

A. Diffusion Models

A.1. DDPM forward process re-parameterization, simplified loss

DDPM [19] suggests re-parameterizing Eq. (4) as

$$\mathbf{x}_t(\mathbf{x}_0, \epsilon) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon \quad (17)$$

with $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Then, let ϵ_θ be a function approximator to predict ϵ from \mathbf{x}_t such that for the reverse process $\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right)$ and L_{t-1} for $2 \leq t \leq T$ in (2) becomes

$$L_{t-1} = \mathbb{E}_{\mathbf{x}_0, \epsilon} \left[\frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1 - \bar{\alpha}_t)} \left\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t) \right\|^2 \right] + \text{const} \quad (18)$$

where the constant term can be dropped in training.

On image data, [19] found it beneficial to sample quality to train on this simplified version of the variational bound ($L_{0:T-1}$):

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left[\left\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t) \right\|^2 \right] \quad (19)$$

with uniform t between 1 and T , dropping the scaling factors of the squared norms. We use Eq. (19) to compute L_{vb}^{DDPM} in MiDiffusion.

A.2. D3PM reverse parameterization, auxiliary loss

Similar to DDPM, [2, 21] suggested that approximating some surrogate variables gives better quality. Specifically, they trained a neural network $\tilde{p}_\theta(\tilde{z}_0|z_t)$, multiplied it with the posterior $q(z_{t-1}|z_t, z_0)$, and marginalized \tilde{z}_0 to obtain $p_\theta(z_{t-1}|z_t)$:

$$p_\theta(z_{t-1}|z_t) \propto \sum_{\tilde{z}_0} q(z_{t-1}|z_t, z_0) \tilde{p}_\theta(\tilde{z}_0|z_t). \quad (20)$$

D3PM [2] introduced an auxiliary denoising objective to encourage good predictions of the data z_0 at each time step. Their complete training loss is:

$$L_\lambda(\theta) = L_{vb}(\theta) + \lambda \mathbb{E}_{q(z_0)q(z_t|z_0)} [-\log \tilde{p}_\theta(z_0|z_t)] \quad (21)$$

which correspond to L_{vb}^{D3PM} and λL_{aux}^{D3PM} in our MiDiffusion algorithm. This can be implemented efficiently in training, since the parameterization of the backward step directly computes $\tilde{p}_\theta(z_0|z_t)$.

B. MiDiffusion Loss Factorization

We provide more details for the factorization step in Eq. (14). We first consider a general case comparing the

KL-divergence between $\tilde{q}(z)\hat{q}(\mathbf{x})$ and $\tilde{p}(z)\hat{p}(\mathbf{x})$ for discrete z and continuous \mathbf{x} :

$$\begin{aligned} D_{\text{KL}}(\tilde{q}(z)\hat{q}(\mathbf{x})||\tilde{p}(z)\hat{p}(\mathbf{x})) &= \int_{\mathbf{x}} \sum_z \tilde{q}(z)\hat{q}(\mathbf{x}) \log \frac{\tilde{q}(z)\hat{q}(\mathbf{x})}{\tilde{p}(z)\hat{p}(\mathbf{x})} \\ &= \int_{\mathbf{x}} \sum_z \left[\tilde{q}(z)\hat{q}(\mathbf{x}) \log \frac{\tilde{q}(z)}{\tilde{p}(z)} + \tilde{q}(z)\hat{q}(\mathbf{x}) \log \frac{\hat{q}(\mathbf{x})}{\hat{p}(\mathbf{x})} \right] \\ &= \sum_z \tilde{q}(z) \log \frac{\tilde{q}(z)}{\tilde{p}(z)} \underbrace{\int_{\mathbf{x}} \hat{q}(\mathbf{x})}_{1} + \int_{\mathbf{x}} \hat{q}(\mathbf{x}) \log \frac{\hat{q}(\mathbf{x})}{\hat{p}(\mathbf{x})} \underbrace{\sum_z \tilde{q}(z)}_1 \\ &= D_{\text{KL}}(\tilde{q}(z)||\tilde{p}(z)) + D_{\text{KL}}(\hat{q}(\mathbf{x})||\hat{p}(\mathbf{x})). \end{aligned} \quad (22)$$

This means we can decouple the KL-divergence computation between mixed-domain probability distributions to a sum of domain-specific KL-divergence computations. Note that this holds for any choices of \tilde{q} , \hat{q} , \tilde{p} , \hat{p} . For our specific case:

$$\begin{aligned} L_{t-1}^{\text{mixed}} &= \mathbb{E}_{q(z_t, \mathbf{x}_t|z_0, \mathbf{x}_0)} [D_{\text{KL}}(q(z_{t-1}, \mathbf{x}_{t-1}|z_t, \mathbf{x}_t, z_0, \mathbf{x}_0) \\ &\quad ||p_\theta(z_{t-1}, \mathbf{x}_{t-1}|z_t, \mathbf{x}_t, \mathbf{y})) \\ &= \mathbb{E}_{\tilde{q}(z_t|z_0)\hat{q}(\mathbf{x}_t|\mathbf{x}_0)} [D_{\text{KL}}(\tilde{q}(z_{t-1}|z_t, z_0)\hat{q}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \\ &\quad ||\tilde{p}_\theta(z_{t-1}|z_t, \mathbf{x}_t, \mathbf{y})\hat{p}_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, z_t, \mathbf{y})) \\ &= \mathbb{E}_{\tilde{q}(z_t|z_0)} [D_{\text{KL}}(\tilde{q}(z_{t-1}|z_t, z_0)||\tilde{p}_\theta(z_{t-1}|z_t, \mathbf{x}_t, \mathbf{y})) \\ &\quad + \mathbb{E}_{\hat{q}(\mathbf{x}_t|\mathbf{x}_0)} [D_{\text{KL}}(\hat{q}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||\hat{p}_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, z_t, \mathbf{y}))], \end{aligned} \quad (23)$$

where the last equality holds because of Eq. (22).

C. Network and Implementation Details

We include detailed network hyper-parameters for our denoising network, and implementation for training.

C.1. Network Hyper-parameters

The architecture of MiDiffusion has at its core a series of 8 transformer blocks using a hidden dimension of 512 with 4 heads and feed-forward layers with an internal dimension of 2048. Following [15], we use GELU [18] as nonlinearity. The semantic attributes are embedded in a learned vector of length 512, and the geometric attributes are mapped by a 3-layer MLP, with internal dimensions of [512, 1024], to a 512-dimensional space before being fed to the transformer decoder blocks. We extract floor plan features of dimension 64 using a 4-layer PointNet [39] with internal dimensions [64, 64, 512] as in LEGO-Net [55]. For baselines and ablation studies, we use the default ResNet-18 [16] image feature extractor proposed by ATISS [36] and also implemented by DiffuScene [50] to compute the 64-dimensional floor plan features from binary floor plan masks. The outputs of the transformer decoder are fed to two separate MLPs to decode the semantic and geometric predictions. The semantic feature decoder is a 1-layer MLP

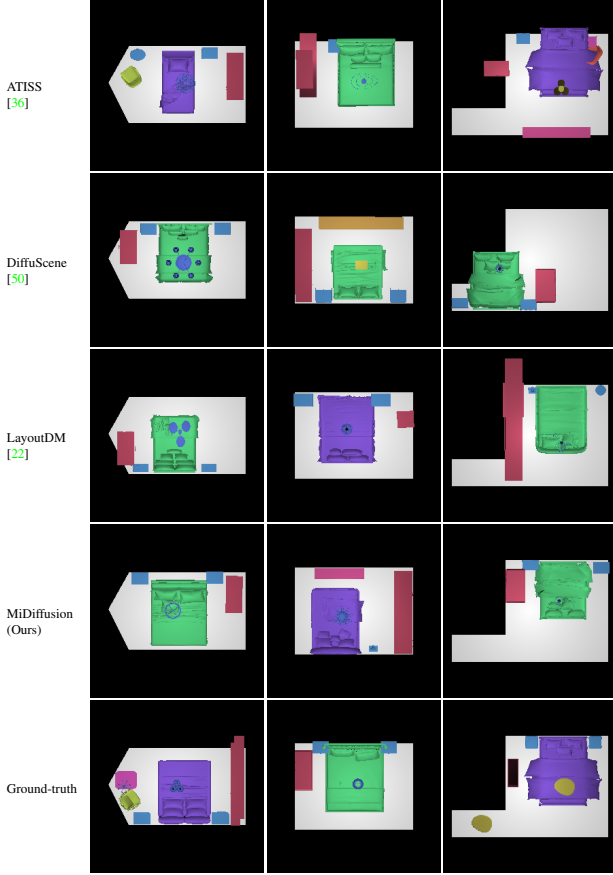


Figure 5. Example bedroom top-down orthographic projection images for quantitative evaluations.

that produces a categorical distribution over z , and the geometric feature decoder is a 3-layer MLP with hidden dimensions [512, 1024], producing the 8-dimensional Gaussian mean for the geometric attributes \mathbf{x} .

C.2. Implementation

We set a fixed learning rate of $l_r = 2e^{-4}$ using the Adam optimizer, and a dropout ratio of 0.1 for multi-head attention and feed-forward layers in the transformer blocks. We train 50k epochs on the bedroom dataset with 0.5 learning rate decay every 10k epochs, and 100k epochs on the living and dining room datasets with 0.5 decay every 15k epochs. We use a linear schedule over 1000 diffusion steps for all noise parameters in the forward process. In the discrete domain α_t and γ_t range from $1 - 1e^{-5}$ to 0.99999 and from $9e^{-6}$ to 0.99999 respectively. In the continuous domain, β_t starts from $1e^{-4}$ at reaches 0.02. We train all our models on a single NVIDIA V100 GPU with under 8GB of GPU RAM usage. The training time range from around 20 hours on living and dining room datasets to about 32 hours on the bedroom dataset for a batch size of 512 scenes.

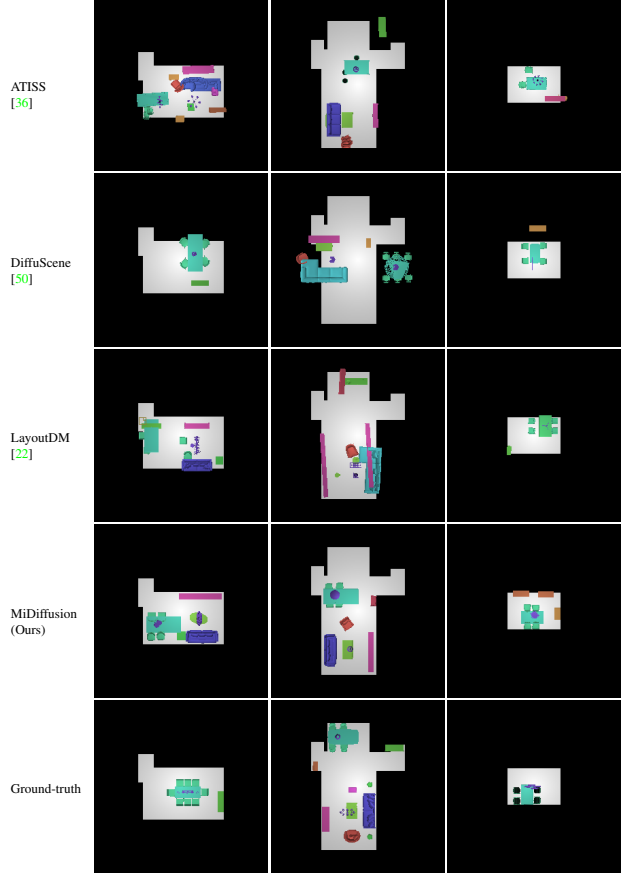


Figure 6. Example dining room top-down orthographic projection images for quantitative evaluations.

D. Example Synthesized Layout Images

We use texture-less objects rendered on white floor plans for quantitative evaluations, since floor and object textures affect the clarity of the results. The object labels are sorted in alphabetical order for each room type, and then the associated colors are evenly sampled along a circular path using Seaborn’s [54] “hls” color palette. We render top-down layout images using simple-3dviz [1] in accordance with prior works [36, 50]. We include examples of rendered images for three comparing approaches in Fig. 5, 6, 7 from the floor-conditioned scene synthesis experiment in Sec 2. Some ground-truth layouts include objects slightly out of boundary. This is a problem in the raw 3D-FRONT dataset. Therefore, we inflate the room boundary by 0.1m when counting the number of out of boundary objects. The bedroom datasets is the easiest (less number of objects, smaller in room size, more training data) so that all comparing approaches can generate good predictions. On the harder living room and dining room datasets, MiDiffusion clearly outperforms the baselines by generating realistic geometric arrangements with desired symmetry and alignment

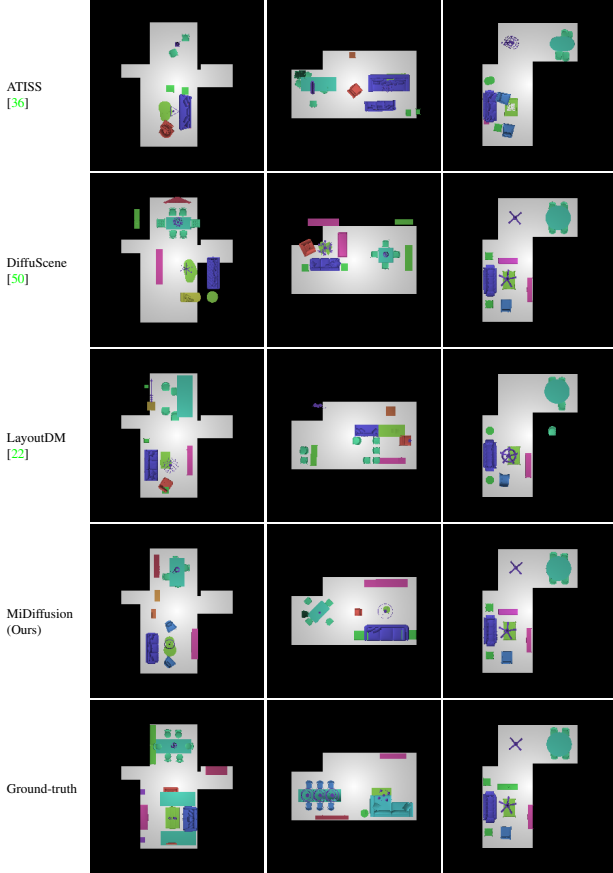


Figure 7. Example living room top-down orthographic projection images for quantitative evaluations.

between objects, while respecting the floor boundary constraints. DiffuScene is also capable of generating good geometric arrangements but their architecture is less optimal for learning floor boundary constraints.

E. Additional Experimental Evaluations

We provide supplementary results to complement Sec. 5. We begin with additional results from the ablation studies to further support our proposed mixed diffusion formulation and network design. Then, we present more results on scene synthesis with partial constraints, including a comparison with the direct masking strategy and applications to other conditional setups. We show quantitative comparison between MiDiffusion and DiffuScene in the unconditioned scene synthesis experiment, which is original target application of DiffuScene. Lastly, we show generalization of MiDiffusion by conditioning our models on out-of-distribution floor plans, which are not seen in the training data.

E.1. Ablation Study

We show an additional ablation study comparing MiDiffusion (also referred as Mixed+PointNet in ablation stud-

ies) and the 2-stage approach to supplement Table 3. The second row of Table 7 reports results of a 2-stage variant where ground-truth semantic labels are sent to the geometric diffusion stage. While bypassing the first semantic diffusion stage improves performance, this variant still underperforms compared to MiDiffusion. This suggests that even with a perfect first-stage label predictor, the 2-stage approach still produces less realistic scene layouts compared to ours.

We show object evaluation results in Table 8. Across all three datasets, there is a consistent improvement over OOB % as we modify the floor plan feature extractor and diffusion framework towards our final design of MiDiffusion. This is consistent with the main results in Table 3.

E.2. Diversity

To provide more insights about MiDiffusion’s small drop in size diversity compared to DiffuScene in Table 4, we show results for other three joint prediction variants in our ablation study in Table 9. MiDiffusion is denoted as “Mixed+PointNet” to highlight the architecture differences similar to its notation in the ablation study. First, note that the DDPM+ResNet variant, differing from DiffuScene only in its transformer backbone, shows that the transformer results in a small diversity drop in size compared to U-Net (in DiffuScene). However, since this variant shows substantial improvements in main evaluation metrics (ablation study in Sec. 5.1) which evaluate realism, we believe this trade-off is justified and might be a result of stronger boundary constraints (OOB% in Table 2, 8). Second, comparison across the last four methods sharing the same transformer backbone in Table 9 shows that, in general, our mixed diffusion formulation and PointNet extractor maintain diversity in position and size while producing more realistic layouts (ablation study in Sec. 5.1).

E.3. Masking Comparison

In Sec. 5.2, we show that, with the proposed corruption-and-masking strategy, MiDiffusion models are able to generate room layouts under various types of object constraints. Compared to ours, direct masking at each denoising step is less principled as it introduces out-of-distribution latent variables in the denoising process. We provide an example comparison in Table 10 to demonstrate that our proposed masking strategy yields better results.

E.4. Additional Applications

Our models can easily extend to other applications with different masking ranges. For example, they can be applied to object category constrained scene synthesis given a set of desired object labels. Fig. 8 shows two examples of label constrained scene synthesis. In particular, the dining room scene is conditioned on five dining chairs, which occur less

Method	Bedroom				Dining room				Living room			
	FID	KIDx0.001	CA %	KLx0.01	FID	KIDx0.001	CA %	KLx0.01	FID	KIDx0.001	CA %	KLx0.01
2-stage + PointNet	65.54	1.53	57.83 \pm 4.89	3.50	38.80	7.02	61.53 \pm 7.85	5.08	35.35	6.30	61.97 \pm 6.55	4.73
2-stage (gt labels) + PointNet	64.37	0.67	56.19 \pm 5.45	–	32.47	5.07	56.29 \pm 4.80	–	32.26	5.86	58.85 \pm 6.29	–
Mixed+PointNet	63.71	0.24	53.65 \pm 2.46	1.05	30.63	2.74	52.10 \pm 1.72	1.90	28.60	1.64	54.04 \pm 3.33	1.54

Table 7. Additional ablation study for floor-conditioned 3D scene synthesis.

Method	Bedroom			Dining room			Living room		
	Obj	OOB %	IoU %	Obj	OOB %	IoU %	Obj	OOB %	IoU %
2stage + PointNet	5.04	5.67	0.72	11.30	11.80	1.57	11.50	13.35	1.48
2stage (gt labels) + PointNet	–	5.20	0.61	–	8.87	1.18	–	11.80	1.28
DDPM+ResNet	5.07	6.63	0.50	10.87	7.08	0.70	11.90	10.93	0.70
Mixed+ResNet	5.15	4.43	0.44	10.78	7.09	0.73	11.91	10.60	0.70
DDPM+PointNet	4.92	5.29	0.70	10.78	6.75	0.78	11.83	10.37	0.64
Mixed+PointNet	5.22	3.91	0.61	10.92	5.77	0.91	11.91	7.93	0.98
Ground truth	5.22	3.37	0.24	11.11	0.73	0.48	11.67	1.55	0.27

Table 8. Geometric evaluations on ablation study for floor-conditioned 3D scene synthesis.

Method	Bedroom				Dining room				Living room			
	Position	Position-IB	Size	Size-IB	Position	Position-IB	Size	Size-IB	Position	Position-IB	Size	Size-IB
DiffuScene [50]	1.073	1.059	0.718	0.691	1.596	1.427	0.454	0.423	1.752	1.575	0.482	0.439
DDPM+ResNet	1.074	1.059	0.713	0.691	1.584	<u>1.557</u>	<u>0.413</u>	<u>0.400</u>	1.634	1.626	0.433	<u>0.424</u>
Mixed+ResNet	1.072	1.067	<u>0.712</u>	<u>0.690</u>	1.565	<u>1.550</u>	0.412	0.394	1.654	1.650	<u>0.434</u>	0.416
DDPM+PointNet	1.071	<u>1.064</u>	0.704	0.686	<u>1.569</u>	<u>1.551</u>	<u>0.413</u>	0.401	<u>1.664</u>	1.668	<u>0.434</u>	0.421
Mixed+PointNet	<u>1.073</u>	1.067	0.698	0.680	1.568	1.558	0.414	0.399	1.668	<u>1.667</u>	0.439	0.428
Ground truth	1.091	1.082	0.722	0.690	1.623	1.618	0.450	0.447	1.735	1.729	0.463	0.460

Table 9. Additional average std of predicted object positions and sizes for floor-conditioned 3D scene synthesis. We highlight the best results in bold and the second-best results with an underline among the four transformer-based architectures.

Method	Bedroom				Dining room				Living room			
	FID	KIDx0.001	OBB %	IoU %	FID	KIDx0.001	OBB %	IoU %	FID	KIDx0.001	OBB %	IoU %
MiDiffusion + DM	59.43	-0.07	7.71	0.56	32.17	4.46	6.49	0.95	30.47	3.65	8.72	0.84
MiDiffusion (ours)	59.08	-0.24	7.34	0.55	31.72	3.91	6.35	0.96	29.79	3.13	8.33	0.84

Table 10. Additional evaluation results for furniture arrangement experiment.

Method	Bedroom				Dining room				Living room			
	FID	KIDx0.001	CA %	KLx0.01	FID	KIDx0.001	CA %	KLx0.01	FID	KIDx0.001	CA %	KLx0.01
DiffuScene [50]	61.12	0.46	51.09 \pm 0.47	1.24	45.04	0.70	51.53 \pm 1.05	1.10	43.30	0.21	53.50 \pm 2.84	0.67
MiDiffusion (ours)	62.02	1.12	52.36 \pm 1.54	0.95	43.37	0.68	51.21 \pm 0.82	0.70	42.43	0.77	53.46 \pm 1.93	0.58

Table 11. Evaluation results for unconditional 3D scene synthesis.

often than an even number of chairs. Our model is still able to generate reasonable layouts. We can even combine partial constraints on object attributes (*e.g.*, label) with existing objects. Fig. 9 shows examples of such label constrained scene completion problems.

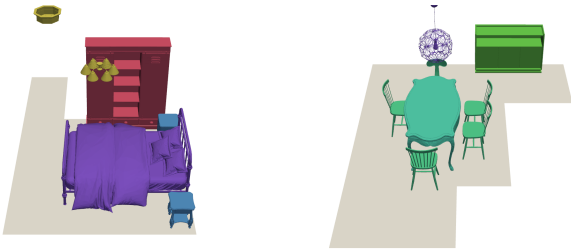


Figure 8. **Label constrained scene synthesis.** Bedroom (left) and dining room (bottom) examples.

E.5. Unconditional 3D Scene Synthesis

DiffuScene was originally proposed for 3D scene synthesis without floor conditioning. Although this is not the

main focus of this paper, we provide experimental results comparing MiDiffusion and DiffuScene for completeness using the released model weights. Since this is an unconditional problem, we slightly modify the proposed architecture in Fig. 1 by removing modules related to the conditional input, including the floor plan feature encoder, the second AdaLN and the second multi-head cross-attention unit in each transformer block. We re-use all other hyperparameters from the floor-conditioned experiments.

We follow the same layout image generation and evaluation method as in Sec. 5, except we project rendered scenes to a white background without floor plans. The results are presented in Table 11. We also note that in [50], DiffuScene used a 6.2m square to capture all synthesized layouts when generating the orthographic projection images for evaluation. Since the dining room and living room training data contains objects beyond this range, some of the predicted objects will be outside the projection range leading to incomplete result images for evaluation. In addition, the synthesized results were compared to the ground-truth training

Method	Bedroom				Dining room				Living room			
	FID	KIDx0.001	CA %	KLx0.01	FID	KIDx0.001	CA %	KLx0.01	FID	KIDx0.001	CA %	KLx0.01
DiffuScene (from [50])	17.21	0.70	52.15	0.35	32.60	0.72	55.50	0.22	36.18	0.88	57.81	0.21
DiffuScene [50]	17.43	0.82	51.09 ± 0.47	0.66	33.07	0.93	53.82 ± 4.01	0.34	35.27	0.58	54.24 ± 3.65	0.36
MiDiffusion (Ours)	18.17	0.82	52.36 ± 1.54	0.16	31.39	1.15	52.38 ± 1.85	0.18	32.49	1.20	51.86 ± 1.80	0.15

Table 12. Evaluation results for unconditional 3D scene synthesis against training scenes and using layouts projected to 6.2 squares.

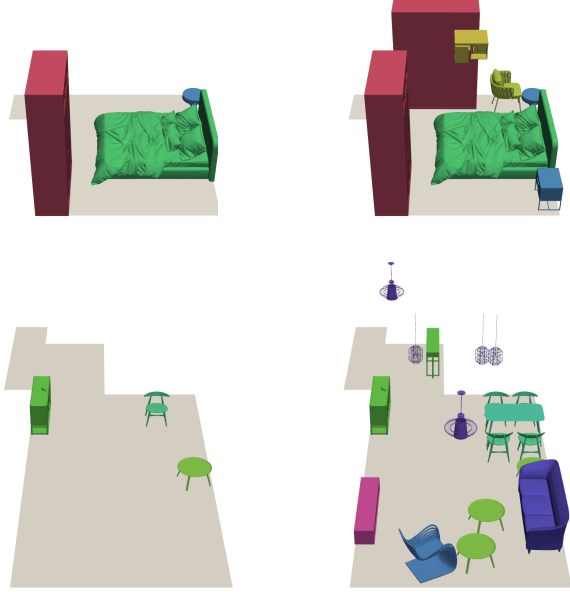


Figure 9. **Scene completion with label constraints.** Bedroom (top) and living room (bottom) label constrained scene completion examples.

split. For completeness, we follow the evaluation setup in DiffuScene and present quantitative results in Table 12 using the same predicted layouts as in Table 11. The first row is directly copied from [50] demonstrating that we reproduced DiffuScene results close to the original publication values. The only exception is CA%, for which DiffuScene does not release their classification network. Therefore, we adopt the AlexNet [27] based classifier from ATISS [36] to be consistent with Sec. 5.

Both Table 11 and 12 show that MiDiffusion achieves similar results, despite that its transformer-based denoising network is chosen for floor-conditioned scene synthesis.

F. Out-of-distribution Floor Plans

To show the generalization of MiDiffusion to out-of-distribution floor plans, we test our models conditioned on extreme floor plans that were not seen during training. Fig. 10 includes qualitative examples given trapezoidal, hexagonal, and circular floor plans. Our models can generally synthesize plausible 3D indoor scenes, aligning objects with walls or placing them in the center of the room. However, failures can arise as clusters of overlapping objects in the center of a circular room (*i.e.*, last example), since the

training data does not cover rooms with non-straight walls.

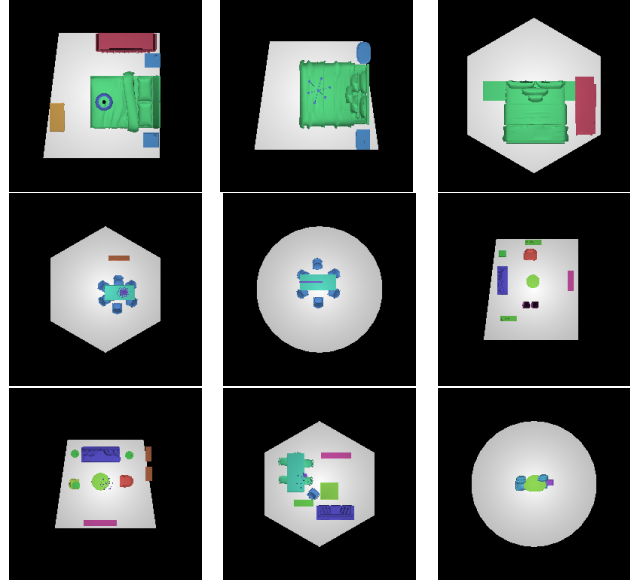


Figure 10. MiDiffusion results given extreme floor plans for bedroom (top), dining (middle), and living room (bottom) scenes.

G. 3D Rendering with SceneTex

Direct CAD model retrieval from 3D-FUTURE [13] results in objects that might be inconsistent in style. To improve visual realism and aesthetics for downstream tasks, we use SceneTex [5] to synthesize textures of 3D meshes given a text prompt describing the design style. Fig. 11 shows SceneTex enhancement over results in Fig. 2.



Figure 11. MiDiffusion results rendered via CAD retrieval [13] and with SceneTex [5] enhancement.

H. Additional Layout Examples

We provide additional layout images generated by MiDiffusion in our floor-conditioned scene synthesis experiment in Fig. 12. These layouts are generated in a sequence of random sampling of test floor plans after removing floor plan duplicates. As noted in Sec. 5 Evaluation Metrics, the raw floor plan data underestimates the actual room boundary by approximately 0.1m. Thus, small boundary violations in the layout images are expected and are found in ground-truth data as well (examples in Appendix D). MiDiffusion is able to generate realistic layouts in most cases, but object overlap still exists and might be due to coarse geometric representation. Also, geometric alignment, such as chair placement around a table, is not always perfect.

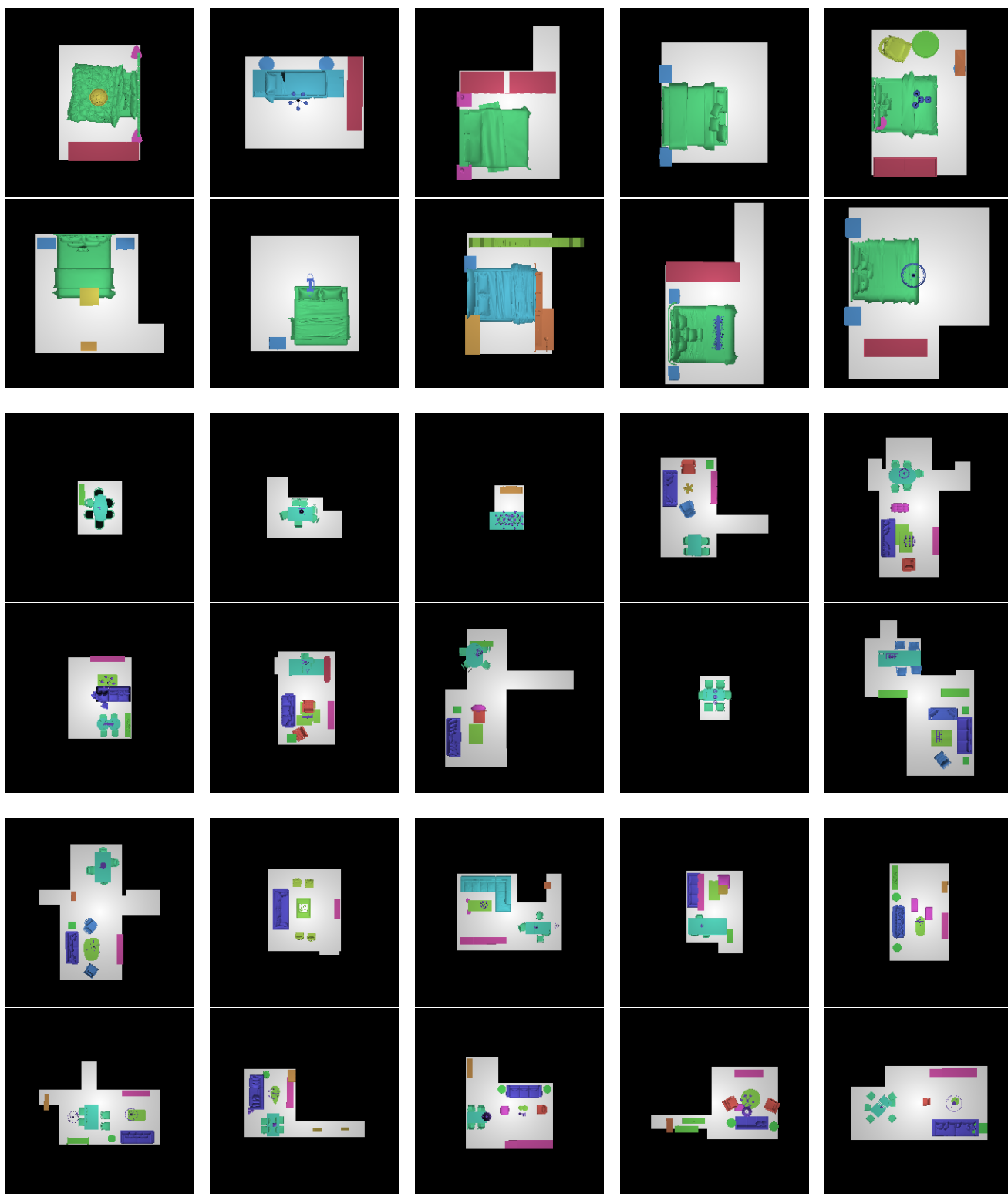


Figure 12. Example top-down orthographic projection images generated using MiDiffusion for bedroom (top), dining room (middle), and living room (bottom) scene synthesis.