PuzzleFusion: Unleashing the Power of Diffusion Models for Spatial Puzzle Solving [Supplementary Material]

Anonymous Author(s)
Affiliation
Address
email

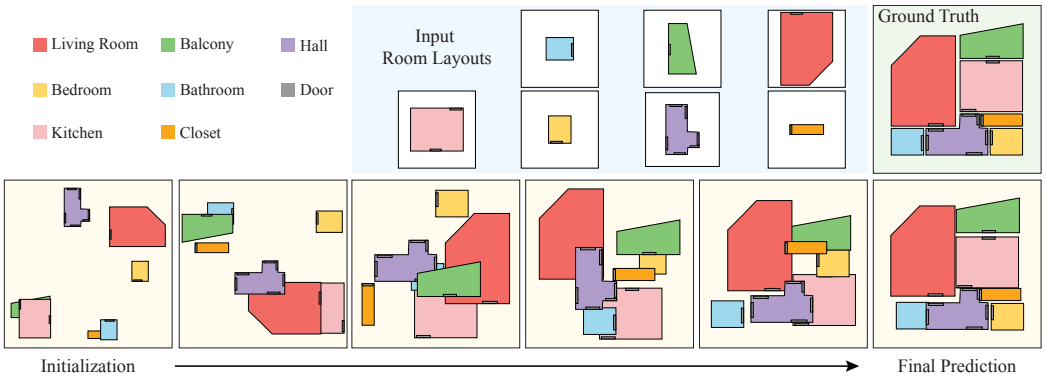

Figure 1: Room layout arrangement is the task of taking a set of room layouts and their corresponding room types as the input and predicting the position and the orientation of each room. The biggest discovery and surprise of this paper is that conditional generation by a Diffusion Model solves this challenging problem.

The supplementary document provides more details on our system and the competing methods (Sect. A), more details on the datasets (Sect. B), additional ablation studies (Sect. C), and additional qualitative examples (Figs. 2, and 4, 3, 8, 9) as promised in the main paper. Figure 2 visualizes the samples predicted by our method at step t. Figure 4 shows more qualitative evaluations of our method for Full MagicPlan and Full RPLAN datasets. Figure 3 shows more qualitative evaluations of our approach against the three competing methods. Please also see the supplementary video for more examples.

A Methods details of our system and competing methods

We benefit from Transformers in our task in two ways. First, Transformers provide the capability of processing sequences with different lengths, which we use to process different number of room layouts/corners in the houses. Second, we utilize the self-attention module of Transformers to create optimal interaction and information-sharing among input tokens. These two features make Transformers an ideal backbone for our model. Our method uses six Transformer encoder blocks, and attention in each block has four heads. We also use an MLP For converting 256D Transformer output to rotation and position (4D). To keep the experiments fair, we use the same architecture for our transformer baselines as much as possible. In the following, we provide details corresponding to each of the baselines.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

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 20-channel 256×256 semantic segmentation image, where there are 20 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. [1] We use VisionTransformer [1] with a CNN decoder that takes a set of input room segmentation images and produces a set of room occupancy images.

In the other word TransRaster uses an encoder part of U-Net, which has 8 down-sampling blocks, converting each input room layout to a feature map of dimension 512. Each feature map (corresponding to a room layout) will become one input token for the Transformer. Input sequence's length is equal to the number of rooms in a house, and information is shared among different rooms. We use six Transformer encoder blocks, and attention in each block has four heads. We pass the output of Transformer to a U-Net up-sampling model with eight Up-sampling blocks to change the dimension from 512 to 256×256.

Transformer with a vector representation uses the same backbone as our method; a linear layer converts the 28D input vector (i.e., 2 for the original corner coordinate and 20 for the room/door type one-hot vector) to the 256D feature map, six Transformer encoder blocks, and the attention in each block have four heads. We also use an MLP to convert 256D output embedding to 4D output.

Diffusion model, one room per node encodes each node as corresponding to a room instead of a corner in the room. To ease the implementation, we set the maximum number of nodes per room to 20 and we pad extra nodes when the room has less than 20 nodes with 0. We flatten the conditions per room and then use a linear layer to convert it to a 256D embedding vector. Each feature map represents a room and an input token for Transformer, we use the same Transformer as our method. After the Transformer blocks, a linear layer converts 256D output to 4D (i.e., 2 for the position and 2 for the rotation).

Shabani *et al.* [2] takes the input layout of each room with the resolution of 256×256 with the same number of channels as the number of room types to pass each pixel as a one-hot vector of the corresponding room type. We use the same model as [2] and change the number of input channels to 11 for RPLAN and 20 for MagicPlan. To generate the arrangement candidates, we use the given room layouts of our dataset to connect doors, while we also use overlap filtering to reduce the number of candidates. Note that our datasets is significantly larger than the one in [2], enabling us to randomly select a positive or a negative candidate in each iteration and therefore remove the class imbalance weight used in [2]. During the training for each house, we randomly select a GT with the label 1 or a faulty candidate with the label [0, 1) based on the number of mismatched doors. During the test, we pass all the possible candidates of each house and select the candidate with the highest score as the final prediction.

Harel *et al.* [3] proposed a two-step algorithm for CJP. Their method considers two types of constraints to find plausible mates based on the length and angle of different pairs of connections. By estimating the matings hierarchically using these constraints, they approach the problem of finding positions as a multi-body spring-mass system. We utilize the authors' provided implementation[2] for comparison with our method. With a test dataset of 1000 crossing-cut puzzles, we restrict the running time of the spring-system algorithm to 2 minutes per puzzle. Furthermore, unlike the provided implementation, we also consider failure cases in the metrics. Regarding the pictorial case, the authors score a candidate mating by extrapolating the images of puzzle pieces and considering the difference of the mean color value on the edges. We do not impose any time limit as we evaluate only on 20 samples.

Room Type	3	4	5	6	7	8	9	10	All
Master bedroom	0.20	0.26	0.29	0.32	0.39	0.49	0.49	0.53	0.34
Living room	0.52	0.56	0.59	0.65	0.71	0.75	0.79	0.79	0.65
Kitchen	0.42	0.47	0.52	0.59	0.68	0.71	0.76	0.79	0.59
Bathroom	0.54	0.70	0.85	0.96	1.12	1.28	1.33	1.47	0.96
Toilet	0.07	0.11	0.15	0.22	0.22	0.25	0.27	0.26	0.18
Corridor	0.07	0.12	0.15	0.19	0.25	0.32	0.37	0.45	0.21
Closet	0.13	0.18	0.22	0.32	0.48	0.68	0.92	1.22	0.41
Hall	0.35	0.55	0.68	0.77	0.85	0.91	0.98	1.08	0.73
Laundry room	0.05	0.05	0.05	0.07	0.10	0.13	0.18	0.23	0.09
Bedroom	0.34	0.69	1.08	1.32	1.51	1.74	1.93	2.10	1.23
Balcony	0.05	0.08	0.16	0.32	0.40	0.48	0.56	0.64	0.29
Dining room	0.13	0.12	0.12	0.14	0.17	0.19	0.22	0.25	0.15
Private office	0.00	0.01	0.02	0.05	0.05	0.07	0.08	0.10	0.05
Den	0.05	0.05	0.06	0.7	0.09	0.11	0.12	0.12	0.08
Storage	0.00	0.01	0.01	0.02	0.02	0.02	0.03	0.03	0.02
Others	0.00	0.01	0.01	0.02	0.03	0.04	0.04	0.07	0.03
Doors	2.84	3.82	4.82	5.43	6.92	8.02	6.10	10.18	5.90

Table 1: MagicPlan dataset consists of floorplans with 3 to 10 rooms. The table shows average number of rooms with a specific room type based on the total number of rooms in the house.

B Datasets Details and Preprocessing

We normalize the puzzles/floorplans for each task and dataset by scaling them to fit within a 1×1 square, and we also resize all corresponding images to dimensions of 256×256 in the case of pictorial CJP. While this normalization process does not introduce any essential additional information during testing in CJP and VJP, it could potentially enables the network to cheat in RLA, as the longer extent of arranged floorplans is always fixed to 1. To address this issue, during testing in RLA, we apply a random scaling factor in the range of [0.8, 1.0] to the room shapes of each house. In the subsequent sections, we provide a detailed description of each dataset. In the following, we provide additional statistics for our floorplan datasets.

The Voronoi Jigsaw Puzzle dataset consists of 200k training puzzles and 1k testing puzzles. These puzzles were created by randomly selecting 3 to 15 points, The individual pieces of the puzzles were obtained by extracting the Voronoi cells corresponding to these points. There are 1,066, 14,033, 23,715, 20,279, 16,073, 15,235, 16,428, 15,018, 16,318, 15,096, 16,487, 16,233, and 15,670 puzzles with 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, and 15 pieces respectively. Each piece has a minimum, maximum, and average of 3, 20, and 4.51 corners respectively. The minimum, maximum, and average number of corners per puzzle are 10, 93, and 42.24.

Cross-cut Jigsaw Puzzle (CJP) are consist of 100k training and 1k testing puzzles, where each one were generated using [3] method which generate a convex polygon and cuts it by 3 to 5 lines. There are 1719, 6046, 15854, 14521, 6905, 10929, 12065, 8361, 6521, 8192, 7663, 4642, 1508, 73, and 1 puzzles with 3, to 18 pieces respectively. Each piece has a minimum, maximum, and average of 3, 13, and 4.47 corners respectively. The minimum, maximum, and average number of corners per puzzle are 16, 76, and 41.99.

MagicPlan dataset consists of roughly 98K houses/apartments, which we divide into 93K training and 5K testing samples. 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. Table 1 shows average number of rooms with a specific room type based on the total number of rooms in the house.

[1]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.

[2]https://icvl.cs.bgu.ac.il/polygonal-puzzle-solving/

In the RPLAN dataset, we divide 60K samples in RPLAN to 55K train and 5K test. The number of rooms in a house ranges from 3 to 8. Concretely 99, 582, 5083, 19551, 21921, and 13235 houses contain 3, 4, 5, 6, 7, and 8 rooms, respectively.

C Additional ablation studies

C.1 Additional ablation studies on room layout arrangement

Table 2: Main quantitative results with two metrics: Positional Error (MPE) and Graph Editing Distance (GED). This table show a case where the ground-truth rotations are given, as TransRaster baseline cannot handle rotations. Small RPLAN (resp. Small MagicPlan) 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 MagicPlan		Full JigsawPlan	
Metric	MPE (↓)	GED (↓)	MPE (↓)	GED (↓)	MPE (↓)	GED (↓)	MPE (↓)	GED (↓)
Shabani *et al*.	17.6	1.0	✗	✗	32.2	1.1	✗	✗
TransRaster	13.9	1.2	15.7	2.1	36.1	2.1	41.9	4.1
TransVector	12.9	1.1	13.9	2.0	37.7	1.9	42.8	4.0
Ours	**4.6**±0.7	**0.4**±0.0	**5.4**±0.7	**0.6**±0.0	**17.5**±0.8	**1.0**±0.4	**27.9**±0.7	**2.7**±0.5

Figure 6 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. Figure 7 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.

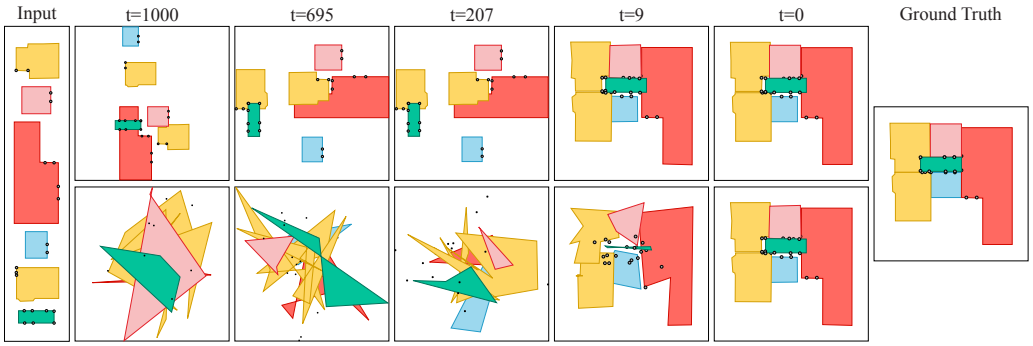

Figure 2: Visualization of predicted layouts at step "t"s. At t=1000, position parameters at each corner are initialized by a Gaussian noise, and at t=0, there is the final predicted layout. The top row shows the predicted layout without averaging/voting, and the bottom row shows with averaging/voting. To make it more clear, we show doors by their corners.

C.1.1 Additional ablation on RPLAN

The main paper shows the ablation studies on the MagicPlan dataset in case of room layout arrangment task. This part of supplementary will present the same study results on RPLAN dataset. Table 3 shows the impact of our attention module and door matching loss on performance and Table 4 shows the impact of noise in the room type and door detection on our performance, although there is a performance drop, our method still works better than the competing methods.

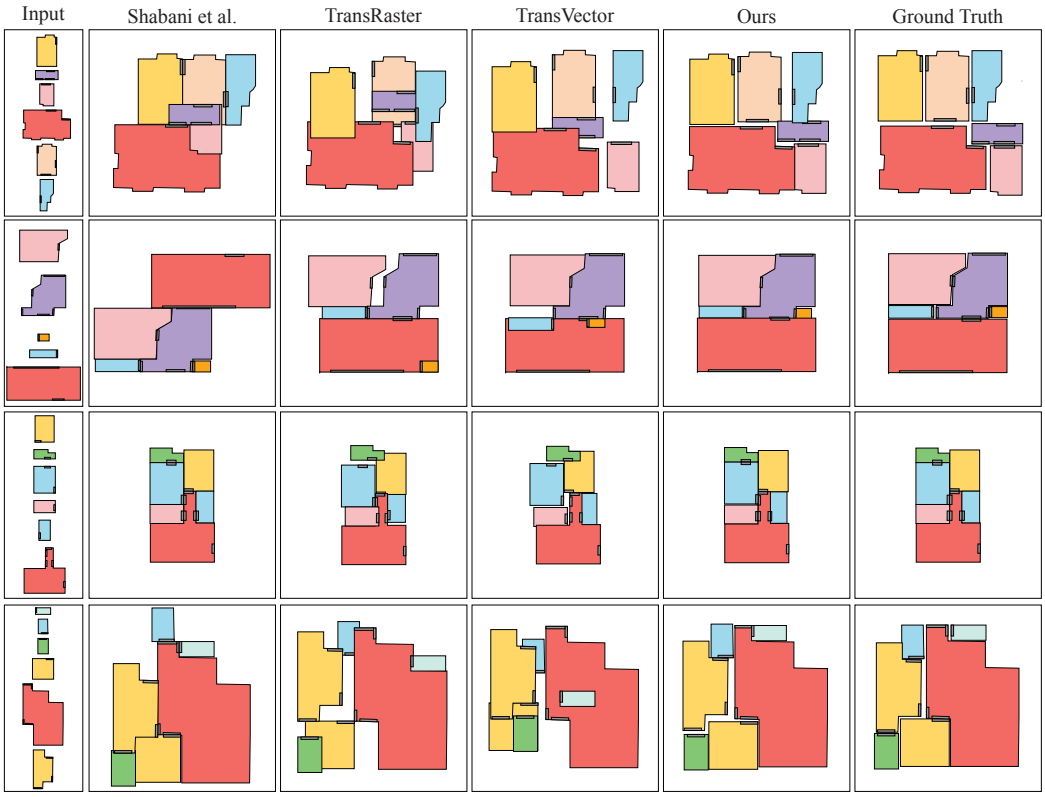

Figure 3: Qualitative evaluations of our approach against the three competing methods. The top two rows are from Small MagicPlan. The bottom two rows is from Small RPLAN. The GT rotations are given for all the cases to enable comparisons with all the methods.

Table 3: Co attention mechanisms (P-SA, G-SA) and the door matching loss (L_{match}). Full RPLAN is used. ✓indicates the feature being used. In case of L_{match} "Doors" means matching loss has been applied only on door corners and "All corners" means matching loss has been applied to all corners including door corners.

P-SA	G-SA	L_{match}	MPE (↓)	GED (↓)
	✓		25.6	1.6
✓	✓		24.2	1.5
✓		Doors	36.9	2.4
	✓	Doors	22.1	1.1
✓	✓	All Corners	10.7	0.9
✓	✓	Doors	10.5	0.9

Table 4: Effects of the room-type (R-type) and the Door information. Full RPLAN 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.

Train		Test		MPE(↓)	GED (↓)
R-Type	Door	R-Type	Door		
✓	✓		✓	17.3	1.4
	✓		✓	16.4	1.5
✓	✓	✓		15.1	1.9
✓		✓		14.3	1.9
✓	✓	✓	✓	10.5	0.9

C.2 Additional ablation studies on puzzle solving

We have provided additional qualitative results of our method in Figure 8 and Figure 9 including noisy samples or samples with missing or duplicate pieces. In case of missing and duplicate experiment, we repeat (remove) each piece with a probability of 10%. Table 5 presents the evaluation metrics of missing and duplicate experiments.

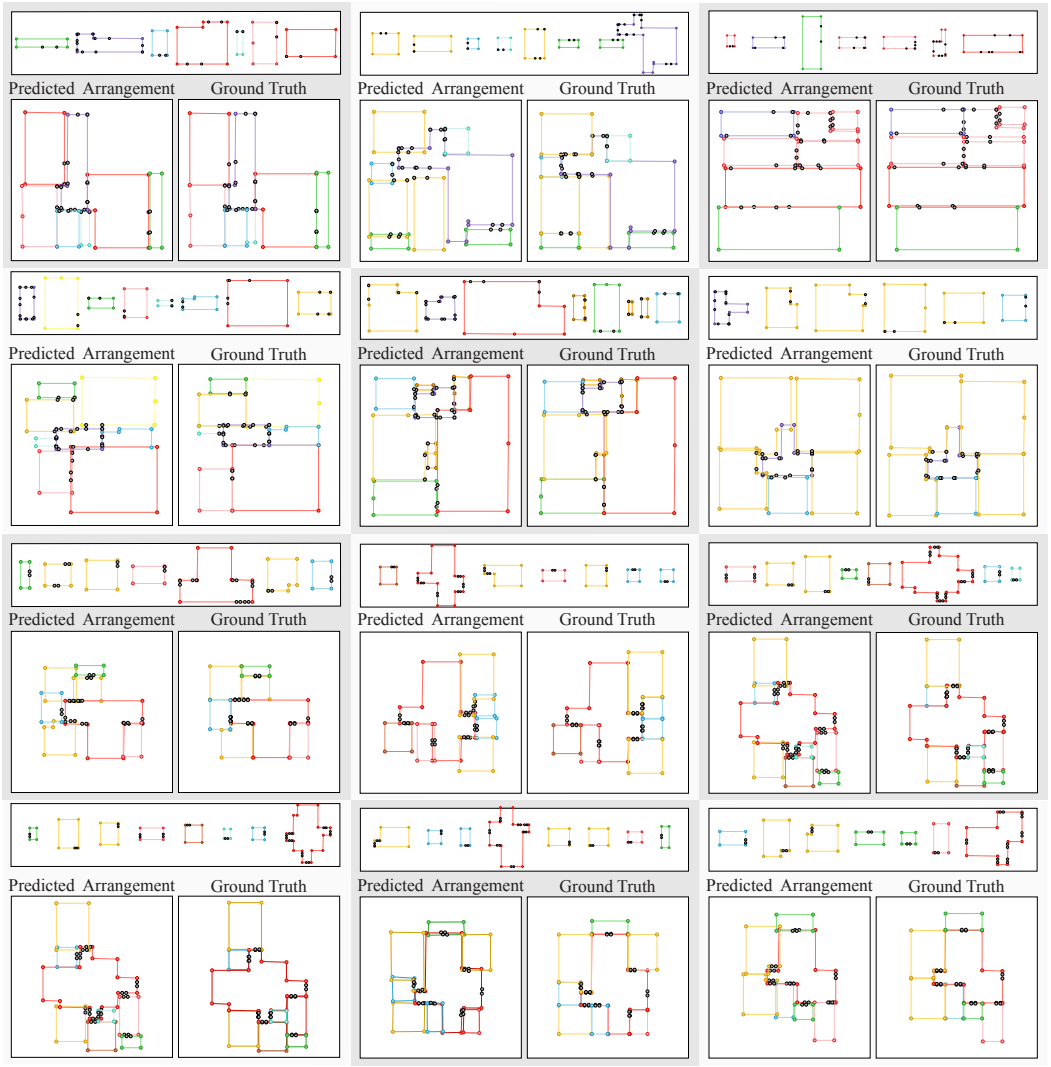

Figure 4: Qualitative evaluations of our method for Full MagicPlan dataset without GT rotations top two rows, and Full RPLAN dataset without GT rotations bottom two rows. We show edges and corners here to show overlaps and noisy annotations more clear.

C.2.1 Pictorial Cross-cut Jigsaw Puzzle

To enhance the integration of image information into our pictorial puzzle diffusion models, we employed a two-step approach. Firstly, we pretrained an auto-encoder utilizing the puzzle pieces. This auto-encoder served as the image embedder for our diffusion model, enabling the conversion of each puzzle into a compact 128D feature vector.

The pretraining process involved training the model to downsample an input image of dimensions $3 \times 256 \times 256$ to a compressed representation of size $32 \times 2 \times 2$ within the encoder, and subsequently reconstructing the original image size in the decoder. We employed the mean squared error (MSE) loss function during training. However, to focus our model's attention on learning the texture features, given that the diffusion model already captured the geometry features, we applied the loss function exclusively to the pixels within the puzzle piece.

By adopting this selective application of the loss function, we prioritize the acquisition of texture-based details, as the geometric characteristics are already embedded within the diffusion model.

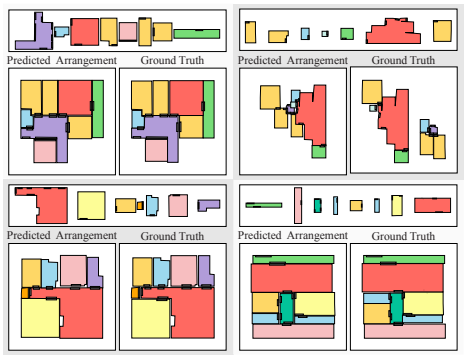

Figure 5: RLA arrangement results with Full MagicPlan dataset. On left two successful cases, on right two failed cases,. Our failures are often attributed to 1) Rare building architecture (top-right) and 2) Inherent ambiguity (bottom-right), whose tasks are challenging even for humans.

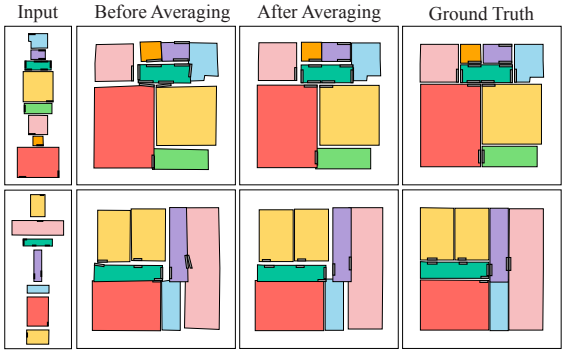

Figure 6: The final room arrangement before and after averaging. Our diffusion model estimates a room position/rotation at each room corner, which may not be consistent in a room. The final arrangement is obtained by taking the average position and rotation within each room.

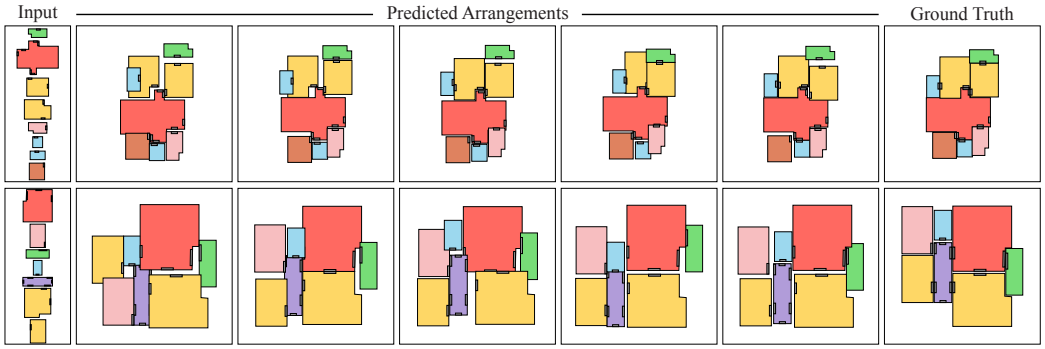

Figure 7: A diffusion model is stochastic and produces a different result every time. The middle rows show five different pose estimation results. The top (resp. bottom) is from Full RPLAN (resp. Full MagicPlan) dataset.

Quantitatively, we also evaluated our method on the full Cross-cut dataset to measure the effectiveness of the pictorial information compared to apictorial scenario. We found that the model converges faster when using pictorial information while it achieves slightly better overlap score of 0.9417 compared to 0.9398 in apictorial scenario. Figure 10 shows additional qualitative results of our method compared to Harel *et al*. [3].

Table 5: Effects of the Missing or Duplicate pieces in puzzle solving problem. ✓indicates it if missing or duplicate piece were presented during test time. In training time we do not have duplicate or missing piece presented to show our model robustness to unseen noise during test.

		Cross-cut			Voronoi		
Missing	Duplicate	Overlap (↑)	Precision (↑)	Recall (↑)	Overlap (↑)	Precision (↑)	Recall (↑)
✓	-	0.88	0.92	0.82	0.68	0.68	0.56
-	✓	0.92	0.97	0.88	0.67	0.71	0.57
-	-	0.94	0.97	0.91	0.70	0.78	0.60

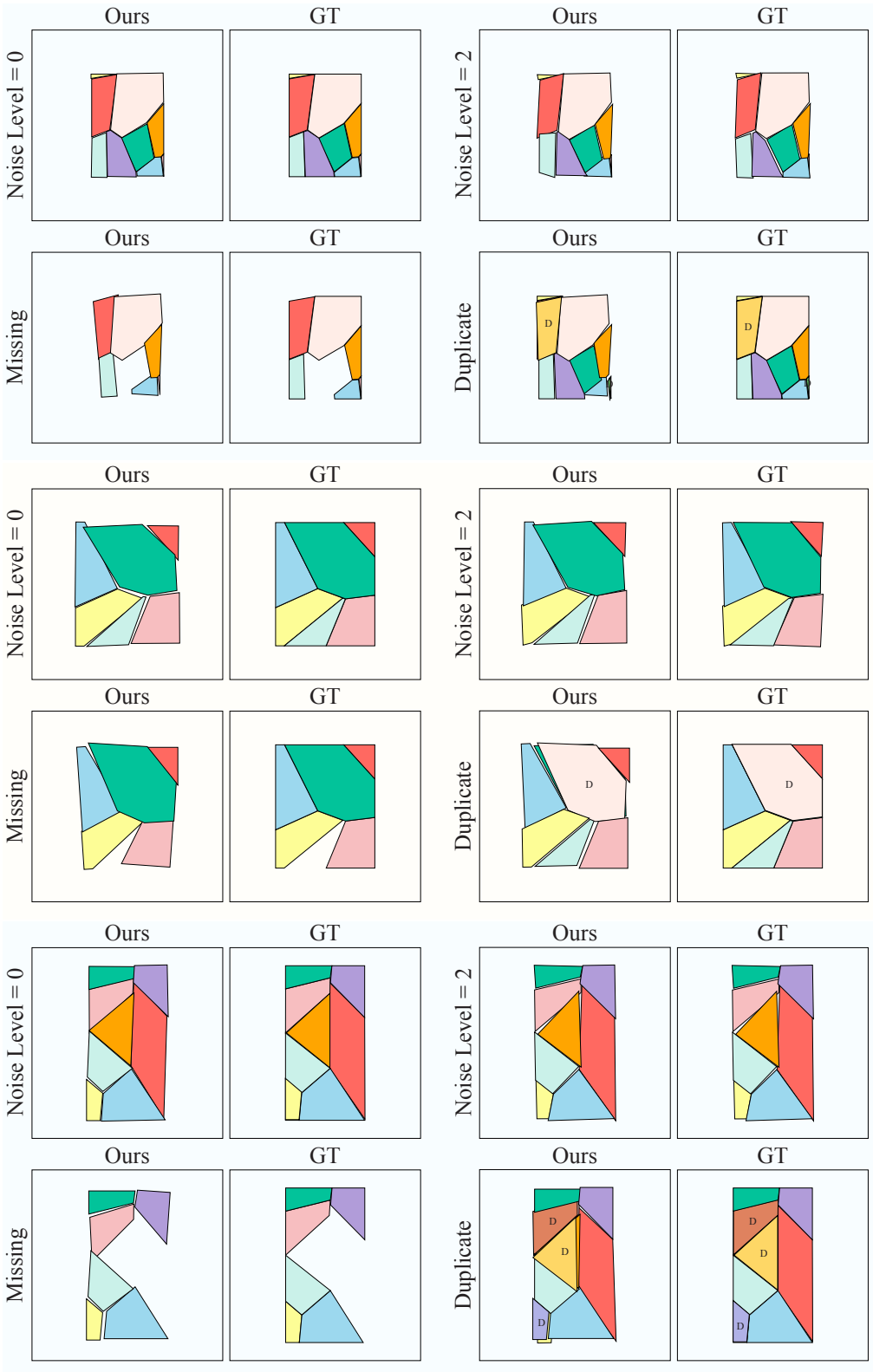

Figure 8: Additional qualitative results of Voronoi jigsaw puzzle are presented in four different setups: 1) No noise, 2) Noise level 2, 3) Missing piece, and 4) Duplicate piece (D indicates the duplicated pieces).

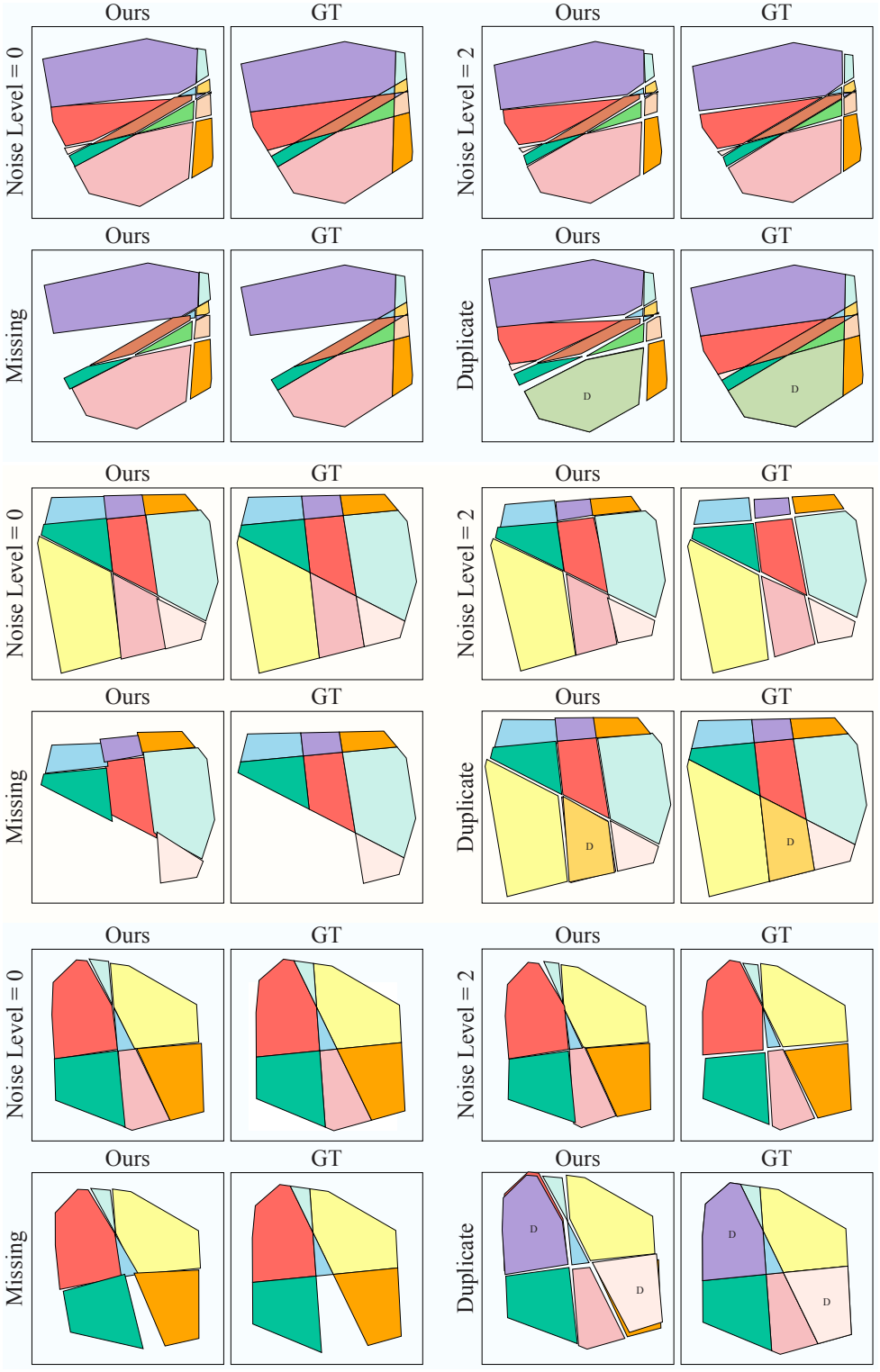

Figure 9: Additional qualitative results of Cross-cut jigsaw puzzle are presented in four different setups: 1) No noise, 2) Noise level 2, 3) Missing piece, and 4) Duplicate piece (D indicates the duplicated pieces).

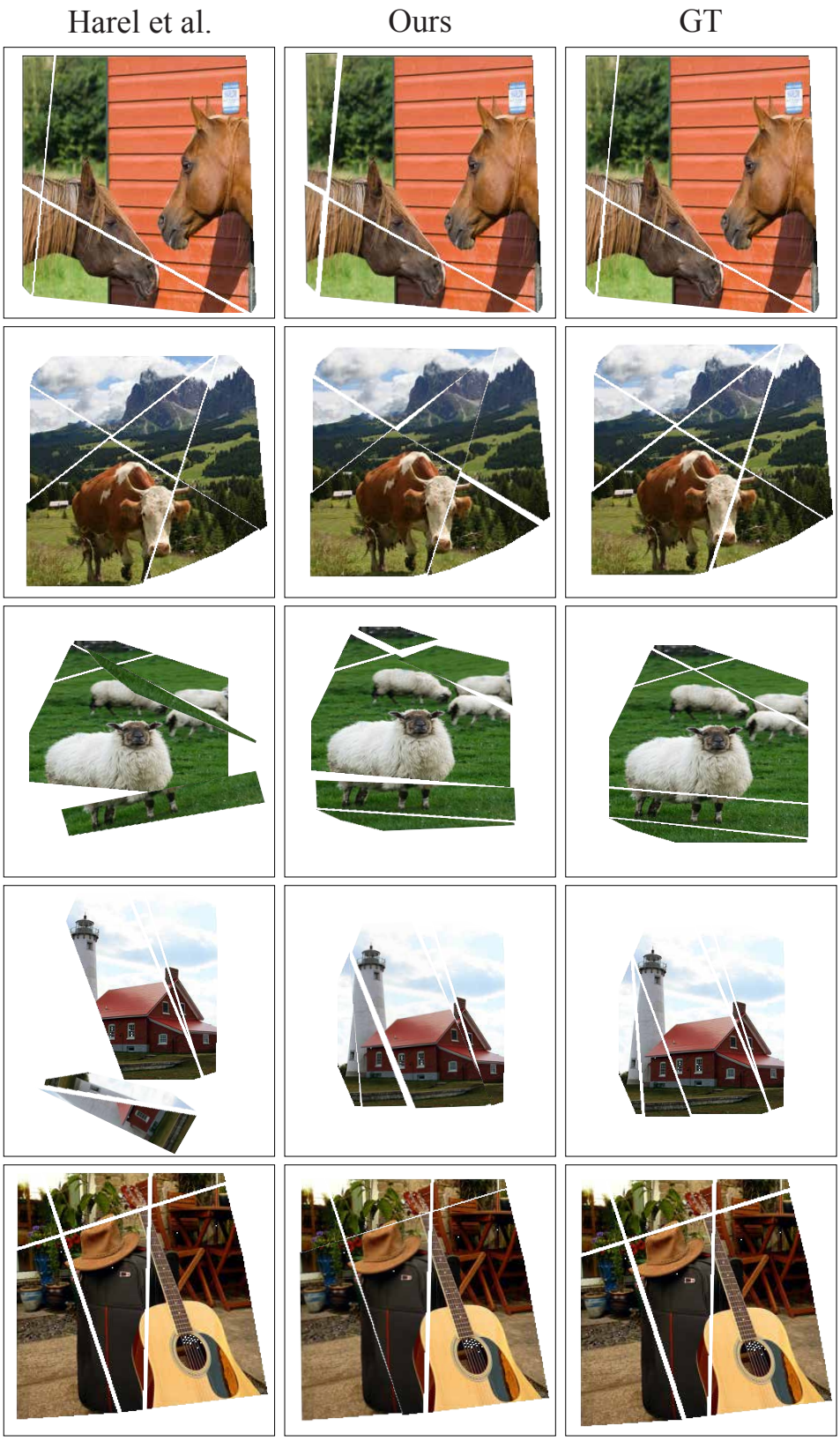

Figure 10: Additional qualitative results of pictorial Cross-cut jigsaw puzzle compared to Harel *et al*. [3].

References

[1] 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.

[2] 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.

[3] Peleg Harel and Ohad Ben-Shahar. Crossing cuts polygonal puzzles: Models and solvers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3084–3093, 2021.