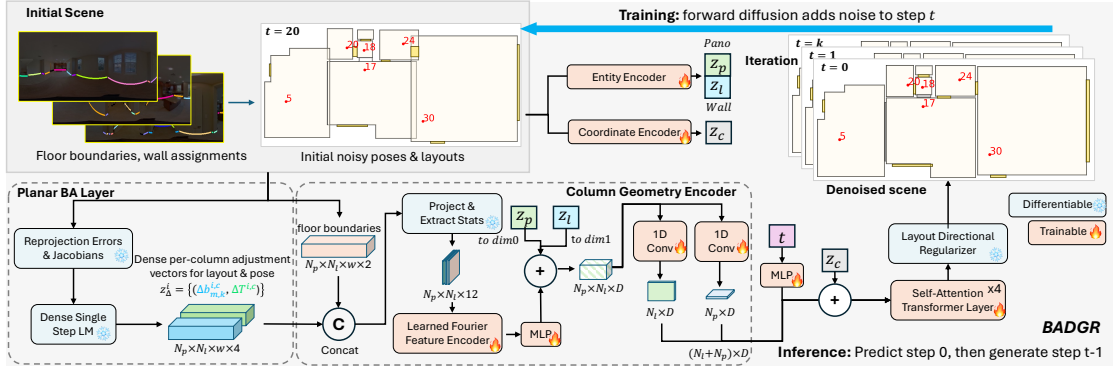


Figure 4.2: Architecture of *BADGR*. The forward process takes a ground truth scene, i.e. layouts and poses, adds noise to sample step t . The inference process uses a transformer, conditioned on dense per-column adjustments generated by the planar *BA* layer and compressed by the Column Geometry Encoder.

posterior sampling of pre-trained diffusion models for improved reconstruction. However, posterior-guided sampling can run into conflicts between guidance and diffusion flow function, which affects convergence [12]. *Non-Markovian DDPMs*, such as denoising diffusion implicit models (*DDIM*) [60], enable conditional diffusion training without explicit transitions from t to $t-1$, facilitating integration with nonlinear optimization independent of step size.

4.3 Problem Statement

Objectives: Given a set of sparsely captured 360° indoor RGB panorama images $\{P^i\}$ in equirectangular projections without pose information, the overall pipeline aims to estimate 3 degrees-of-freedom (*DoF*) camera poses $\{E^i, E^i \in SE(2)\}$, i.e. xy position, horizontal rotation angle, and 2D floor plans represented as a set of closed-loop polygons $\{V_m, V_m = (v_{m,1}, v_{m,2}, \dots, v_{m,k})\}$ for rooms and doors in a single global coordinate system, where m, k is room id and corner id, and $v_{m,k}$ is a 2D xy vertex coordinate. Our main focus, the proposed *BA*-based refinement component *BADGR*, aims to denoise camera xy positions $\mathcal{T}^i$ and angular constrained layouts $\{V_m\}$ by moving walls along their normal directions. *BADGR* optimizes for view-consistency on re-projected floor boundaries, assuming each image column contributes to the positional adjustment of a single wall and a camera, where the column-to-wall relations have been pre-assigned. *BADGR* also learns to maintain the layout-structural constraints of floor plans from training data.

Assumptions: We assume that all images are straightened [86] and can be connected to a single pose graph with co-visibility [17] to nearest neighbor greater than 10%. Relative camera heights are known across the floor. During *BADGR* optimization, wall angles are assumed to be fixed. The number of walls and wall connectivity are given; they can either come from merging the single-view layout estimation, post-processed under the *Atlanta World* assumption [28, 51], or via a quick human-in-the-loop process, see Section 4.5. Our floor plans follow the *Atlanta World* assumption [55] and have no curved walls.

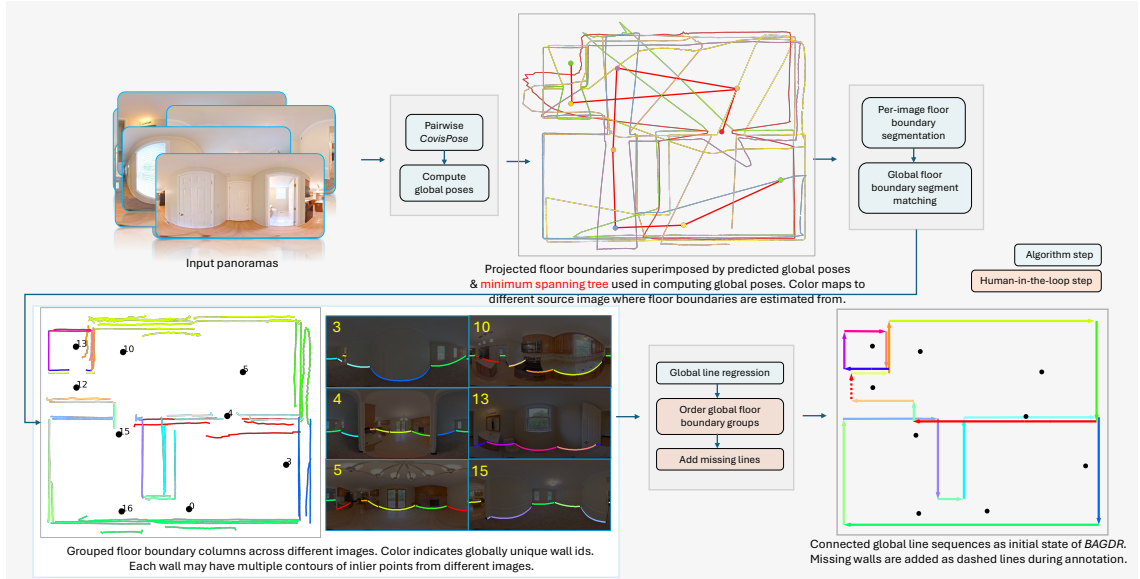


Figure 4.3: Overview of coarse scene initialization.

4.4 Pipeline Overview

We designed a coarse-to-fine pipeline, as shown in Figure 4.1, that initializes a scene with estimated camera poses and room layouts, as closed-loop polygons, combining per-view floor boundary predictions and image column-to-wall assignments, similar to semantic column matches across multiple images [76, 2]. This coarse initialization is subsequently refined with *BADGR*, which focuses on accurate reconstruction. Starting from the coarse scene, rotations from estimated poses and walls are corrected via vanishing point snapping [86]. During *BADGR* refinement, the scene is angularly constrained: *BADGR* optimizes only the 2-DoF camera xy position $\mathcal{T}^i$ for each pose E^i and the 1-DoF line translation $b_{m,k}$ for each wall $l_{m,k}$. While various algorithms could initialize the coarse floor plan for *BADGR*, we outline a specific pipeline in Section 4.5 using practical algorithms for coarse scene setup. The scene representation is discussed in Section 4.6, with details of *BADGR*, the proposed learned *BA* diffusion model, in Section 4.7.

4.5 Coarse Scene Initialization

BADGR operates on a coarse scene initialization consisting of camera poses, wall assignments, and an initial floor-plan layout. This initialization defines the scene state to be refined by the subsequent bundle-adjustment diffusion process.

Initial poses. From input panoramas $\{P^i\}$, a modified *CovisPose* model [27] is executed exhaustively on each pair of panoramas from the same floor. The model has the same architecture as the original *CovisPose* model and predicts: 1) relative camera pose $\tilde{E}^{(i,j)} \in SE(2)$, 2) floor boundaries $\{\tilde{B}^i\}$, and 3) cross-view co-visibility and angular correspondences $\{\tilde{a}^{i,j}\}, \{\tilde{\varphi}^{i,j}\}$. In addition, it predicts per-column binary room-corner labels $\{\tilde{V}^i\}$. The

model is trained on the ZInD dataset [17] using the same image pairs as [27], with an additional corner loss similar to [65]. Pose pairs with co-visibility score greater than 0.1 are selected to form a minimal spanning tree of the pose graph using a greedy algorithm, similar to [47]. Prior to computing global poses $\tilde{E}^i$, the pairwise poses $\tilde{E}^{(i,j)}$ are corrected through axis alignment with a 45° interval using predicted vanishing angles [86].

Initial walls. The per-panorama floor boundaries $\{\tilde{\mathcal{B}}^i\}$ are segmented using the predicted room corners $\{\tilde{\mathcal{V}}^i\}$. Inlier boundary points are then extracted with *RANSAC*, and initial local wall parameters $(\vec{v}_{m,k}^i, b_{m,k}^i)$ are computed for each wall segment detected from panorama P^i . Voting-based heuristics are used to match inlier boundary points, and hence local wall segments, across panorama pairs using $\{\tilde{\alpha}^{i,j}\}$, $\{\tilde{\phi}^{i,j}\}$, and $(\vec{v}_{m,k}^i, b_{m,k}^i)$. These pairwise local wall matches are aggregated into uniquely identifiable global wall identities $l_{m,k}$ shared across panoramas $\{P^i\}$. The corresponding global wall parameters $(\vec{v}_{m,k}, \hat{b}_{m,k})$ are then estimated with linear regression, where $\vec{v}_{m,k}$ is selectively axis-aligned to a 45° interval. Only wall angles within 10° of the dominant vanishing directions, e.g. $0^\circ, 45^\circ, 90^\circ$, and 135° , are corrected. Finally, an annotator uses a graphics interface to: 1) specify global wall connectivity, shown as arrows in the bottom-right image of Figure 4.3, and 2) add missing room corners with rough initial positions using the images and top-down projected floor boundaries, shown as dotted lines in the same figure. The number of room corners and wall orientations are treated as fixed input to *BADGR*.

Initialization during testing. During testing, a subset of panoramas is selected as described in Section 4.9. To generate coarse initial layouts for the selected subset, we use the connectivity of the annotated global scene described above, recompute the local wall parameters $(\vec{v}_{m,k}^i, b_{m,k}^i)$ for visible walls from the inlier boundary points of the selected panoramas, and inherit the parameters of invisible walls from the coarse scene generated using all available panoramas from ZInD. Only rooms with visible walls are included in the coarse initial layout. Similar to *PolyDiffuse* [8], this initialization uses light annotation. Our focus is the more difficult problem of global refinement; fully automating this initialization is left for future work.

4.6 Scene Representation for BA Optimization

Our proposed representation aims to uniquely define walls and cameras while assigning image columns to global walls, allowing *BA* to perform cross-view reprojections at any scene state. Each column links either to a specific wall or remains unassigned, simplifying reprojections by avoiding wall occlusion handling during floor boundary rendering. The global scene comprises room layouts $\{V_m\}$, doors as polygons with at max N_l walls, camera extrinsics $\{E^i\}$ of N_p panoramas, per-panorama floor boundaries $\{\hat{\mathcal{B}}^{i,c}\}$ for columns $\{\mathcal{C}^{i,c}\}$ with image width w , and a column-to-wall semantic assignment represented as a one-hot 3D array of shape $N_p \times w \times N_l$ for mapping $\{\mathcal{M} : \mathcal{C}^{i,c} \rightarrow l_{m,k}\}$. The layout walls

$\{V_m\}$ are represented as line segments $\{l_{m,k} \mid l_{m,k} = (v_{m,k}, v_{m,k+1})\}$ with each line represented in the Hesse normal form [1] allowing us to easily work with rotations and offsets $(\vec{v}_{m,k}, b_{m,k})$ from the origin. *BADGR* optimizes layout vertices V_m and camera positions $\mathcal{T}^i$, with these parameters normalized within $[-1, 1]$ as a continuous 2D coordinate array. Constant scene parameters include camera and wall rotation vectors $\mathcal{R}_i, \vec{v}_{m,k}$, camera height z^i and column-to-wall assignments $\{\mathcal{M}\}$. Details on scene initialization are covered in Section 4.5.

4.7 Bundle Adjustment Diffusion

Wide-baseline indoor reconstruction often suffers from a lack of robust matching features with sub-pixel accuracy. However, each wall can be observed by several image columns, with their floor boundary modeled as a line, as shown in Figure 4.4. *BADGR* tackles the multi-view floor plan and pose reconstruction problem using a planar bundle adjustment (BA), minimizing reprojection errors between the input predicted floor boundaries from image-based models and the projected wall positions. These errors are computed at the column level and are used to adjust the wall translation along its normal direction and the corresponding camera pose, optimized via a *Levenberg-Marquardt* (LM) step.

Noise in the floor boundaries can introduce errors in the final BA results. To mitigate the noisy signal and incorporate learned layout-structural constraints, *BADGR* integrates the planar BA mechanism into a conditional denoising diffusion process. Instead of averaging column-wise wall and camera movements, the model encodes these dense signals and uses them to condition a transformer-based diffusion model. In this framework, posterior sampling is employed to generate the final adjustments, using raw BA adjustments as inputs. The transformer model predicts the xy positions of camera poses and room vertices. To facilitate iterative interaction with planar BA, an angle-constrained scene representation is used, along with a *Layout Directional Regularizer* (LDR) to form a bidirectional map between 2D room vertex positions and wall positions while maintaining fixed angles. The details of this process are explained in the following section.

Architecture: Figure 4.2 shows the architecture of our proposed optimization pipeline. Taking inspiration from *HouseDiffusion* [59], *BADGR* has a diffusion model [25] based architecture with a truncated denoising inference process [40, 87] during training and inference. At each time t , our model takes a scene of 2D positions, i.e. $(\{\mathcal{T}^{i,(t)}\}, \{V_m^{(t)}\})$, conditioned on: 1) the wall and panorama metadata encoded by an Entity Encoder discussed below, 2) the floor boundary and adjustments embedding encoded by the Column Geometry Encoder, and generates an updated scene, i.e. $(\{\mathcal{T}^{i,(t-1)}\}, \{V_m^{(t-1)}\})$, for time $t - 1$. The architecture consists of: (a) a planar BA layer, (b) a Column Geometry Encoder module, where column-wise dense adjustments are encoded into a 1D embedding for each wall and panorama, (c) a Coordinate Encoder to embed 2D vectors for directional vector and xy

Algorithm 2: BA Optimization on Column $\mathcal{C}^{i,c}$

Input: Wall parameters $(b_{m,k}, \vec{v}_{m,k})$, camera pose $(\mathcal{T}^i, \mathcal{R}^i)$, camera height z^i , and floor boundary $\tilde{\mathcal{B}}^{i,c}$

Output: $\Delta b_{m,k}^{i,c}$ and $\Delta \mathcal{T}_{m,k}^{i,c}$

- 1 $\vec{ray}_{m,k}^{i,c} \leftarrow (\mathcal{T}^i, [\sin \mathcal{R}^i, \cos \mathcal{R}^i])$
- 2 $\vec{wall}_{m,k} \leftarrow (\vec{v}_{m,k}, b_{m,k})$
- 3 $pt_{m,k}^{i,c} \leftarrow \text{Intersect}(\vec{ray}_{m,k}^{i,c}, \vec{wall}_{m,k})$ // global coordinate system
- 4 $pt_{m,k}^{i,c} \leftarrow \text{GlobalToCam2D}(pt_{m,k}^{i,c}, \mathcal{T}^i, \mathcal{R}^i)$ // camera coordinate system
- 5 $\hat{\mathcal{B}}_{m,k}^{i,c} \leftarrow \text{Cam2DToPixel}(pt_{m,k}^{i,c}, z^i).row$
- 6 $\epsilon_{m,k}^{i,c}(b_{m,k}, \mathcal{T}^i) \leftarrow |\hat{\mathcal{B}}_{m,k}^{i,c} - \tilde{\mathcal{B}}^{i,c}|$
- 7 $\mathcal{J}_{m,k}^{i,c}(b_{m,k}, \mathcal{T}^i) \leftarrow \text{Jacrev}(\epsilon_{m,k}^{i,c})$
- 8 $(\Delta b_{m,k}^{i,c}, \Delta \mathcal{T}_{m,k}^{i,c}) \leftarrow \text{LM}(\epsilon_{m,k}^{i,c}, \mathcal{J}_{m,k}^{i,c}, b_{m,k}, \mathcal{T}^i)$
- 9 $\Delta b_{m,k}^{i,c} \leftarrow \text{AdaptiveHuber}(\Delta b_{m,k}^{i,c})$ // [4]
- 10 $\Delta \mathcal{T}_{m,k}^{i,c} \leftarrow \text{AdaptiveHuber}(\Delta \mathcal{T}_{m,k}^{i,c})$

coordinates, d) an Entity Encoder to generate an entity embeddings for metadata of each wall and panorama, functioning similarly as a positional encoding, and (e) a self-attention Transformer denoiser to integrate the embeddings along with time step and to estimate new 2D positions.

Entity Encoder: Similar to the “input conditions” from *HouseDiffusion*, a size-D identity embedding is generated for each wall and camera using metadata, e.g. wall direction vector, one-hot room type, room id and vertex id for wall, and camera direction vector, one-hot camera ids for panorama. Wall and camera embeddings are generated with separate *MLP* units.

Planar BA Layer: This layer generates camera and wall position adjustments by comparing the projected floor boundary from a given scene state, i.e. layouts and camera poses, with the input predicted floor boundary. The adjustments are computed densely on each image column with a *LM* optimization algorithm, which is set to run a single step. The process is demonstrated in Figure 4.4. Compared to *gradient descent* (*GD*) based optimization, *LM* is more efficient for convergence as it combines *GD* with the *Gauss-Newton* algorithm [24]. *LM* optimization requires the computation of the *Jacobian* matrix from each adjusted parameter at each column position. To enable efficient training of the proposed learned *BA* diffusion model, the *LM*-adjusted parameters for each feature column are limited to three primary values: $\Delta b_{m,k}^{i,c}$ representing wall bias and $\Delta \mathcal{T}_{m,k}^{i,c}$ denoting the *xy* displacement of the camera. This approach reduces computational complexity and memory overhead, enabling an efficient planar *BA* implementation as a PyTorch layer. However, this constraint necessitates pre-assigning each image column to a designated global wall to streamline the reprojection process by directly referencing the corresponding wall and camera positions without occlusion checks for multiple walls. This initial column-to-wall assignment is established during scene initialization, using feature-matching models such as *CovisPose*

to ensure robust alignment prior to *BADGR* optimization. The *BA* layer is implemented with the differentiable non-linear optimization library *Theseus* [50]; this library builds on PyTorch and applies a sparse solver with both CUDA and CPU implementations. See Algorithm 2 for implementation details.



Figure 4.4: Column-wise planar *BA* module. Positional adjustment for walls and cameras are computed for each image column. At each column, the associated wall $l_{m,k}$ is projected with the current camera pose E^i . The adjustments are computed by comparing the projected point to the floor boundary $\tilde{B}^{i,c}$ value. The dense per-column adjustments are estimated in parallel with our *BA* layer implementation.

Column Geometry Encoder: This encoder is designed to compress the dense adjustments $(\Delta b_{m,k}^{i,c}, \Delta \mathcal{T}_{m,k}^{i,c})$ and the floor boundary $\{\tilde{B}^{i,c}\}$ into a 1D embedding for each wall and camera. The input array of shape $N_p \times N_l \times w \times 6$ is stored as three 2D vectors densely collected from feature columns of N_p panoramas. Each column is assigned to at max a single wall. The resulting 1D embeddings, $N_p \times D$ for walls and $N_l \times D$ for cameras, are computed by compressing the three remaining axes. In our experiments, feature dimension $D = 1024$. Specifically, from a sparse array of shape $N_p \times N_l \times w \times 6$, the Column Geometry Encoder reduces the w dimension by computing the projected mean and standard deviation vectors along their corresponding wall normal and wall vector axes, among the valid values along the w axis. The Coordinate Encoder, which is based on *learned Fourier Features* [70], then encodes each stat vector into a size- D embedding z_c by learning two sets of weights for mean and standard deviation vectors. The weights for mean vectors are also used to encode $(\{\mathcal{T}^i\}, \{V_m\})$. These features are concatenated and fed into a single *Fully-Connected* (FC) layer to generate a geometric guidance embedding of shape $N_p \times N_l \times 1024$. Since adjustments $(\Delta b_{m,k}^{i,c}, \Delta \mathcal{T}_{m,k}^{i,c})$ are relatively small numbers compared to $(b_{m,k}, \mathcal{T}^i)$, We scale them by a factor of 100 prior to input to Column Geometry Encoder.

To further compress features at per-wall level, the entity embedding of cameras produced by the Entity Encoder is added into each of the N_l array columns. 2-layer 1D convolutions [74] followed by *LeakyReLU* [52] and a single FC layer are applied to produce a per-wall geometric guidance embedding of shape $N_p \times D$. Similarly, the entity embedding of walls is added into the N_p array rows. A separate similar 1D convolutional network is used to generate per-camera geometric guidance embeddings of shape $N_l \times D$.

Transformer Denoiser: Four layers of self-attention mechanisms [75] are used to denoise the input scene and reason about the relationships between different scene entities and geometric guidance. The layers use input / output features of shape $(N_p + N_l) \times D$. The output features are fed into a *FC* layer to produce xy coordinates as final outputs for $(\{\mathcal{T}^i\}, \{V_m\})$. In each attention layer, 3 types of attention heads with different masking schemes, i.e. Component-wise Self Attention mask (CSA), Global Self Attention mask (GSA) and Relational Cross Attention mask (RSA), were applied, similar to *HouseDiffusion* [59]. For the additional camera inputs, all cells related to valid cameras were left unmasked. Each masked type above contains 4 heads. The outputs of attention layers are summed and fed into add and norm layers. At the end, a single-layer *MLP* is used to predict diffusion noise from current time stamps t_n to t_0 .

Angle-Constrained Layouts: These layouts are critical to connect the planar *BA* module with the Transformer Denoiser in an iterative pipeline. The Planar *BA* module processes layouts as inputs and outputs with line representation $(\vec{v}_{m,k}, b_{m,k})$ of 1-DoF walls, while the Transformer works with 2D coordinates $\{V_{m,k}\}$. Essentially, the Angle-Constrained Layouts enable two-way mappings between $(\vec{v}_{m,k}, b_{m,k})$ and $\{V_{m,k}\}$, by defining half of the $\{V_{m,k}\}$ xy coordinate values predicted from the Transformer Denoiser as irrelevant to the final layout prediction. With wall directions $\vec{v}_{m,k}$ as fixed vectors, the DoF of $\{V_m\}$ are reduced by half using the xy validity mask $\tau_{m,k}$ of the same shape as $\{V_m\}$. $\tau_{m,k}$ is generated in the scene initialization stage, along with the wall directional vector $\{\vec{v}_{m,k}\}$. The invalid values defined in $\tau_{m,k}$ will be overwritten by outputs of the proposed *Layout Directional Regularizer* (LDR).

The *LDR* starts from a point $v_{m,k}$, with two valid xy coordinates, and updates the invalid xy positions of next vertex $v_{m,k+1}$, using wall direction $\vec{v}_{m,k}$, $v_{m,k+1}$ and validity $\tau_{m,k+1}$. The *LDR* is applied around the loop of each layout polygon to update all vertices. Angle constrained walls allow *BADGR* to be trained to predict wall movement along the normal direction, while still using 2D coordinates to represent the layout for design simplicity. $\tau_{m,k}$ is also used as an input condition and a mask in L2 loss computation.

Diffusion Model: *BADGR* adopts the *DDPM* process to learn to invert a diffusion process which adds noise to data with function $q(x)$ [25]. In the forward process, Gaussian noise is added to directly produce x_t from x_0 , same as [25]. We train the *non-Markovian* diffusion processes, to predict noise from x_t to x_0 and uses them to interpolate x_{t-1} [60]. This allows *BADGR* to combine nonlinear optimization and diffusion without explicitly defining each step size. Loss for measuring view-consistency can be directly applied to the predicted x_0 , which reuses the weighting scheme of a regular *DDPM* model for different time stamp during training. For inference, *probability flow ODE* [61] is used to iteratively denoise samples.

Loss Function: *BADGR* is trained to perform *BA* optimization, with 1) input conditions from dense column-wise *BA* adjustments, and 2) a reprojection loss, similar to the traditional *BA*, to regularize view-consistency. The loss function can be written as:

$$\mathcal{L}^{(t)} = \mathcal{L}_{L2}^{(t)} + \mathcal{W}_{proj} \mathcal{L}_{proj}^{(t)} \quad (3)$$

where $\mathcal{L}_{L2}^{(t)}$ is the masked L2 reconstruction loss, and is only computed for valid xy coordinates defined by $\tau_{m,k}$, from our Angle-constrained layouts. It's computed as:

$$\mathcal{L}_{L2}^{(t)} = \left\| \left((v_{m,k}^{pred}, \mathcal{T}^{i,pred}) - (v_{m,k}^{gt}, \mathcal{T}^{i,gt}) \right) \tau_{m,k} \right\|_2 \quad (4)$$

$\mathcal{L}_{proj}^{(t)}$ is the layout-to-image re-projection loss, which is computed among all the columns with pre-assigned global wall for *BA* adjustment using estimated scene at time 0.

$$\mathcal{L}_{proj}^{(t)} = \left\| \epsilon_{m,k}^{i,c,t=0} \right\|_1 \quad (5)$$

The process is described in steps 1-6 in Algorithm 2 measured in pixel units. $\mathcal{W}^{proj}$ is the time independent weight for the projection loss, set to 100.

4.8 Experiments

We design experiments using two types of floor plan data. First, we employ an end-to-end pipeline involving coarse scene initialization followed by *BADGR*, trained on a newly introduced FloorPlan-60K dataset and evaluated on ZInD [17], which has similar data collection and distribution characteristics. Second, we assess *BADGR* independently by training and testing it on the RPLAN dataset [81], with controlled noise added to both layout and virtual camera poses. We also report RPLAN-trained accuracy evaluated on ZInD to highlight *BADGR*'s key ability to train on 2D schematic views and generalize across datasets. While it doesn't perform as well as the FloorPlan-60K-trained model, RPLAN-trained *BADGR* still significantly outperforms our baseline approach.

4.9 Experiments with FloorPlan-60K Data

We use FloorPlan-60K, an extended version of ZInD [17], generated through a similar production pipeline to provide the scale needed for our diffusion-based *BA* training. FloorPlan-60K includes 68,147 floor plans, with an average of 8.7 rooms per plan and around 6.9 walls per room. Most walls align with Manhattan-world assumptions: 96.2% are at 90 degrees, 3.0% at 45 or 135 degrees, and 0.8% form other angles. We have permission from Zillow to use this dataset, with a public release pending from the owner. At a minimum, model weights will be available for reproducibility. Our training stage uses

only layouts and simulated poses, minimizing privacy concerns, while all evaluations are conducted on the public ZInD test set to ensure reproducibility.

BADGR is trained using a 2D ‘cleanup’ layer of floor plans, where larger spaces are represented by unions of multiple partially annotated room shapes, following the annotation approach of ZInD [17]. Panorama poses are randomly sampled within each room. For each input image, *BADGR* simulates data with a CUDA-based 1D renderer given floor plan layouts and a sampled camera pose. The renderer operates on connected rooms through doors, omitting door polygons and matching wall segments along the front and back planes. Random masking is applied on $\hat{B}^i$ and $\mathcal{M}$ to occasionally bypass the BA layer for selected image columns. During diffusion training, scenes are rotated by $[0^\circ, 90^\circ, 180^\circ, 270^\circ]$.

BADGR has a capacity of 300 walls and 30 panoramas, selected to accommodate 99% of the floor plans in FloorPlan-60K. It is trained with a batch size of 48, and with a learning rate of 10^{-4} for 140 epochs, 10^{-5} for 50 epochs, and 10^{-6} for 50 epochs by stepwise decay. *BADGR* is trained for the last 20 steps of a 1000-step diffusion process, using a second-moment schedule sampler for time t . *Ordinary Differential Equations* (ODE) sampling [61] is used during inference. Training peaks at 55GB GPU memory usage on a single GPU. During inference, *BADGR* processes a batch size of 1 in approximately 25 seconds on a CPU-only Apple M1 MacBook Pro with 32GB of memory, and around 4.0 seconds on an A100 GPU.

For end-to-end testing, we use panoramas from the public ZInD test set, consisting of 275 floor plans. Initial scenes, floor-boundary depths, and column-to-wall assignments are estimated from real panorama images, with coarse room layouts and positions initialized by the *CoVisPose+* method. These serve as inputs to our main contribution, the learned-based global refinement stage of *BADGR*.

4.9.1 Baseline Models

***CovisPose*:** The modified *CovisPose* model (see Section 4.5) was applied to each pair of straightened panoramas [86] from a given floor. For the first baseline, we used the original *CovisPose* method followed by the *Greedy Spanning Tree* (GST) algorithm [47] to generate global poses. For the second baseline, which is also the initial state of *BADGR*, we introduced *CovisPose+*, an improved version of *CovisPose* that incorporates vanishing point snapping in the pairwise pose estimation step before applying the GST.

***BA-Only*:** This method applies only the planar BA layer to refine camera and wall positions iteratively, starting from the coarse initialization in Section 4.5. First, per-column adjustments are made as described in steps 8-10 of Algorithm 2. These adjustments are then grouped by wall and camera, and averaged to produce the final position updates. To resolve conflicts during averaging, majority voting selects the dominant direction for each

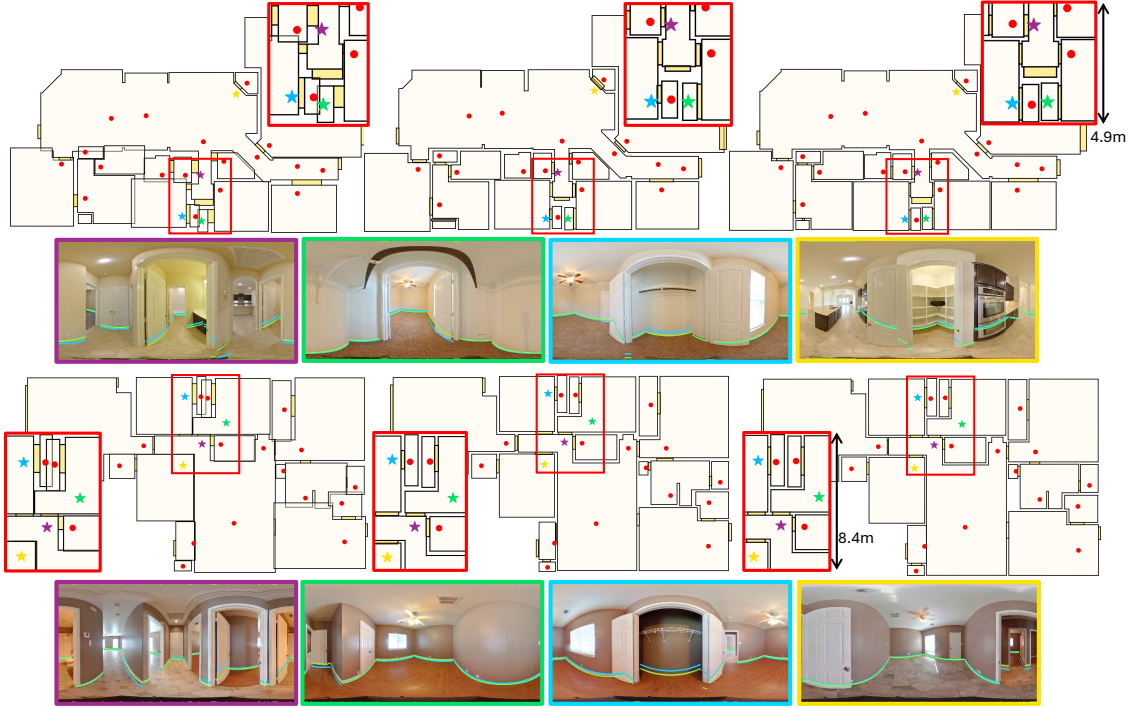


Figure 4.5: Qualitative results: top-down layouts and poses before (left), after *BADGR* optimization (middle), and GT (right). The reprojected geometry, *before* and *after* optimization, is shown in several images, highlighting the improved view-consistency, border colors indicate the capture positions. Example areas with significant improvements are *highlighted* and zoomed in. More examples in Appendix.

adjustment. The *LM* optimization is run for 100 iterations with a step size 2.5 times larger than the original.

Other Methods: *CovisPose* [27] outperforms point-matching *SfM*-based methods [39, 67] and recent learning-based approaches like *DirectionNet* [9] and *SALVe* [30] in wide-baseline indoor pose estimation. While *GraphCovis* estimates poses from three to five panoramas, it cannot be applied directly to our test cases, which typically involve more than 10 panoramas (see Table 4.1). Additionally, the lower pose errors shown in [47] are limited to configurations with only three panoramas, unsuitable as a comprehensive baseline. Similarly, *PanoPose* [73] estimates relative poses between pairs of panoramas but lacks publicly available code for reproducibility.

4.9.2 Results

We evaluate models on panorama subsets with varying input densities, as shown in Table 4.1. Pose errors (mean, median, standard deviation, and p90) are computed after aligning predicted and ground truth pose graphs using RANSAC [47], with similar metrics for layout errors. The number of images per room used in experiments is defined with partial rooms, where a complete *ZInD* room contains one or more partial rooms [17].

Table 4.1: Pose and layout error from predictions on ZInD dataset. Note that *BA-Only* and *BADGR* don’t optimize camera rotations, hence share the same rotation errors as *CovisPose+*.

Imgs/ Rm	Methods	Pose Rot.(°)			Pose Transl.(cm)			Visible walls (cm)				#walls
		Mn	Med	Std	Mn	Med	Std	Mn	Med	Std	p90	#panos
0.6	<i>CovisPose</i>	1.83	1.25	1.48	22.9	17.4	13.4	14.0	8.0	12.8	32.4	
	<i>CovisPose+</i>	0.24	0.20	0.30	20.7	15.7	12.0	11.5	6.9	10.5	26.0	59.4
	<i>BA-Only</i>	0.24	0.20	0.30	19.1	12.2	10.9	12.8	6.8	11.9	29.3	8.0
	<i>BADGR</i>	0.24	0.20	0.30	12.2	9.5	7.2	7.1	4.5	6.7	15.3	
1	<i>CovisPose</i>	1.88	1.36	1.40	23.0	18.0	13.2	15.0	9.0	13.0	39.1	
	<i>CovisPose+</i>	0.26	0.21	0.30	19.7	15.6	11.8	12.3	7.0	10.8	27.4	65.7
	<i>BA-Only</i>	0.26	0.21	0.30	17.6	12.6	12.0	12.6	6.7	11.4	27.5	10.4
	<i>BADGR</i>	0.26	0.21	0.30	11.2	8.8	6.5	7.0	4.6	6.6	15.8	
2	<i>CovisPose</i>	1.90	1.49	1.31	22.3	17.7	12.6	14.7	8.8	12.5	33.6	
	<i>CovisPose+</i>	0.26	0.19	0.30	17.9	14.5	10.2	12.0	7.0	9.0	26.0	65.9
	<i>BA-Only</i>	0.26	0.19	0.30	14.3	10.8	9.2	10.5	6.2	9.9	23.3	16.0
	<i>BADGR</i>	0.26	0.19	0.30	10.7	8.9	6.0	6.4	4.4	5.9	13.9	
3	<i>CovisPose</i>	1.70	1.19	1.23	21.5	17.1	11.9	14.3	8.5	12.1	31.8	
	<i>CovisPose+</i>	0.23	0.19	0.27	18.1	14.8	10.5	12.3	7.2	10.2	26.9	66.5
	<i>BA-Only</i>	0.23	0.19	0.27	13.4	10.7	8.2	10.6	6.2	9.0	22.1	18.3
	<i>BADGR</i>	0.23	0.19	0.27	10.6	8.9	6.0	6.6	4.3	6.0	14.6	

As shown in Table 4.1, the *BA-Only* baseline reduces pose and layout errors from *CovisPose+* when input density is sufficient, e.g. more than one image per room, with accuracy improving as input images increase. *BADGR* follows a similar trend, consistently achieving lower layout and pose errors across different input densities, including in extreme-baseline scenario with 0.6 images per room. This highlights the effectiveness of *BADGR*’s learned layout-structural constraints and its understanding of global context when compared to the view-consistency-only approach from the *BA-Only* baseline.

4.10 Reprojection Errors

Reprojection errors are reported in Table 4.2 to measure the view-consistency between the floor boundary projected from the predicted layout and poses and the per-image estimations, similar to the blue and green lines in the bottom-right images of Figure 1 of the main paper. The stats are computed from the per-column reprojection errors across all wall-assigned image columns, which is defined in Algorithm 1 of the main paper.

Table 4.2: Reprojection errors ($L1$ distance by pixel relative to an image size of 256×512) for wall-assigned columns, which measures view-consistency compared to the predicted floor boundary. Alongside Table 1 of the main paper, we observe that while *BADGR* sometimes produces higher re-projection errors than *BA-Only*, it consistently achieves lower layout and pose errors. This suggests that reprojection error influences accuracy but is not the sole factor in achieving high reconstruction accuracy. The stats are collected similarly to those from Table 1 of the main paper.

Img/Rm Method	0.6				1				2				3			
	Mn	Med	Std	p90	Mn	Med	Std	p90	Mn	Med	Std	p90	Mn	Med	Std	p90
<i>CovisPose+</i>	1.38	0.92	1.81	2.77	1.52	0.95	2.08	3.05	1.69	1.03	2.40	3.67	1.79	1.05	2.57	3.93
<i>BA-Only</i>	0.70	0.29	1.21	1.64	0.78	0.32	1.46	1.88	0.81	0.39	1.48	2.07	0.89	0.39	1.45	2.06
<i>BADGR</i>	0.65	0.31	1.17	1.36	0.75	0.32	1.34	1.77	0.78	0.35	1.37	1.81	0.81	0.36	1.33	2.04
GT Scene + Pred. Boundary	0.91	0.82	0.56	2.77	0.90	0.80	0.56	1.55	0.89	0.78	0.56	1.56	0.89	0.77	0.57	1.56

As Table 4.2 shows, overall reprojection error increases with the number of input images. This is caused by the accumulating pose errors and inconsistencies in floor boundary estimates across overlapping regions. Both *BA-Only* and *BADGR* consistently show lower reprojection errors compared to *CovisPose+*. In most cases, *BADGR* reports slightly lower reprojection errors than *BA-Only*, likely because *BA-Only* can get stuck in local minima of the loss function and uses a PyTorch implementation that also considers memory and speed. In this implementation, adjustments are computed at per-column level and averaged to update poses and walls, rather than optimizing the total reprojection error across all columns.

4.11 Experiments with RPLAN

The RPLAN dataset [81] is used to evaluate *BADGR* in controlled noise experiments during training and testing. RPLAN is a large-scale dataset of real residential floor plans, each with 4 to 8 Manhattan rooms, spanning 65 to 120 square meters, but without real-world scale. After pre-processing [59] and aligning door edges with walls, we use 57,303 plans for training and 19,000 for testing. The *BADGR* model is trained for 20 diffusion steps with an embedding size $D = 512$, handling up to 100 walls and 15 cameras, using the training procedure from FloorPlan-60K. Since RPLAN lacks real indoor images, we use simulated camera poses and rendered floor boundaries for evaluation. Noisy layouts and poses are created by adding Gaussian noise (mean = 0, standard deviation = 3.3%) to the ground truth wall and camera positions. These inputs are normalized to a range of $[-1, 1]$, and error distances (shown in Table 4.3) are reported in this normalized space.

Simulated Noise in Floor Boundary Inputs We added noise to each rendered floor boundary to simulate bias from boundary prediction models, caused by factors like occlusion and limited training data. Before rendering the floor boundary for each simulated camera position, we randomly translate each visible wall along its normal direction, adjusting the opposite side to avoid self-intersection. The walls altered are selected by chance. Noise

Table 4.3: Pose and layout errors from predictions on RPLAN dataset. Since RPLAN floors are under 120 square meters, the reported distance error in percentage roughly translates to 1% to 0.1 meter in real scale.

Imgs / Rms	State / Method	Pose Err Dist (%)				Visible Layout Err Dist (%)				# walls # panos
		Mn	Med	Std	p90	Mn	Med	Std	p90	
1	Start	3.15	2.94	1.56	4.98	1.69	1.46	1.25	3.42	46.7 8.0
	<i>BA-Only</i>	0.58	0.41	0.31	0.93	0.62	0.30	0.73	1.25	
	BADGR	0.35	0.29	0.18	0.55	0.33	0.15	0.38	0.58	
2	Start	3.21	2.93	1.59	5.20	1.70	1.45	1.26	3.44	46.7 15.0
	<i>BA-Only</i>	0.62	0.31	0.38	1.09	0.58	0.23	0.73	1.11	
	BADGR	0.34	0.23	0.17	0.54	0.27	0.13	0.33	0.47	

Table 4.4: Errors from predictions on RPLAN dataset with simulated noise in floor boundaries. First column contains input density, the chance and max scale of noise added to each visible wall.

Imgs/Rms %noise	State/ Method	Pose Err Dist (%)				Visible Layout Err Dist (%)			
		Mn	Med	Std	p90	Mn	Med	Std	p90
1	Start	3.17	2.87	1.54	4.98	1.69	1.46	1.24	3.40
5% chance	<i>BA-Only</i>	1.47	0.60	1.19	2.70	2.22	0.52	2.81	2.53
2%scale	BADGR	0.59	0.42	0.35	0.98	0.49	0.27	0.50	0.95
1	Start	3.17	2.92	1.57	4.99	1.69	1.45	1.24	3.41
10% chance	<i>BA-Only</i>	1.63	0.79	1.25	2.93	1.95	0.62	3.07	2.67
2%scale	BADGR	0.64	0.46	0.36	1.04	0.51	0.29	0.52	1.00
2	Start	3.20	2.86	1.58	5.18	1.69	1.45	1.24	3.40
5% chance	<i>BA-Only</i>	1.48	0.58	1.19	2.49	1.24	0.45	2.69	2.23
2% scale	BADGR	0.65	0.41	0.33	1.05	0.43	0.22	0.45	0.80
2	Start	3.16	2.84	1.57	5.11	1.69	1.45	1.26	3.41
10% chance	<i>BA-Only</i>	1.57	0.74	1.22	2.74	2.16	0.55	0.60	2.51
2% scale	BADGR	0.72	0.46	0.38	1.17	0.56	0.25	0.57	0.87

follows a uniform distribution with a max scale, and is applied independently for each rendering. The resulting floor boundaries guide the *BADGR* denoising process during testing.

With a maximum noise level of 2% (about 20 cm), both *BADGR* and *BA-Only* show increased pose and layout errors compared to no noise. However, *BADGR* is significantly less impacted by input perturbations than *BA-Only*. In terms of absolute distance error for poses and layouts, *BADGR* achieves lower errors compared to the ZInD test case when approximate scale is applied, likely due to the simpler structure of RPLAN floor plans.

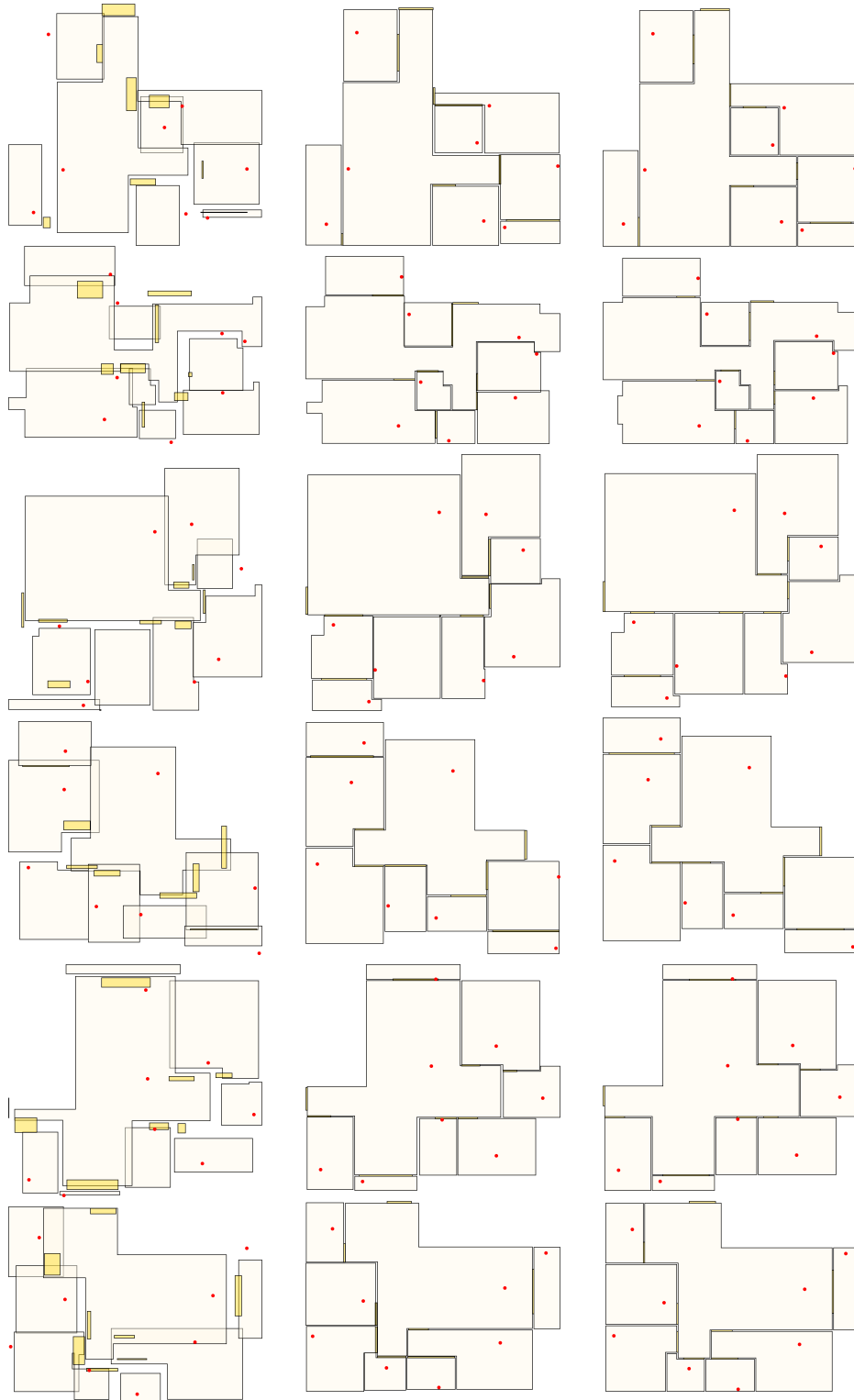


Figure 4.6: Qualitative results trained and tested on RPLAN dataset (page 4), with input densities at one input image from each partial room. The topdown views from left to right are before, after *BADGR* optimization and the ground truth. The initial state is created by adding Gaussian noise from 20-step diffusion q-sampling [25] into the ground truth poses and layouts. Details see Section 8.2 of the main paper. This figure demonstrates *BADGR*'s capability to refine initial scenes with much higher noise than those from the ZInD test cases.

4.12 Cross dataset training and validation

We additionally trained a *BADGR* model of a max capacity of 300 walls and 30 cameras with the RPLAN training set, and with a similar settings of sampling camera positions for generating simulated floor boundaries and column-to-wall assignments. This model is evaluated on the ZInD test set. The results are listed in Table 4.5 alongside with existing results for comparison. Although *BADGR* trained with RPLAN dataset doesn't produce similar or higher accuracy than *BADGR* trained with FloorPlan-60K dataset, it still outperform the *CovisPose+* and *BA-Only* baselines. This trend is expected as RPLAN contains Manhattan floors only and overall have less rooms and panoramas during training.

Table 4.5: Pose and layout errors tested on the ZInD dataset, trained with various datasets. Row 3 of each block presents additional results compared to the main paper. Mn, Med, and Std denote mean, median, and standard deviation, respectively. We also report the 90th percentile (p90) of the absolute translation errors for the estimated camera poses.

Imgs/ Rm	Methods	Training Set	Camera Translation(cm)				Visible walls (cm)			
			Mn	Med	Std	p90	Mn	Med	Std	p90
0.6	<i>CovisPose+</i>	ZInD	20.7	15.7	12.0	33.6	11.5	6.9	10.5	26.0
	<i>BA Only</i>	N/A	19.1	12.2	10.9	32.7	12.8	6.8	11.9	29.3
	<i>BADGR</i>	RPLAN	14.7	11.4	8.0	24.0	9.4	5.8	8.6	20.6
	<i>BADGR</i>	FloorPlan-60K	12.2	9.5	7.2	18.5	7.1	4.5	6.7	15.3
1	<i>CovisPose+</i>	ZInD	19.7	15.6	11.8	33.7	12.3	7.0	10.8	27.4
	<i>BA Only</i>	N/A	17.6	12.6	12.0	34.2	12.6	6.7	11.4	27.5
	<i>BADGR</i>	RPLAN	15.1	11.1	8.5	26.5	9.2	5.9	8.8	20.6
	<i>BADGR</i>	FloorPlan-60K	11.2	8.8	6.5	18.6	7.0	4.6	6.6	15.8
2	<i>CovisPose+</i>	ZInD	17.9	14.5	10.2	30.8	12.0	7.0	9.0	26.0
	<i>BA Only</i>	N/A	14.3	10.8	9.2	30.1	10.5	6.2	9.9	23.3
	<i>BADGR</i>	RPLAN	14.1	10.5	8.6	27.8	9.1	5.6	8.7	20.1
	<i>BADGR</i>	FloorPlan-60K	10.7	8.9	6.0	17.1	6.4	4.4	5.9	13.9
3	<i>CovisPose+</i>	ZInD	18.1	14.8	10.5	31.6	12.3	7.2	10.2	26.9
	<i>BA Only</i>	N/A	13.4	10.7	8.2	30.5	10.6	6.2	9.0	22.1
	<i>BADGR</i>	RPLAN	13.3	10.2	7.7	22.9	9.4	5.9	8.7	21.1
	<i>BADGR</i>	FloorPlan-60K	10.6	8.9	6.0	17.5	6.6	4.3	6.0	14.6

4.13 Compatibility with *MvFFN* Initialization

In addition to the original coarse scene initialization used in the main *BADGR* experiments, we also study a smaller-scale initialization obtained directly from *MvFFN* predictions. This experiment is intended as a compatibility test showing that the wall-instance representation predicted by *MvFFN* can be used as input to downstream refinement. We apply planar bundle adjustment and *BADGR* [36] as post-processing to the *MvFFN* predictions from Sec. 3.12.2, and report the resulting matched-wall distance statistics in Table 4.6.

Planar bundle adjustment [36] refinement helps the majority of scenes but degrades a substantial minority. 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.

Table 4.6: Matched-wall distance before and after refinement on ZInD-XV. Distances are in cm.

View	Method	All samples				Improved samples				Degraded samples				
		Mean dist		Med. dist		Improved (%)	Mean dist		Med. dist		Mean dist		Med. dist	
		before	after	before	after		before	after	before	after	before	after	before	after
5-view	<i>MvFFN+BA</i>	13.9	13.7	6.9	6.7	68.0%	14.8	12.3	7.5	5.9	12.0	17.2	6.0	8.3
	<i>MvFFN+BADGR</i>	13.9	17.7	6.9	8.1	45.9%	16.2	14.1	8.2	6.2	12.1	20.5	6.1	10.3
8-view	<i>MvFFN+BA</i>	19.0	18.8	8.8	8.8	66.0%	20.6	17.8	9.7	8.2	15.9	20.8	7.2	10.0
	<i>MvFFN+BADGR</i>	19.0	21.0	10.1	10.4	48.5%	22.9	20.0	8.8	8.8	15.6	23.9	7.5	11.9

Table 4.6 reports matched-wall distance per method, split by scenes that improve or degrade. Planar *BA* improves 68% of 5-view scenes and 66% of 8-view scenes (mean error 13.9 to 13.7 cm and 19.0 to 18.8 cm), but the 32-34% that degrade absorb most of the gain. Unsurprisingly, *BADGR* is less effective because it assumes complete layouts and clean correspondences. It improves only 46-48% of scenes and raises mean error overall (13.9 to 17.7 cm and 19.0 to 21.0 cm). *MOIS* correspondences support refinement, but reliably detecting when to apply *BA* needs further study.

4.14 Ablation Studies

We demonstrate the impact of *BADGR* in recovering poses and layouts by comparing to four models and training three *BADGR* variants in Table 4.7. Our conditional diffusion model *BADGR* outperforms guided diffusion model *BA+DM* in both tasks. *BADGR* trained with *BA* inputs only has a larger improvement compared with *BADGR* trained with reprojection loss only. We also found that although having higher errors from the first four models, the resulting layouts are mostly plausible looking, with improved layout and pose accuracy from the starting point.

4.15 Qualitative Results

As shown in Figure 4.5, *BADGR* improves layouts and view-consistency, even in extreme cases with minimal visual overlap, from a coarsely initialized scene. *BADGR* is able to learn the physical constraints from training data and correct issues like overlapping room layouts and varying wall thicknesses. While pose errors are not easily visualized in a top-down view, the image view reveals these inaccuracies and highlights the substantial improvements in both poses and layouts achieved by *BADGR*. Reprojection errors is also provided below.

Table 4.7: Ablation analysis on different variants of *BADGR*. All models are trained on FloorPlan-60k datasets and tested on ZInD test set. Diffusion model (*DM*) is *BADGR* trained without planar *BA* layer and reprojection loss. *BA+DM* is a guided diffusion model, where *BA* adjustment is added to the diffusion adjustment from *DM* above without *BA* conditioning.

BA Inputs	Reproj Loss	Imgs/Rms	1				2			
		Method	Vis. walls (cm)		Pose (cm)		Vis. walls (cm)		Pose (cm)	
			Mn	Std	Mn	Std	Mn	Std	Mn	Std
✗	✗	<i>DM</i>	8.6	8.3	26.7	17.4	8.6	8.1	25.7	17.7
✗	✓	<i>BADGR</i>	8.4	8.4	23.6	14.1	8.3	8.1	24.4	14.3
✗	✗	<i>BA + DM</i>	8.6	9.0	15.0	10.0	8.0	8.3	13.4	8.7
✓	✗	<i>BADGR</i>	7.2	6.6	12.0	6.8	6.9	6.4	11.3	6.5
✓	✓	<i>BADGR</i>	7.0	6.6	11.2	6.5	6.4	5.9	10.7	6.0

4.16 Discussion

PuzzleFusion (*PF*) [26] and *Extreme SfM* (*E-SfM*) [58] also produce floor plan layouts and camera poses. Here we provide a discussion on their differences to *BADGR*. Both *PF* and *E-SfM* estimate the rotation and translation of given unposed non-deformable room layouts by solving jigsaw puzzles. Camera poses are then inferred from the puzzle solution. This has different objectives than ours: 1) within the same room their relative positions among individual walls and multiple camera poses stay unchanged; 2) neither method uses information from a set of horizontal-facing images without precise poses as input constraints to guide optimization for view-consistency. Both can be used for initialization of *BADGR* like *CovisPose*. Code and weights of *PF* trained on RPLAN aren’t publicly available. We contacted the authors, and the code no longer runs. *E-SfM* takes hours or even days to process a single house [26], so neither can be used as baselines. *BADGR* solves a different task as we are deforming room shapes. Instead, we simulate *BADGR* refining *PF*-initialized layouts by adding Gaussian noise (10.55 *Mean Positional Error in pixels* (MPE) matches *PF*) to room translations, with relative poses among cameras and walls within each room given for initialization. *BADGR* reduces the MPE↓ of room placement from 10.55 or 4.1% (normalized by 256×256 pixel resolution) (full RPLAN) to 0.93% (77.3% lower), calculated by average shift of each vertex. We also report 0.98%, 1.45% mean translation errors in layout and poses. MPE of *E-SfM* is only reported for small RPLAN as 29.44 or 11.5% [26].

GraphCovis [47] and *PanoPose* [73] didn’t publicly release their code. *GraphCovis* estimates global poses among up-to-5 input panorama images. We compared the pose errors of *BADGR* with *GraphCovis* under the similar input settings originally evaluated on ZInD [17] in table 4.8. It demonstrates *BADGR*’s robust performance among different sizes of homes and missing room scenarios.

Table 4.8: Statistics of *absolute translation error* and *absolute rotation error* on group of three, four, and five panoramas for *GraphCovis* (GC) and *BADGR*. The accuracy of *GraphCovis* is imported from Table 1 of [47].

Group-Size # imgs	GC Rot ↓ (°)			BADGR Rot ↓ (°)			GC Transl. ↓ (cm)			BADGR Transl. ↓ (cm)		
	Mn	Med	Std	Mn	Med	Std	Mn	Med	Std	Dist (cm)	Med	Std
3	2.00	0.85	9.15	0.25	0.20	0.28	8.1	3.8	29.2	9.2	6.1	5.9
4	3.19	0.94	13.36	0.26	0.22	0.30	15.3	6.1	43.0	10.6	6.1	6.0
5	3.29	1.07	12.04	0.25	0.19	0.30	17.2	8.2	38.4	10.4	6.3	6.7

4.17 Failure Cases

We present three failure cases to highlight the challenges and opportunities for *BADGR*. Overall, *BADGR* achieves high accuracy when input images are minimally-connected by covisible walls through column-to-wall assignments from the initial coarse scene. However, since *BADGR* is trained on simulated panorama poses and column-to-wall assignments, the model can struggle when faced with scenarios outside the training distribution. An example is shown in Figure 4.7, where the initial scene contains large errors over wide areas, and the column-to-wall assignments fail to establish critical covisible walls between panoramas. This underscores the need for future development of an end-to-end initialization method to establish global column-to-wall assignments.

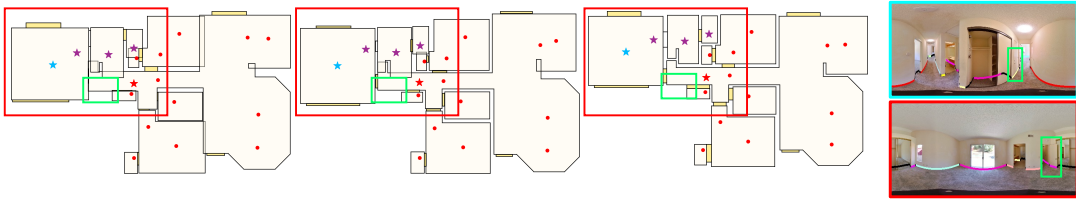


Figure 4.7: Failure case example caused by errors in coarse scene initialization. The colored lines in the images on the right represent estimated floor boundaries, with colors indicating their assigned unique global wall id. The heuristic failed to match two wall segments (highlighted in **rectangles**) using dense column correspondences and floor boundaries from the *CovisPose* model. Additionally, the large initial error in the highlighted section (highlighted in **rectangles**) may fall near the boundary of the 20-step truncated diffusion data distribution, contributing to the issue.

BADGR relies on floor boundaries for positional information along the normal direction of the target surface. This explains the failure case in Figure 4.8, where the wall length is estimated incorrectly due to a lack of guidance to the model. Future work could incorporate cues from wall junctions, similar to [65, 76], or encode pixel positions of pre-assigned columns to better constrain visible walls and infer invisible wall positions.

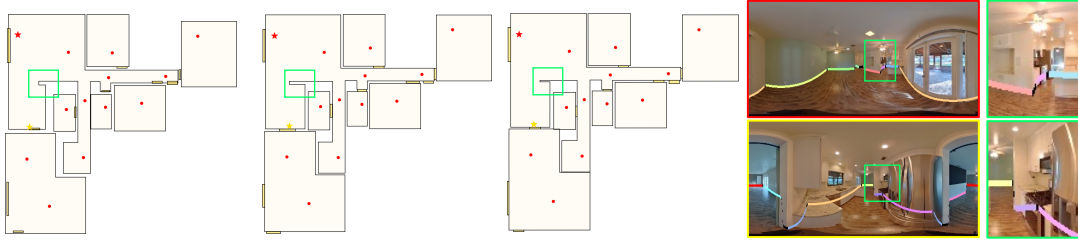


Figure 4.8: Failure case example where the **highlighted** wall is predicted with an incorrect length due to the limited image column coverage. The colored lines in image views represent similarly as in Figure 4.7.

BADGR assumes consistent floor heights throughout the area. When this assumption is violated, such as with a sunken floor (Figure 4.9), planar *BA* may place walls farther than their actual positions. Future work may include extending *BADGR* to represent varying camera height and wall heights, and expanding the training data to cover this issue.

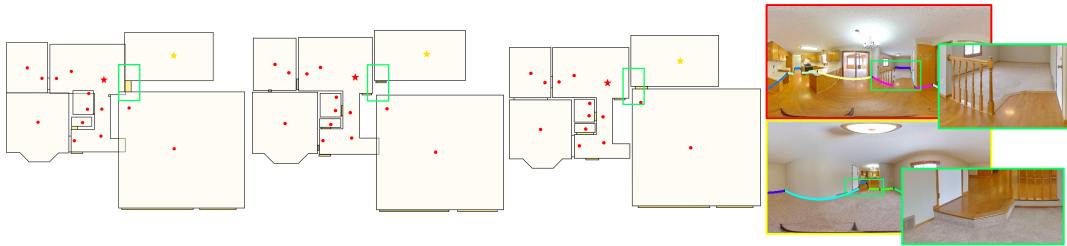


Figure 4.9: Failure case example due to inconsistent floor heights. The colored lines in image views represent similarly as in Figure 4.7.

4.18 Conclusion

This chapter introduced *BADGR*, a refinement framework for joint pose and layout improvement from coarse indoor reconstructions. By combining learned structural priors with bundle-adjustment-style optimization, *BADGR* moves beyond one-shot prediction and enforces scene-level consistency under sparse, wide-baseline observations. The experiments show that this combination improves reconstruction quality across challenging indoor settings, while also generalizing across datasets and input densities. More broadly, this chapter shows that accurate indoor reconstruction is best approached not only as a feed-forward prediction problem, but also as a refinement problem in which learned priors and geometric optimization work together under global scene constraints.

Conclusion

5.1 Summary

This thesis studied structured indoor reconstruction from sparse, unposed, wide-baseline 360° panoramas, with the goal of recovering globally consistent camera poses and multi-room floor-plan layouts from limited visual observations. Across the progression of this dissertation, the central challenge was not only estimating local geometry from individual views, but also resolving the global structural ambiguity that arises when poses are uncertain, overlap is sparse, and room layout must remain consistent across views. The results of this thesis show that accurate reconstruction in this setting depends on treating pose, geometry, and layout as tightly coupled aspects of the same problem rather than as loosely connected stages.

The early systems in this thesis, centered on the ZInD pipeline, *SALVe*, and *PSMNet*, established several practical insights that shaped the later methods. First, they showed that pose and layout are the two core components of floor-plan reconstruction, and that accurate floor plans depend fundamentally on cross-view consistency rather than isolated single-view predictions. Second, they showed that pipelines built by first estimating local room layouts and then stitching them together can work in practice, but remain limited: structural details that are weak, occluded, or incomplete in one panorama often require evidence from another view, and errors in pose and layout propagate across stages. Third, they showed that wide-baseline indoor localization is itself a major bottleneck, since sparse overlap, repeated structure, and weak texture often make classical feature-based matching incomplete or unstable. Together, these systems clarified that reliable large-scale reconstruction requires representations and inference procedures that bring cross-view reasoning earlier, couple pose and structure more directly, and maintain global consistency throughout reconstruction.

Motivated by these observations, this thesis reformulates floor-plan reconstruction through *Multi-view Ordered Wall Instance Segmentation* (MOIS). Rather than directly predicting a full multi-room floor plan, which is difficult to learn under sparse observations, ambiguous room partitions, and noisy cross-view supervision, MOIS casts the problem as estimating cross-view wall instances and their within-room connectivity. Under this formulation, the thesis introduces *MvFFN*, a feed-forward multi-view transformer with two complementary branches: a geometry branch that predicts coarse camera poses and dense global geometric support, and a semantic branch that predicts wall-instance-level outputs with globally consistent identities and local connectivity. These predictions are assembled

into globally consistent multi-room layouts through *Layout Chaining*, while the resulting wall-centered representation also enables downstream layout bundle adjustment for scene-level geometric refinement. Together, these results show that the main advance is not only a stronger model, but a reformulation that makes sparse wide-baseline floor-plan reconstruction substantially more tractable and structurally grounded.

The thesis then studies refinement beyond feed-forward prediction through *BADGR*. While *MvFFN* provides strong coarse reconstructions, precise pose and layout recovery remains difficult in challenging scenes with wide baselines, sparse observations, and noisy boundary evidence. *BADGR* addresses this by unifying layout generation and bundle-adjustment-style optimization, rather than treating learned priors and geometric refinement as separate stages. It is conditioned on dense per-column updates from a single-step *Levenberg–Marquardt* optimizer and learns to predict camera and wall positions while minimizing reprojection error for view consistency. Under this design, diffusion-based generation contributes global structural constraints, including wall adjacency, collinearity, and plausible room organization, while bundle adjustment enforces consistency from co-visible evidence across images. More broadly, this result shows that robust indoor reconstruction from sparse panoramas requires not only stronger intermediate representations, but also a tight coupling between learned structural priors and iterative geometric optimization.

Taken together, the contributions of this thesis trace a progression from early human-in-the-loop systems and targeted subproblems toward more unified methods for joint pose and layout reconstruction. They show that reliable indoor reconstruction from sparse panoramic imagery requires cross-view reasoning to be built directly into both the representation and the inference procedure. More broadly, these results suggest a path toward indoor scene understanding systems that jointly reason about geometry, semantics, structure, and uncertainty across views, while remaining compatible with global optimization and scalable real-world deployment.

5.2 Future Work

MvFFN. A primary limitation of *MvFFN* is data scale. The current model is trained in a regime where richly annotated cross-view layout data remain scarce, limiting both representation capacity and generalization. A clear next step is to scale *MvFFN* toward a broader multi-view foundation model for structured indoor reconstruction by training on more diverse datasets with wider variation in scene type, capture pattern, and annotation noise. A second direction is to build on stronger geometry backbones. *MvFFN* currently operates on a largely 2D geometric representation, whereas recent multi-view 3D foundation models such as *MV-DUST3R* and *VGGT* suggest richer alternatives for cross-view geometric reasoning. This opens two promising paths: adapting such models to produce

layout-oriented 2D representations compatible with the *MOIS* formulation, or extending *MvFFN* itself into a 3D version that preserves *MOIS*-style prediction while reasoning over stronger multi-view geometry. A third limitation is that *MvFFN* is not fully end-to-end trainable. The current system relies on *Layout Chaining* as a non-learned post-processing step, and residual noise in wall instances and local connectivity can still degrade final layout quality. Replacing this stage with a learned post-processing or graph-repair module could improve end layout accuracy and make the reconstruction pipeline more tightly optimized. Finally, *MvFFN* still depends on accurate structural annotation. Reducing this dependency through self-supervised, semi-supervised, or continual-training strategies is an important direction for scaling the method to broader real-world data with weaker supervision.

***BADGR*.** An important next step for *BADGR* is to move beyond refinement of relatively coherent coarse layouts and explicitly address structurally incomplete initialization. In practice, coarse floor plans may contain partial room layouts, broken wall connectivity, or missing wall instances due to incomplete capture coverage or upstream prediction failures from *MvFFN*. A learned layout-repair stage that restores the layout graph before geometric refinement could therefore substantially improve robustness. Such a module could be used either as a preprocessing stage before *BADGR* or as an integrated component within the refinement loop itself, allowing graph repair and bundle-adjustment-style optimization to reinforce one another. A second direction is tighter coupling with *MvFFN*. Although *BADGR* is compatible with *MvFFN* initialization, its training distribution can be expanded further by fine-tuning on real *MvFFN* predictions rather than cleaner synthetic or oracle-style initializations. This would expose the model to the noisier and more diverse failure modes encountered in practice and improve its ability to handle realistic bundle-adjustment tasks. More broadly, this points toward a unified coarse-to-fine system in which feed-forward prediction, structural repair, and generative geometric refinement are trained to complement one another under a shared reconstruction objective.

References

- [1] Hesse normal form. https://en.wikipedia.org/wiki/Hesse_normal_form. Accessed: 2025-03-18.
- [2] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.
- [3] 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.
- [4] Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4331–4339, 2019.
- [5] Robert Bridson. Fast Poisson Disk Sampling in Arbitrary Dimensions. In *ACM SIGGRAPH 2007 Sketches*, pages 22–es. ACM, 2007.
- [6] 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.
- [7] Jiacheng Chen, Yiming Qian, and Yasutaka Furukawa. HEAT: Holistic edge attention transformer for structured reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3866–3875, 2022.
- [8] Jiacheng Chen, Ruizhi Deng, and Yasutaka Furukawa. PolyDiffuse: Polygonal Shape Reconstruction via Guided Set Diffusion Models. *arXiv preprint arXiv:2306.01461*, 2023.
- [9] Kefan Chen, Noah Snavely, and Ameesh Makadia. Wide-baseline relative camera pose estimation with directional learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3258–3268, 2021.
- [10] Yun Chen, Jingkan Wang, Ze Yang, Sivabalan Manivasagam, and Raquel Urtasun. G3R: Gradient Guided Generalizable Reconstruction. In *European Conference on Computer Vision*, pages 305–323. Springer, 2025.
- [11] 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.
- [12] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems*, 35:25683–25696, 2022.
- [13] Hyungjin Chung, Jeongsol Kim, Sehui Kim, and Jong Chul Ye. Parallel diffusion models of operator and image for blind inverse problems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6059–6069, 2023.
- [14] Hyungjin Chung, Dohoon Ryu, Michael T McCann, Marc L Klasky, and Jong Chul Ye. Solving 3D inverse problems using pre-trained 2D diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22542–22551, 2023.
- [15] Andrea Cohen, Torsten Sattler, and Marc Pollefeys. Merging the Unmatchable: Stitching Visually Disconnected SfM Models. In *ICCV*, 2015.
- [16] Andrea Cohen, Johannes L. Schönberger, Pablo Speciale, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. Indoor-Outdoor 3D Reconstruction Alignment. In *ECCV*, 2016.

- [17] 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.
- [18] 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.
- [19] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.
- [20] Frank Dellaert. Factor graphs and GTSAM: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012.
- [21] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- [22] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015.
- [23] P Gargallo, Y Kuang, et al. OpenSfM, 2016.
- [24] Henri P Gavin. The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems. *Department of Civil and Environmental Engineering Duke University August, 3*, 2019.
- [25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [26] Sepidehsadat Sepid Hossieni, Mohammad Amin Shabani, Saghar Irandoust, and Yasutaka Furukawa. PuzzleFusion: unleashing the power of diffusion models for spatial puzzle solving. *Advances in Neural Information Processing Systems*, 36, 2024.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] John W Lambert, Yuguang Li, Ivaylo Boyadzhiev, and Lambert E Wixson. Automated Building Floor Plan Generation Using Visual Data Of Multiple Building Images, May 4 2023. US Patent App. 17/585,433.
- [32] Yuguang Li. Automated Building Floor Plan Generation From Building Images Using A Combination Of Diffusion And Bundle Adjustment, February 12 2026. US Patent App. 19/295,540.
- [33] Yuguang Li. Automated Building Floor Plan Generation Using Transformer-Based Analysis Of Visual Data Of Building Images, February 12 2026. US Patent App. 19/295,544.

- [34] Yuguang Li, Grant Gingell, and Ivaylo Boyadzhiev. Automated tools for generating mapping information for buildings, February 8 2022. US Patent 11,243,656.
- [35] Yuguang Li, Ivaylo Boyadzhiev, and Lambert E Wixson. Automated tools for generating mapping information for buildings, June 18 2024. US Patent 12,014,120.
- [36] 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.
- [37] Yuguang Li, Will A Hutchcroft, and Sing Bing Kang. Automated inter-image analysis of multiple building images for building floor plan generation, July 1 2025. US Patent 12,347,033.
- [38] 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.
- [39] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.
- [40] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.
- [41] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *ICCV*, 2013.
- [42] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. OpenMVG: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*. Springer, 2016.
- [43] 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.
- [44] Dominik Muhle, Lukas Koestler, Krishna Murthy Jatavallabhula, and Daniel Cremers. Learning correspondence uncertainty via differentiable nonlinear least squares. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13102–13112, 2023.
- [45] Nelson Nauata, Kai-Hung Chang, Chin-Yi Cheng, Greg Mori, and Yasutaka Furukawa. House-GAN: Relational Generative Adversarial Networks for Graph-constrained House Layout Generation. In *ECCV*, 2020.
- [46] Nelson Nauata, Sepidehsadat Hosseini, Kai-Hung Chang, Hang Chu, Chin-Yi Cheng, and Yasutaka Furukawa. House-GAN++: Generative Adversarial Layout Refinement Network towards Intelligent Computational Agent for Professional Architects. In *CVPR*, 2021.
- [47] 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.
- [48] Rémi Pautrat, Iago Suárez, Yifan Yu, Marc Pollefeys, and Viktor Larsson. GlueStick: Robust image matching by sticking points and lines together. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9706–9716, 2023.
- [49] 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.

- [50] Luis Pineda, Taosha Fan, Maurizio Monge, Shobha Venkataraman, Paloma Sodhi, Ricky TQ Chen, Joseph Ortiz, Daniel DeTone, Austin Wang, Stuart Anderson, et al. Theseus: A library for differentiable nonlinear optimization. *Advances in Neural Information Processing Systems*, 35:3801–3818, 2022.
- [51] 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.
- [52] Alec Radford. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [53] 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.
- [54] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.
- [55] Grant Schindler and Frank Dellaert. Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE, 2004.
- [56] 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.
- [57] Mohammad Amin Shabani, Weilian Song, Makoto Odamaki, Hirochika Fujiki, and Yasutaka Furukawa. Extreme Structure From Motion for Indoor Panoramas Without Visual Overlaps. In *ICCV*, 2021.
- [58] Mohammad Amin Shabani, Weilian Song, Makoto Odamaki, Hirochika Fujiki, and Yasutaka Furukawa. Extreme structure from motion for indoor panoramas without visual overlaps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5703–5711, 2021.
- [59] Mohammad Amin Shabani, Sepidehsadat Hosseini, and Yasutaka Furukawa. House-Diffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5466–5475, 2023.
- [60] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [61] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.
- [62] Sinisa Stekovic, Mahdi Rad, Friedrich Fraundorfer, and Vincent Lepetit. Montefloor: Extending mcts for reconstructing accurate large-scale floor plans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16034–16043, 2021.
- [63] Jheng-Wei Su, Chi-Han Peng, Peter Wonka, and Hung-Kuo Chu. GPR-Net: Multi-view layout estimation via a geometry-aware panorama registration network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6469–6478, 2023.

- [64] 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.
- [65] 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.
- [66] 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, 2021.
- [67] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021.
- [68] Christopher Sweeney, Tobias Hollerer, and Matthew Turk. Theia: A fast and scalable structure-from-motion library. In *Proceedings of the 23rd ACM international conference on Multimedia*, 2015.
- [69] Mingxing Tan and Quoc Le. EfficientNetv2: Smaller models and faster training. In *International conference on machine learning*, pages 10096–10106. PMLR, 2021.
- [70] 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.
- [71] Chengzhou Tang and Ping Tan. BA-Net: Dense bundle adjustment network. *arXiv preprint arXiv:1806.04807*, 2018.
- [72] 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.
- [73] 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.
- [74] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, et al. WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 12, 2016.
- [75] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.
- [76] 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.
- [77] 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.
- [78] Jianyuan Wang, Christian Rupprecht, and David Novotny. PoseDiffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9773–9783, 2023.

- [79] 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.
- [80] 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.
- [81] Wenming Wu, Xiao-Ming Fu, Rui Tang, Yuhan Wang, Yu-Hao Qi, and Ligang Liu. Data-driven interior plan generation for residential buildings. *ACM Transactions on Graphics (TOG)*, 38(6):1–12, 2019.
- [82] 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.
- [83] 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.
- [84] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. PhysDiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16010–16021, 2023.
- [85] 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.
- [86] 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.
- [87] Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Truncated diffusion probabilistic models and diffusion-based adversarial auto-encoders. *arXiv preprint arXiv:2202.09671*, 2022.
- [88] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3D: A large photo-realistic dataset for structured 3D modeling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 519–535. Springer, 2020.
- [89] Chuhan 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.

A Appendix

A.1 Publication Notes

ZInD. This section is based on the Zillow Indoor Dataset paper [17], published in the main conference of CVPR 2021, on which I am the third author. The paper and dataset compilation were carried out by the broader Zillow team with interns. My primary contribution was leading the end-to-end development of Zillow’s floor-plan pipeline, spanning technical design and production implementation, which later scaled to support the generation of more than 0.2 million floor plans per year. The discussion here also reflects technical insights and practical lessons I developed through that work. In addition, I am an inventor on three first-author patents related to this subject area [34, 35, 37].

SALVe. This section is based on the *SALVe* paper, published in the main conference of ECCV 2022 [30], on which I am the second author. The work was carried out in collaboration with my intern, John Lambert. My main contributions were the key formulation that indoor image localization can be cast as a classification task using pose priors from semantic matches and monocular geometric predictions, and the initial implementation of the method prior to the neural network-based version. John Lambert led the design of the *SALVe* network module, the broader pipeline, and the experiments. This work also led to a U.S. patent [31].

PSMNet. The systems presented in this chapter, including *SALVe* and *PSMNet*, therefore serve not as final solutions but as diagnostic steps that reveal the limitations of partially coupled pipelines. In particular, they show that sparse floor-plan reconstruction requires tighter cross-view coupling between pose, geometry, and layout. Chapter 3 addresses this by introducing a unified feed-forward multi-view formulation for joint pose and layout reconstruction.

CovisPose. This section is based on the *CovisPose* paper, published in the main conference of ECCV 2022 [27], on which I am the second author. The work was primarily carried out by Will Hutchcroft, who developed the main architecture. My contributions included participating in the network and pipeline iterations throughout the project, implementing the baselines and evaluation pipeline, and contributing to the writing of the paper.

MvFFN. We plan to submit *MvFFN* paper to the main conference of NeurIPS 2026, which I’m the first author. There are several co-authors from University of Washington. But I came up with the main idea of objectives, architectures and the core ideas of *MOIS* reformulation, layout chaining post-processing, bundle adjustment integration, and inductive bias strategy. I also cleaned and prepared the ZInD dataset and did most of the paper writing. This work also leads to a relevant pending first-author patent application [33].

BADGR. This section is based on the *BADGR* paper, published in the main conference of CVPR 2025, on which I am the first author. The paper was selected as a Highlight, recognizing approximately 3% of submissions. I led the project end-to-end, including the

core method development, implementation, experiments, and writing, with support from my co-authors and mentors. We also have a pending patent application related to this work [32].

A.2 Implementation Details of *MvFFN* Experiments.

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.

A.3 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 [47, 36]. *VGGT* [79] does not operate directly on 360° panoramas, so we adapt it through perspective decomposition. We exclude *GraphCovis* [47] and *PanoPose* [73] 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 [47]: 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 [47, 36]. 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 [17]. 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.

A.4 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* [28] 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* [85] and *PolyRoom* [38] 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* [85] 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 $< 100 \text{ px}^2$ are discarded, and surviving corners are mapped back to metric floor plane coordinates using a precomputed homography.

PolyRoom. *PolyRoom* [38] uses the same rendered input representation, but adds a *Mask2Former* [11] 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 ($\epsilon = 5$ 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 $< 100 \text{ px}^2$, and map the surviving polygons back to metric floor-plane coordinates.

A.5 More Qualitative Results from *BADGR*

The reprojected wall lines in the image are drawn with a thickness equal to 1.1% of the image height. This thickness can cover significant floor distances, especially when walls are near the image center. For example, at a pitch angle of 30° (floor distance of 0.58 camera heights), the line covers 4.5% of the camera height; at 45° (1 camera height), it covers 7.4%; at 60° (1.73 camera height), 14.7%; and at 75° (3.73 camera height), 55.2%. While the blue and green lines represented in small images sometimes appear to overlap, particularly for walls farther from the camera, *BADGR* processes continuous inputs and outputs for coordinates, enabling higher precision. See quantitative results for more precise details.

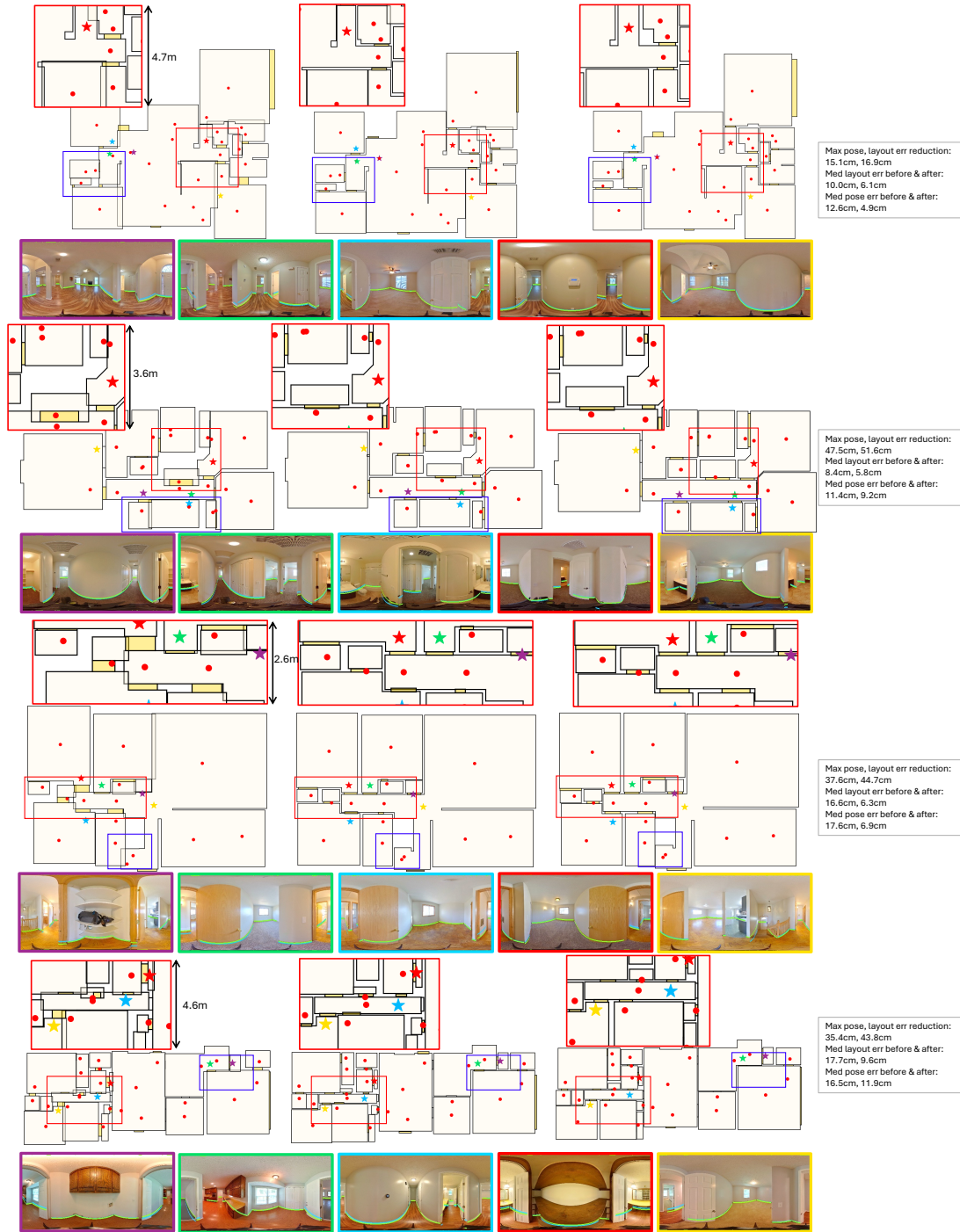


Figure A.1: More qualitative results trained on FloorPlan-60K dataset and tested on ZInD dataset (page 1), with input densities at a maximum of 2 input images from each input partial room. The topdown views from left to right are before, after *BADGR* optimization and the ground truth.

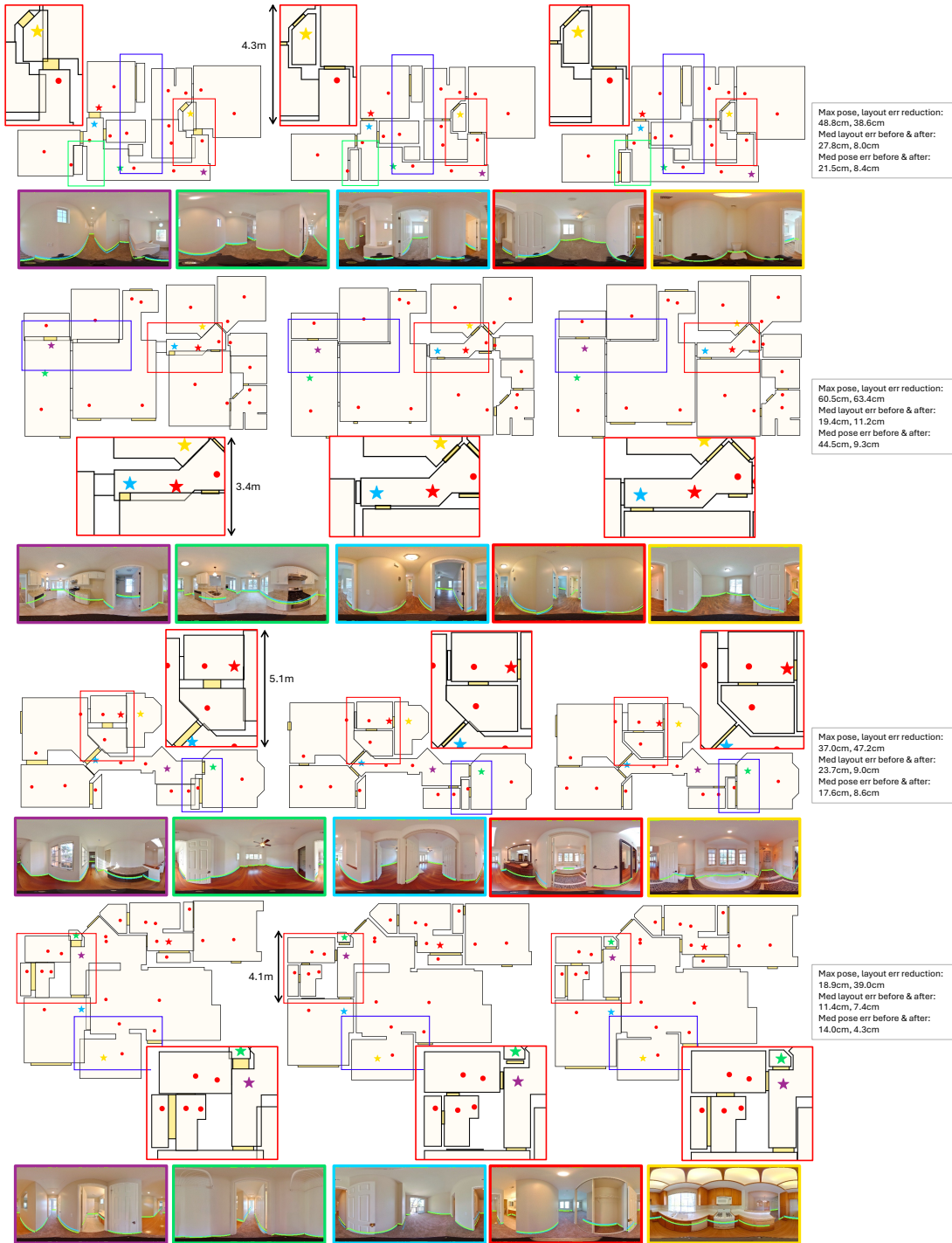


Figure A.2: More qualitative results trained on FloorPlan-60K dataset and tested on ZInD dataset (page 2), with input densities at a maximum of 2 input images from each partial room. The topdown views from left to right are before, after *BADGR* optimization and the ground truth.

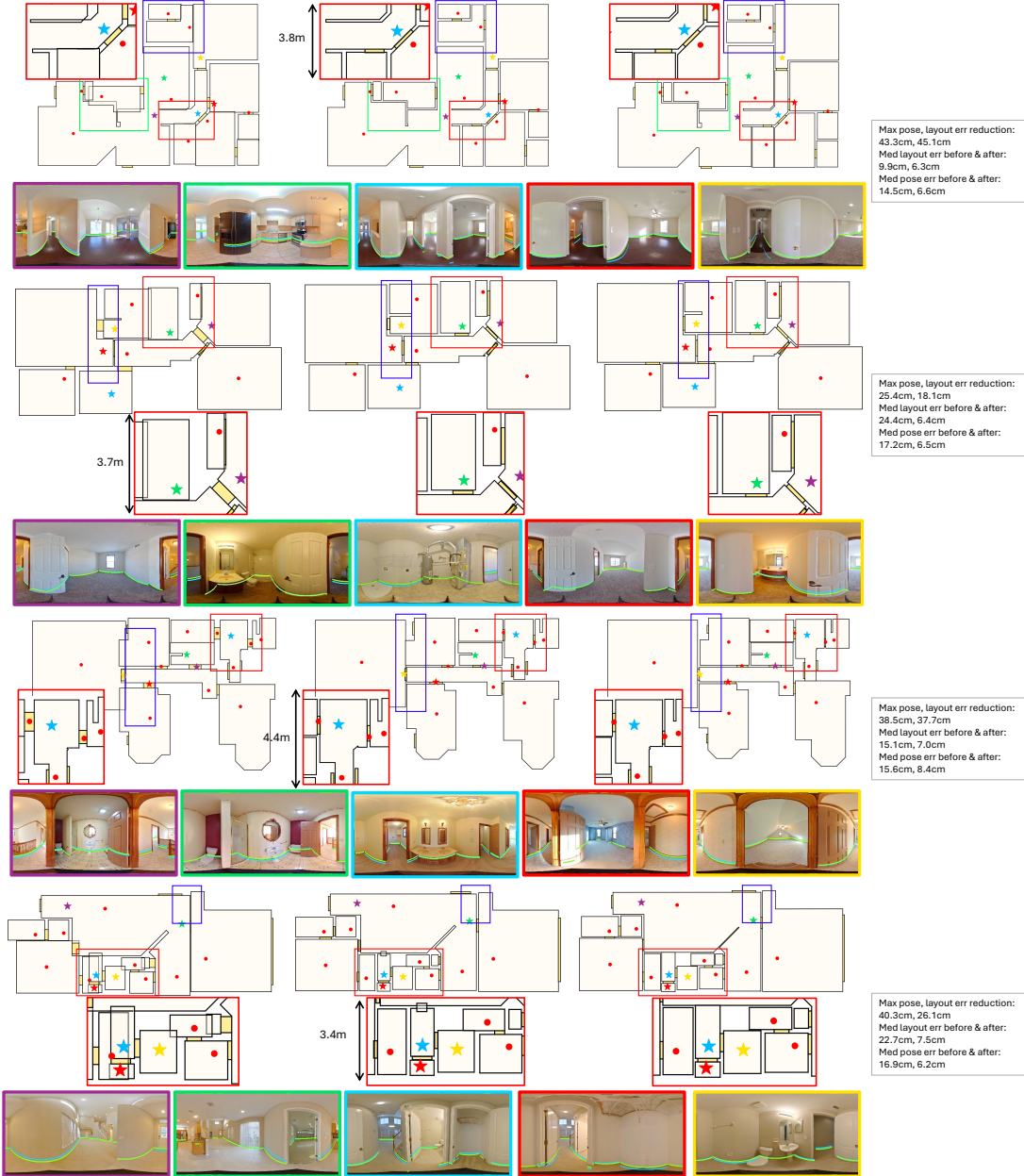


Figure A.3: More qualitative results trained on FloorPlan-60K dataset and tested on ZInD dataset (page 3), with input densities at a maximum of 1 input images from each partial room. The topdown views from left to right are before, after *BADGR* optimization and the ground truth.