JigsawPlan: Room Layout Jigsaw Puzzle Extreme Structure from Motion using Diffusion Models

Sepidehsadat Hosseini, Mohammad Amin Shabani, Saghar Irandoust, Yasutaka Furukawa
Simon Fraser University
{sepidh, mshabani, sirandou, furukawa}@sfu.ca

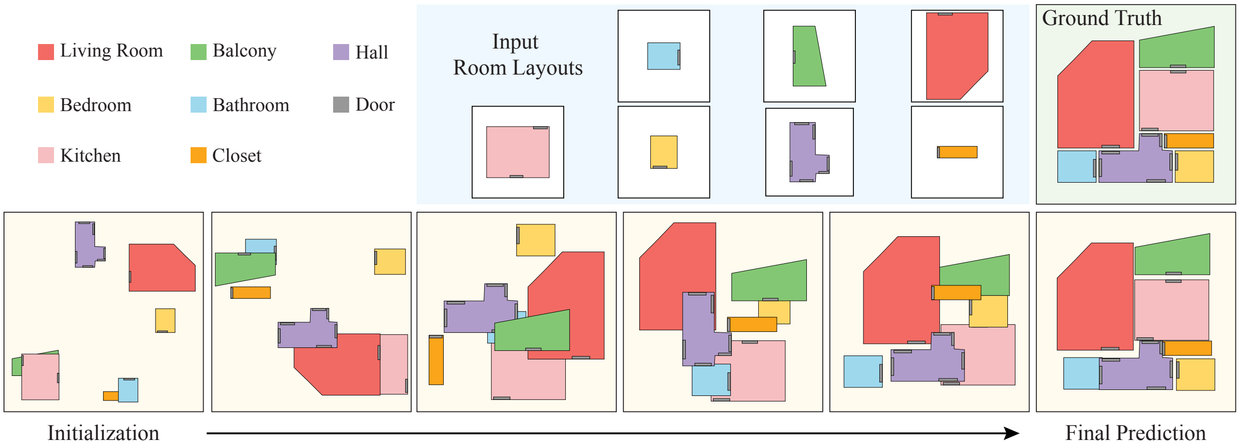

Figure 1. Extreme Structure from Motion is the task of taking a set of room layouts and their corresponding room types as the input and predicting the position and the orientation of each room. The biggest discovery and surprise of this paper is that conditional generation by a Diffusion Model solves this challenging pose estimation problem.

Abstract

This paper presents a novel approach to the Extreme Structure from Motion (E-SfM) problem, which takes a set of room layouts as polygonal curves in the top-down view, and aligns the room layout pieces by estimating their 2D translations and rotations, akin to solving the jigsaw puzzle of room layouts. The biggest discovery and surprise of the paper is that the simple use of a Diffusion Model solves this challenging registration problem as a conditional generation process. The paper presents a new dataset of room layouts and floorplans for 98,780 houses. The qualitative and quantitative evaluations demonstrate that the proposed approach outperforms the competing methods by significant margins. We will share all our code, models, and data.

1. Introduction

Mobile computers are becoming more powerful and equipped with better cameras every year. With the advancement of computing hardware, mobile indoor mapping applications with consumer-grade equipment are rising. Apple recently released a 3D indoor modeling app called RoomPlan. For 2D floorplan reconstruction specifically, many apps have been prevalent on the market (*e.g.*, AR Plan 3D, CubiCasa, MagicPlan, Ricoh, Zillow, and more). Structure from Motion (SfM) is a fundamental computer vision technology with a high impact on these applications.

Different from research-oriented projects [3, 19] or high-end commercial systems (*e.g.*, Matterport), these emerging mobile applications pose unique challenges to the current SfM techniques. The first challenge is inefficient image coverage. App users do not or are not motivated to follow complex instructions to take pictures. In many applications, only one image is available in each room, simply because the primal app objective is often a house image gallery, not floorplan reconstruction. Second, heavy computations with large neural networks are prohibitive on mobile devices. Large data upload (*e.g.*, a house tour movie) to the cloud is also prohibitive from mobile devices, which could further trigger privacy concerns. Third, the processing needs to be near real-time for interaction and verification.

This paper proposes a novel Extreme SfM (E-SfM) algorithm in these extremely challenging settings with a simple use of a Diffusion Model (DM) [9, 26, 28]. The biggest discovery and surprise of this paper is that conditional generation by DM solves a challenging E-SfM problem while overcoming all the above challenges, where DM is largely recognized as a generative model.

More concretely, our input is a set of room layouts as 1D polygonal curves with room types, which are highly compact (*i.e*., a few hundred bytes even for a large house with 6 to 7 rooms) and can be uploaded to the clouds easily for processing. The input data are obtained with a room layout estimation technique [35, 39] or user interactions with Augmented Reality (*e.g*., AR Plan 3D and MagicPlan). We train a diffusion model which generates the 2D translation and the 4-fold rotation (under the Manhattan assumption) for each room while specifying the input data as the condition, akin to solving a jigsaw puzzle of room layouts.

We have evaluated the proposed approach with a new dataset of room-layouts and floorplans for 98,780 houses from a production pipeline, which will be shared with the research community. We have also used the synthetic RPLAN dataset to assess the system's robustness across different datasets. Qualitative and quantitative evaluations show that the proposed approach significantly outperforms the existing and baseline methods.

2. Related Work

There are three approaches to the extreme pose estimation problems: feature matching, geometry inference, and arrangement learning.

Feature matching has been successful for the SfM problem [16, 18, 33]. The rise of deep neural networks enables more robust feature matching by learning [30,36,42]. However, these techniques require visual overlaps. Our task has little to no visual overlaps between adjacent images.

Geometry inference allows us to estimate a relative pose between images or partial scans with minimal overlaps by registering inferred or hallucinated geometry. A popular approach learns the priors of room shapes and alternates pairwise scan alignment and scene completion [17,40]. Yang *et al*. [41] combines global relative pose estimation and local pose refinement with panorama images. These techniques learn priors of a single room, while our approach learns the arrangements of multiple rooms in a house scale.

Arrangement learning is the current state-of-the-art for indoor E-SfM. The main idea is to use doors/windows to align two images or reconstructions that might not have any overlaps. An early work aligns indoor and outdoor reconstructions via windows [5], despite no learning on the arrangement itself. Shabani *et al*. [32] is the closest to ours, where they use doors to enumerate room arrangements and learn to score each candidate. Their approach is exponential in the number of rooms with many heuristics. Lambert *et al*. [12] uses doors, windows, and openings to create room alignment hypotheses. They utilize depth maps to create top-down views and learn to verify the correctness, which improves a run-time from exponential to polynomial. Our approach is end-to-end, does not enumerate arrangement candidates, and makes significant performance improvements. Lastly, while not aligning doors nor windows, an annotated site map and SfM reconstructions are aligned to solve a challenging structure from motion problem [20], which they coined as a "3d jigsaw puzzle".

Diffusion models (DMs) are emerging generative models, which slowly corrupt training samples by adding noise [6,9, 23], learn to invert the process, and generate a diverse set of samples from noise signals. DMs have established SOTA performances in numerous tasks such as image colorization/inpainitng [24,34], image to image translation [31,43], text to image [25], super-resolution [14, 27, 29], image and semantic editing [1, 21], and denoising [10]. Recent works use DMs as representation learners for discriminative tasks such as image segmentation [2,38]. However, to our knowledge, few works use DMs for non-generative tasks. We are the first to use DMs for challenging pose estimation.

3. Datasets

The paper provides a new E-SfM dataset, dubbed JigsawPlan, containing room layouts and floorplans for 98,780 single-story houses/apartments from a production pipeline (See Fig. 2). An Augmented Reality (AR) application creates room layouts by asking users to click room corners. The ground-truth arrangements of room layouts are manually specified, which this research seeks to automate. Note that the layouts are Manhattan-rectified as an enforcement of the AR app. A room is represented as a polygonal loop in the top-down view. A door is a line segment along the loop. Each room is associated with a room type (*e.g*., kitchen, living room, bedroom, toilet, closet, corridor, balcony, bathroom, den, etc.). The number of rooms in a house ranges from 3 to 10. Concretely, 11661, 16322, 19171, 17582, 13200, 9649, 6780, and 4415 houses contain 3, 4, 5, 6, 7, 8, 9, and 10 rooms, respectively. The minimum and maximum numbers of corners in a house are 12 and 182.

Our second E-SfM benchmark is based on a synthetic floorplan dataset RPLAN to demonstrate robustness across different datasets. RPLAN contains 60K vector floorplans made by architects. We take a floorplan and save each room shape with incident doors as a task input. The number of rooms in a house ranges from 3 to 8. The number of corners ranges from 14 to 98. See the supplementary for more details on this benchmark. Note that RICOH dataset [29] and ZIND dataset [13] are too small for network training and are not used in our experiments.

4. Problem Definition

We borrow the problem formulation by Shabani *et al*. [32] with minor adaptions to the input and the metrics:

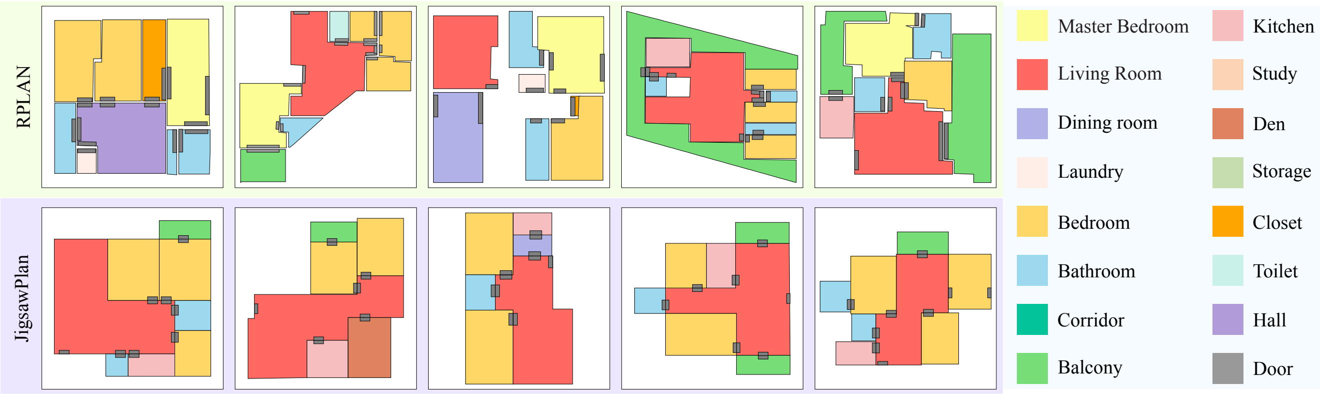

Figure 2. Dataset samples. JigsawPlan is our new dataset from a production pipeline. RPLAN is an existing synthetic floorplan dataset.

Input: Their original input was a set of panorama images, where a room layout estimation [35] and an object detection network were used to convert the panoramas to top-down semantic images. In our datasets, room layouts do not come from panoramas but from an interactive AR application. Therefore, we define our input to be a set of N room shapes, each of which is a sequence of room-corner coordinates forming a 1D polygonal loop in the top-down view. A door is a line segment consisting of two door corners. For simplicity, we mix room-corners and door-corners, and use C^r_i to denote the 2D coordinate of the i_{th} corner in the r_{th} room. In order to remove the ground-truth position information, the mean room-corner coordinate is subtracted from all the corner coordinates for each room. A room is associated with a room type, which is represented as a one-hot vector $\mathcal{T}^r$.

Output: The output is the position of the room center and the 4-fold rotation under the Manhattan assumption for each room. The rotation is around the room center, which is the average of the room corner coordinates.

Metrics: Our quantitative evaluations use two metrics. The first metric is the Mean Positional Error in pixels (MPE) over the rooms. [1] The second metric evaluates the correctness of the room connectivity in the reconstruction. We borrow a Graph Edit Distance (GED) metric by Nauata *et al.* [22], which counts the number of user edits that are necessary to fix the connectivity graph. We declare that two rooms are connected if the centers of the two doors (from the two rooms) are within 5 pixels.

Coordinate scaling: Our benchmarks are constructed from the ground-truth floorplans whose scales were normalized, so that entire floorplans fit inside a 256 × 256 square. This could allow cheating because the longer extent of arranged floorplans is always 255 pixels. Therefore, at test time, we scale the room shapes by a random scaling factor in the range of [0.8, 1.0] for each house.

5. Method

Our idea is simple, using a Diffusion Model to "conditionally generate" a room-layout arrangement, where the input room layouts and types are the conditions. This section explains the forward and the reverse processes.

5.1. Forward process

The forward process adds a Gaussian noise to a room-layout arrangement. A compact representation would be per-room positions and rotations. Instead, we will use a redundant representation, where a room center position and a 4-fold rotation are stored at each room/door corner. There are a few reasons. Our reverse process is based on a Transformer architecture where each position/rotation estimation becomes a node. Our approach 1) enriches the capacity of the arrangement representation (also an adaptive capacity, where a complex room with more corners is given more capacity); 2) allows direct communications between doors for which we will have a door-specific loss; and 3) makes it straightforward to combine with the condition (*i.e.*, original corner coordinates and room types).

Concretely, we use $x^r_{i,t}$ to denote the position/rotation of the r_{th} room stored at the i_{th} corner at time t of the diffusion process, where t varies from 0 to 1,000 in our experiments:

$$x^r_{i,t} \quad = \quad \left(p^r_{i,t}, o^r_{i,t}\right). \tag{1}$$

$p^r_{i,t}$ and $o^r_{i,t}$ denote the room-center position (a 2D vector) and the 4-fold rotation (a 4D one-hot vector). The forward process adds a noise by sampling $\delta^r_{i,t} \in \mathcal{N}(\mathbf{0}, \mathbf{I})$ with a standard linear noise scheduling with variance $(1-\alpha_t)$ [9]:

$$x^r_{i,t} = \sqrt{\bar{\alpha}_t} x^r_{i,0} + \sqrt{1-\bar{\alpha}_t}\delta^r_{i,t},, \quad \bar{\alpha}_t := \prod_{i=1}^{t} \alpha_i. \tag{2}$$

[1] Shabani et al. [32] used a more permissive metric (the availability of the "correct" solution in the k results with a certain error-tolerance) as the task was too challenging. This work significantly boosts the performance and uses a standard SfM metric.

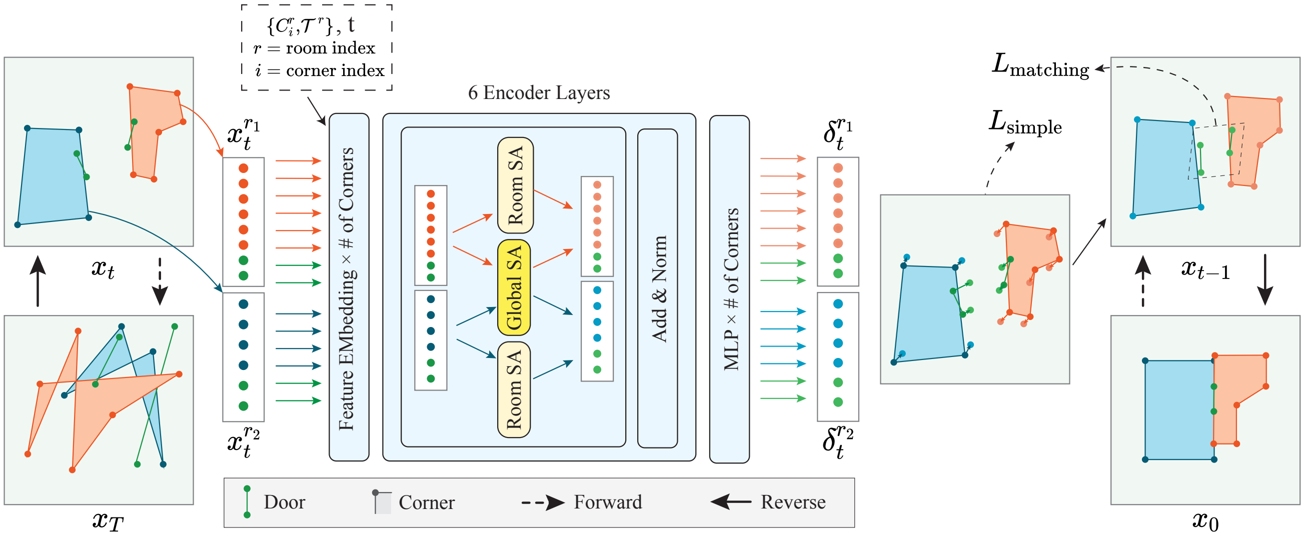

Figure 3. A diffusion model architecture for Extreme Structure from Motion. Given the arrangement estimation $x_t = \{x^r_{i,t}\}$, the reverse process infers the noise $\{\delta^r_{i,t}\}$ and recovers $x_{t-1} = \{x^r_{i,t-1}\}$, while injecting the original room shapes $\{C^r_i\}$ and types $\{\mathcal{T}^r\}$ as the condition. Each room corner holds the room position and rotation estimation. The reverse process starts from x_T and denoises towards x_0.

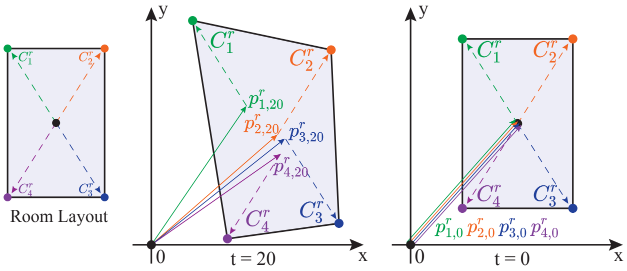

Figure 4. Each room/door corner holds the room position $p^r_{i,t}$ and the rotation $o^r_{i,t}$ (not visualized here) estimation. At time $t = 0$ before noise injection, they share the same ground-truth values per room. C^r_i denotes the original room corner coordinates with respect to the room center. $p^r_{i,t} + C^r_i$ indicates the estimated position of the i_{th} corner of the r_{th} room at time t.

5.2. Reverse process

Figure 3 illustrates our reverse process, which takes the arrangement $\{x^r_{i,t}\}$ at time t and infers the noise $\widetilde{\delta^r}_{i,t}$ under the condition of the original room shapes $\{C^r_i\}$ (*i.e.*, a corner position with respect to the room center) and the room types as a one-hot vector $\{\mathcal{T}^r\}$.

Feature embedding: The reverse process is based on a Transformer architecture where every corner is a node (See Fig. 4). We initialize its feature embedding as

$$\hat{x}^r_{i,t} \leftarrow \mathrm{Lin}(x^r_{i,t}) + \mathrm{Lin}([C^r_i, \mathcal{T}^r]) + \mathrm{MLP}(t) + \mathrm{PE}(i'). \quad (3)$$

The first term uses a linear layer to convert a 6d vector (*i.e.*, 2 for the room center coordinate and 4 for the rotation one-hot vector) to a 128d embedding vector. The second term also uses a linear layer to convert a 28d condition vector (*i.e.*, 2 for the original corner coordinate and 26 for the room/door type one-hot vector) to the same dimension. The third term uses a 2-layer MLP to convert a time step t to a 128d vector. The fourth term is a standard frequency position encoding [37] of a corner index i'. Note that this corner index is across all rooms instead of one room, where room orders are random. An alternative way is to use linear layers to embed a room index and a corner index. Our index mixes rooms and does not indicate which corners belong to the same room, but is free from any network parameters to learn and works well in practice. Our transformer architecture injects corner-to-room association information by structured self-attentions next.

Attention modules: Feature embeddings $\{\hat{x}^r_{i,t}\}$ go through six blocks of self-attention modules that have two different attention mechanisms: Room Self Attention (R-SA) and Global Self Attention (G-SA). R-SA limits pairwise interactions between corners in each room. R-SA is akin to a sparse self attention family [4, 8, 15], which helps to generate consistent positions and rotations at different corners of a room. G-SA is a standard self-attention between all corners of a house. After the attention blocks, a linear layer converts 128d embedding back to a 6d representation $\widetilde{\delta^r}_{i,t}$, which is used for the following denoising formula [9]:

$$x^r_{i,t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x^r_{i,t} - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\tilde{\delta}^r_{i,t}\right) + \sqrt{1-\alpha_t}z. \quad (4)$$

$z \sim \mathcal{N}(0, I)$ for $t > 1$ and otherwise 0. For the final result at time $t = 0$, we take the average room center position and the room rotation (in fact, the majority vote after picking the Manhattan-rotation with the highest score at each corner).

Loss functions: There are two loss functions. First, we follow [6, 9] and use a standard noise regression loss on δ:

$$L_{\text{simple}} = E_{t, x^r_{i,0}, \delta^r_{i,t}} \left[\left\| \delta^r_{i,t} - \tilde{\delta}^r_{i,t} \right\|^2 \right]. \tag{5}$$

The second loss is on door corners, each of which is shared by two rooms. Let us use r_1 and r_2 to denote its room indexes. i_1 and i_2 denote the corresponding door-corners indexes in r_1 and r_2. The corresponding door corners must be at the same position and the loss is the L2 distance of their positions:

$$\begin{aligned} L_{\text{match}} &= E_{t, x^r_{i,0}, \delta^r_{i,t}} \left[\left\| \hat{C}^{r_1}_{i_1} - \hat{C}^{r_2}_{i_2} \right\|^2 \right], && (6) \\ \hat{C}^r_i &= \tilde{p}^r_{i,0} + R_{\tilde{o}^r_{i,0}} C^r_i, && (7) \\ (\tilde{p}^r_{i,0}, \tilde{o}^r_{i,0}) &= \tilde{x}^r_{i,0} = \left(x^r_{i,t} - \sqrt{1 - \bar{\alpha}_t} \tilde{\delta}^r_{i,t} \right) / \sqrt{\bar{\alpha}_t}. && (8) \end{aligned}$$

$R_{\tilde{o}^r_{i,0}}$ denotes a Manhattan rotation matrix corresponding to the largest entry in $\tilde{o}^r_{i,0}$. Note that obtaining the rotation matrix R is not differentiable and the gradients are not propagated. The rotation branch is only trained with the first noise regression loss. The total loss is defined as $L_{\text{total}} = L_{\text{simple}} + 0.01 \cdot L_{\text{match}}$.

6. Experimental Results and Discussions

We have implemented the system with PyTorch 1.12.1 and Python 3.9.13, using a workstation with a 3.70GHz Intel i9-10900X CPU (20 cores) and two NVIDIA RTX A6000 GPUs. We use the AdamW optimizer with b1=0.9, b2=0.999, weight_decay=0.05, and batch_size=512. The learning rate is initialized to 0.0003. We use a step-learning scheduler with a step size of 20 epochs. The pose estimation process is stochastic and we run our system 5 times and report the mean and the standard deviation. It takes roughly 24 hours to train a model and 3 seconds to estimate one arrangement for one house.

6.1. Competing methods

Shabani *et al.* [32] is the existing state-of-the-art, which enumerates arrangement candidates and learns to classify a layout candidate to be realistic or not. We use their public implementation[2], where RPLAN and JigsawPlan datasets have a different number of room/door types, and we made minor modifications to the data-loader and the network architecture. We refer details of the baseline systems to the supplementary.

Transformer with a raster representation (TransRaster) uses the raster images to represent the input room layouts/types and the output room positions. Note that this baseline does not handle rotations as explained below. An input room layout is represented as a 25-channel 256 × 256 semantic segmentation image, where there are 25 room/door types. The room center is aligned with the center of an image. An output room position is represented as a 256 × 256 room occupancy image, which is ideally a translated version of the input room segmentation image at the correct room location. Given an output room occupancy image, we perform an exhaustive search over the possible room translations and find one with the most overlap between the occupancy image and the translated room segmentation image. [3] We use VisionTransformer [7] with a CNN decoder that takes a set of input room segmentation images and produces a set of room occupancy images.

Transformer with a vector representation (TransVector) is the third baseline, which uses our transformer network module at the core without the Diffusion. The baseline directly estimates the pose parameters without iterations, and we make the following changes to our architecture: 1) Remove time-dependent features ($x^r_{i,t}$ and t) from the embedding (3); 2) Change the supervision $\delta^r_{i,t}$ from the noise to the position/rotation parameters; and 3) Use $\delta^r_{i,t}$ instead of $\hat{C}^r_i$ in the door matching loss (6).

6.2. Main results

Table 1 compares the Mean Positional Error and the Graph Editing Distance for the four methods (ours and the three competing methods). TransRaster does not handle rotations, and the first group of four rows shows a case when room layouts are fixed to the ground-truth orientations. We modify the architecture and remove features and the loss function related to the rotation. The second group of three rows shows the full pose estimation results where inferring both the positions and the rotations, room rotations are randomly initialized for each room. The left and the right of the table show results for RPLAN and JigsawPlan datasets, respectively. Since the run-time of Shabani *et al.* [32] is exponential in the number of rooms, it takes hours or even days to process a single house with seven or more rooms. Therefore, we extract small houses (*i.e.*, at most six rooms) and create “Small RPLAN” and “Small JigsawPlan” datasets for separate evaluations. For each experimental setting (*e.g.*, Small JigsawPlan without rotation), we train a network for each method. Shabani *et al.* is evaluated only with Small RPLAN and Small JigsawPlan.

Our method outperforms all the baselines in all the settings and metrics, except the GED metric for Small Jigsaw-

[2] https://github.com/aminshabani/extreme-indoor-sfm

[3] We could expand the search space with possible room rotations, but rooms are often symmetric. To be simple, we use this baseline only for experiments when ground-truth rotations are given.

Table 1. Main quantitative results with two metrics: Positional Error (MPE) and Graph Editing Distance (GED). A group of four rows in the middle show a case where the ground-truth rotations are given. A group of three rows at the bottom show a case where both the positions and the rotations are estimated, where TransRaster baseline cannot handle rotations and is not evaluated. Small RPLAN (resp. Small JigsawPlan) is a subset of the corresponding full dataset, consisting of houses with at most 6 rooms. The small datasets are created for Shabani *et al.*, which is not scalable to many rooms. Our method is stochastic and shows both the mean and the standard deviation.

Dataset	Small RPLAN		Full RPLAN		Small JigsawPlan		Full JigsawPlan	
Metric	MPE (↓)	GED (↓)	MPE (↓)	GED (↓)	MPE (↓)	GED (↓)	MPE (↓)	GED (↓)
Shabani *et al.*	17.64	1.02	✗	✗	32.24	1.10	✗	✗
TransRaster	13.92	1.16	15.72	2.11	36.06	2.11	41.93	4.12
TransVector	12.87	1.08	13.97	1.99	37.72	1.93	42.85	4.03
Ours	**4.74**±0.64	**0.40**±0.00	**5.34**±0.81	**0.62**±0.01	**17.71**±0.81	**1.05**±0.44	**28.25**±0.70	**2.67**±0.47
Shabani *et al.*	29.44	1.28	✗	✗	36.63	**1.89**	✗	✗
TransVector	36.09	1.51	46.18	2.27	40.80	2.38	53.11	6.41
Ours	**8.70**±1.0	**0.96**±0.02	**10.65**±1.2	**0.98** ±0.04	**34.22**±0.97	1.96±0.53	**41.23**±0.85	**3.16**±0.57

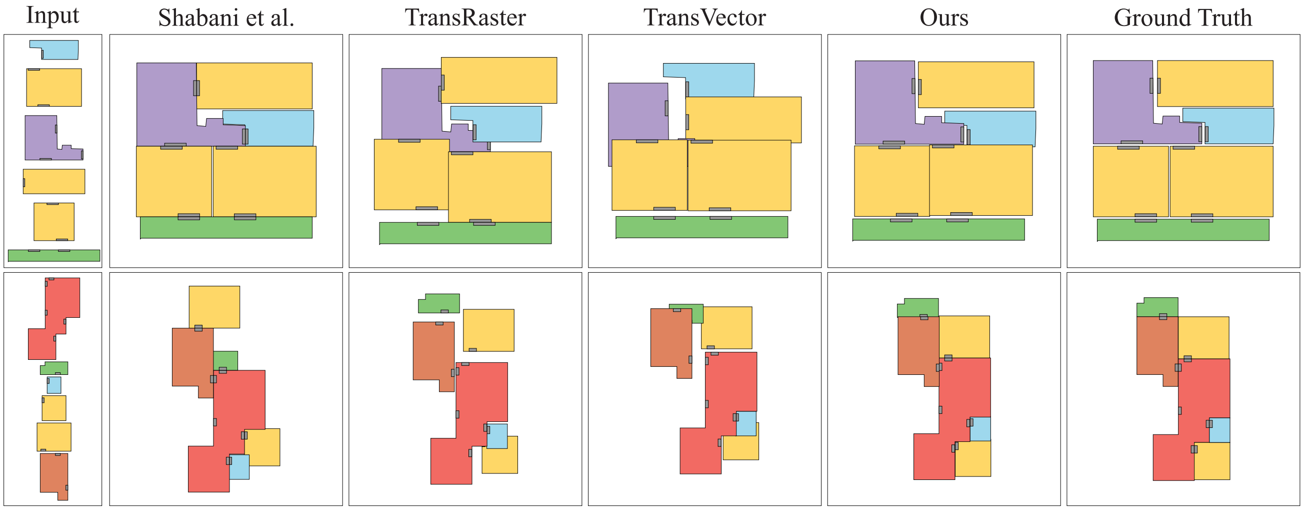

Figure 5. Qualitative evaluations of our approach against the three competing methods. The top two rows are from Small JigsawPlan. The bottom row is from Small RPLAN. The GT rotations are given for all the cases to enable comparisons with all the methods.

Plan, where the existing state-of-the-art Shabani *et al.* [32] achieves a slightly better score. Their method enumerates all possible candidate arrangements by matching doors and evaluates the realism of each arrangement one by one. Their run-time is exponential in the number of rooms, and the system involves many heuristics.

TransRaster and TransVector are end-to-end and achieve comparable performance with Shabani *et al.*; in fact, a much better MPE score for Small RPLAN. However, our system performs much better in every metric, demonstrating the power of Diffusion Models even for non-generative tasks, in our case, pose estimation.

Figure 5 shows qualitative comparisons of all methods for the "Small" datasets with the given ground-truth rotations. Note that Shabani *et al.* cannot handle Full datasets, and TransRaster requires ground-truth rotations. This is the easiest setting (*i.e.*, small houses with ground-truth rotations) where the competing methods produce reasonable results. Our approach produces the best, especially the middle example, where the arrangement of the three rooms on the right (blue, purple, and ping) is challenging. Figure 6 compares Shabani *et al.* and ours again on the small datasets but without the ground-truth rotations. Figure 9 shows our results for the most challenging setting (*i.e.*, Full JigsawPlan dataset without ground-truth rotations), where only our method produces reasonable results. Our failures are often attributed to 1) Rare building architecture (top-left of Fig. 9) and 2) Inherent ambiguity (top-left of Fig. 9), whose tasks are challenging even for humans.

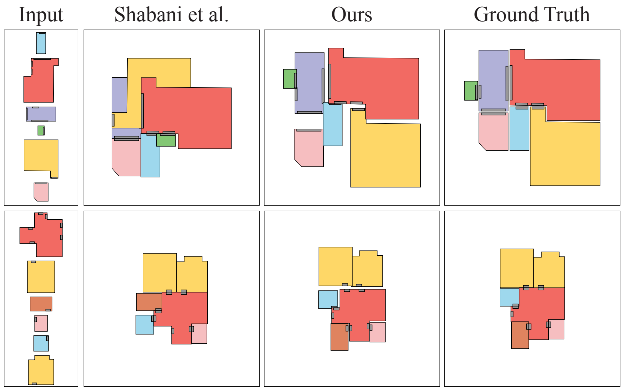

Figure 6. Qualitative evaluations when the GT rotations are not given. The top row is from Small JigsawPlan. The bottom row is from Small RPLAN.

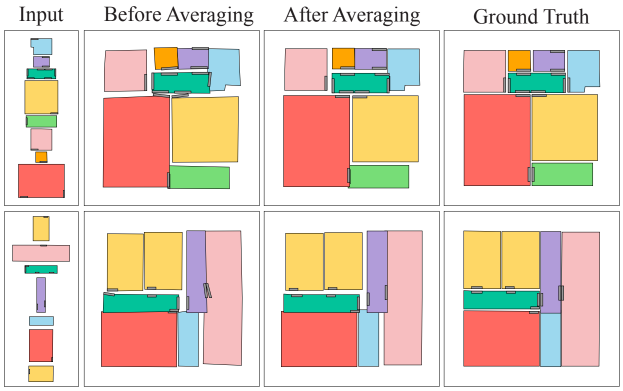

Figure 7. The final room arrangement before and after averaging. Our diffusion model estimates a room position/rotation at each room corner, which may not be consistent in a room. The final arrangement is obtained by taking the average position (the majority vote for rotation) within each room. The second (resp. third) column shows the results before (resp. after) the averaging.

Table 2. Contributions of our two attention mechanisms (R-SA, G-SA) and the door matching loss (L_{match}). Full JigsawPlan without the GT rotations are used for the evaluations. ✓indicates the feature being used.

R-SA	G-SA	L_{match}	MPE (↓)	GED (↓)
	✓		49.99±0.91	5.00±0.92
✓	✓		44.84±1.49	4.69±0.30
✓		✓	57.66±1.52	9.05±0.62
	✓	✓	44.29±0.76	4.37±0.80
✓	✓	✓	41.23±0.85	3.16±0.57

Table 3. Effects of the room-type (R-type) and the Door information. Full JigsawPlan without the GT rotations is used. ✓indicates the information being used. When a room-type is not used, we set a zero vector as a room-type one-hot vector. When the door information is not used, we do not pass the door-corner nodes to the network. Note that GED cannot be computed when the door information is removed.

Train		Test		MPE(↓)	GED (↓)
R-Type	Door	R-Type	Door		
✓	✓		✓	49.79	3.73
	✓		✓	48.38	3.52
✓	✓	✓		47.21	N/A
✓		✓		47.05	N/A
✓	✓	✓	✓	41.23	3.16

6.3. Ablation studies

We conduct a few ablation studies to further demonstrate the effectiveness of our ideas. Table 2 shows the contributions of our two attention mechanisms (R-SA, G-SA) and the door matching loss. G-SA provides communications between every pair of corners in a house, and its removal has the most impact on the performance. Removing R-SA or the matching loss also leads to a significant drop in both MPE and GED metrics. Our method with all the components achieves the highest performance.

Our layout arrangement uses a redundant representation, enriching the capacity and enabling direct communications between room/door corners (Sect. 5.1). Figure 7 shows the raw estimated position information at each room/door corner before the room-wise averaging. Since the ground-truth has the same pose parameters for all corners in a room/door, the network learns to produce consistent parameters. To assess the effects of the redundant representation, we create a variant of our system with the compact representation, that is, each room/door has only a single node estimating a single copy of the room position and the rotation. We aggregate corner coordinates into a single embedding vector and pass as a condition (See supplementary for the details). The MPE/GED metrics for Full JigsawPlan without ground-truth rotations change from (41.23/3.16) to (51.62/5.52), a significant performance drop.

A popular approach to indoor Extreme Structure-from-Motion is to align door detections/annotations to enumerate arrangement candidates [12, 32]. Such approaches are susceptible to errors in the door information. Our approach trains a network to infer the room arrangements instead of using heuristics to enumerate them, and is robust to input errors such as missing doors or incorrect room type annotations, which often happen in real world applications. To demonstrate the robustness of our approach, we remove the room type and the door information during training and or testing, and measure the performance drops in Table 3. While the performance does drop, the effects are marginal. Our numbers are still much better than TransVector without any data corruption (MPE=53.11 and GED=6.41 in Ta-

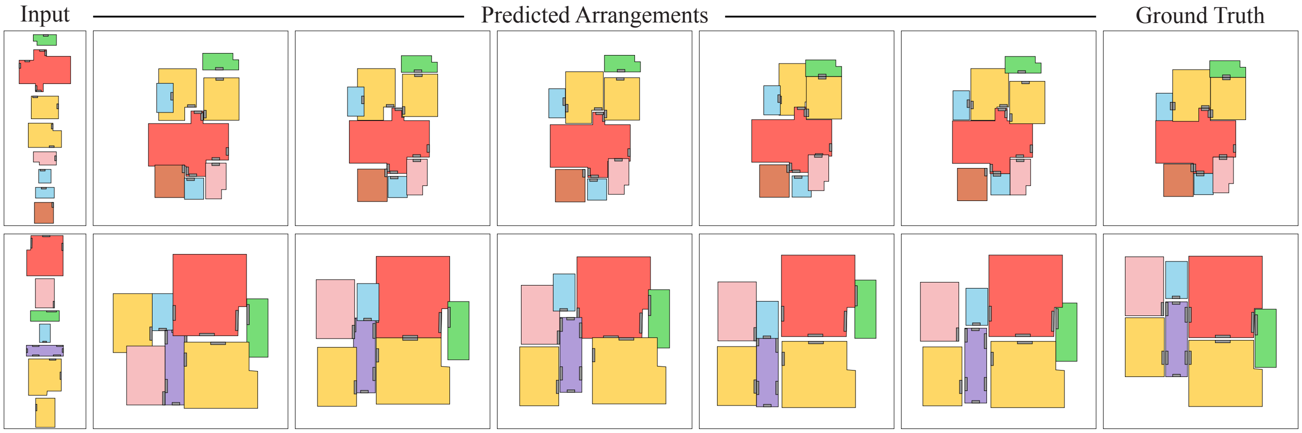

Figure 8. A diffusion model is stochastic and produces a different result every time. The middle rows show five different pose estimation results. The top (resp. bottom) is from Full RPLAN (resp. Full JigsawPlan) dataset without GT rotations.

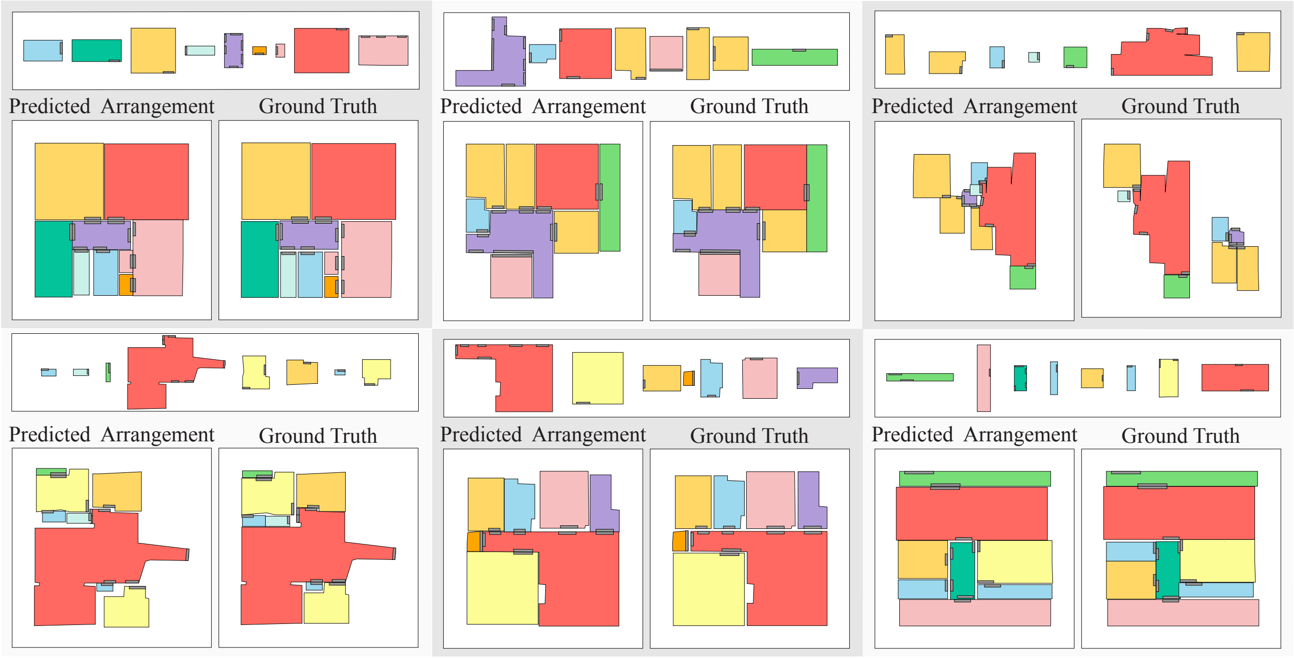

Figure 9. Qualitative evaluations of our method for Full JigsawPlan dataset without GT rotations.

ble 1), which was the only competing method capable of handling this most challenging setting. See supplementary for the same experiment on RPLAN dataset.

Lastly, Figure 8 shows five pose estimation results by our system while varying the initial noise x_T. While there are minor differences, the overall room arrangements are similar and close to the ground-truth, indicating that the Diffusion model is capable of producing consistent results given enough constraints as a pose estimation system, as opposed to a generative model whose original goal is to create a diverse set of answers. Please see the supplementary document and the video for more results and visualizations.

6.4. Future work

The proposed approach is faster, robust to data corruptions, end-to-end, and far superior to existing methods in all the metrics. However, the end-to-end design with a powerful neural architecture requires more training data, prohibiting us from processing some other smaller datasets. One future work is the integration with more data efficient neural architecture [11]. Another future work is the utilization of image information [12,32], while minimizing the amount of data transfer by utilizing on-device processing.

Acknowledgement: This research is partially supported by NSERC Discovery Grants with Accelerator Supplements and DND/NSERC Discovery Grant Supplement.

References

[1] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *arXiv preprint arXiv:2206.02779*, 2022. 2

[2] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models, 2021. 2

[3] 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. 1

[4] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 4

[5] Andrea Cohen, Johannes L. Schönberger, Pablo Speciale, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. Indoor-outdoor 3d reconstruction alignment. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 285–300, Cham, 2016. Springer International Publishing. 2

[6] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *CoRR*, abs/2105.05233, 2021. 2, 5

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. 5

[8] Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. Star-transformer. *arXiv preprint arXiv:1902.09113*, 2019. 4

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020. 1, 2, 3, 4, 5

[10] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*, 2022. 2

[11] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020. 8

[12] 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 *ECCV*, 2022. 2, 7, 8

[13] 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 *ECCV*, 2022. 2

[14] Haoying Li, Yifan Yang, Meng Chang, Huajun Feng, Zhihai Xu, Qi Li, and Yue-ting Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *CoRR*, abs/2104.14951, 2021. 2

[15] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, 32, 2019. 4

[16] Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-resolution correspondence networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. 2

[17] Cheng Lin, Changjian Li, and Wenping Wang. Floorplan-jigsaw: Jointly estimating scene layout and aligning partial scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5674–5683, 2019. 2

[18] Wen-Yan Lin, Siying Liu, Nianjuan Jiang, Minh N. Do, Ping Tan, and Jiangbo Lu. Repmatch: Robust feature matching and pose for reconstructing modern cities. In *ECCV*, 2016. 2

[19] Chen Liu, Jiaye Wu, and Yasutaka Furukawa. Floornet: A unified framework for floorplan reconstruction from 3d scans. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 201–217, 2018. 1

[20] Ricardo Martin-Brualla, Yanling He, Bryan C Russell, and Steven M Seitz. The 3d jigsaw puzzle: Mapping large indoor spaces. In *European Conference on Computer Vision*, pages 1–16. Springer, 2014. 2

[21] 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. In *International Conference on Learning Representations*, 2022. 2

[22] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13632–13641, 2021. 3

[23] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *CoRR*, abs/2102.09672, 2021. 2

[24] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. *CoRR*, abs/2112.10741, 2021. 2

[25] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[26] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *CoRR*, abs/2102.12092, 2021. 1

[27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 2

[28] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. 1

[29] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *arXiv:2104.07636*, 2021. 2

[30] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2

[31] Hiroshi Sasaki, Chris G. Willcocks, and Toby P. Breckon. UNIT-DDPM: unpaired image translation with denoising diffusion probabilistic models. *CoRR*, abs/2104.05358, 2021. 2

[32] 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 (ICCV)*, October 2021. 2, 3, 5, 6, 7, 8

[33] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *ACM siggraph*, pages 835–846, 2006. 2

[34] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *CoRR*, abs/2011.13456, 2020. 2

[35] Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1047–1056, 2019. 2, 3

[36] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021. 2

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[38] Julia Wolleb, Robin Sandkühler, Florentin Bieder, Philippe Valmaggia, and Philippe C. Cattin. Diffusion models for implicit image segmentation ensembles, 2021. 2

[39] 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 *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, pages 3363–3372, 2019. 2

[40] Zhenpei Yang, Jeffrey Z. Pan, Linjie Luo, Xiaowei Zhou, Kristen Grauman, and Qixing Huang. Extreme relative pose estimation for rgb-d scans via scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[41] Zhenpei Yang, Siming Yan, and Qixing Huang. Extreme relative pose network under hybrid representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2455–2464, 2020. 2

[42] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2666–2674, 2018. 2

[43] Min Zhao, Fan Bao, Chongxuan Li, and Jun Zhu. Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations. *arXiv preprint arXiv:2207.06635*, 2022. 2