DiffuScene: Denoising Diffusion Models for Generative Indoor Scene Synthesis – Supplementary Material –

Jiapeng Tang[1] Yinyu Nie[1] Lev Markhasin[2] Angela Dai[1] Justus Thies[3] Matthias Nießner[1]

[1] Technical University of Munich [2] Sony Europe RDC Stuttgart
[3] Technical University of Darmstadt

In this supplemental material, we provide details for our implementation in Sec. 1, dataset pre-processing and text prompt generation in Sec. 2, baseline implementations in Sec. 3, additional comparisons and results in Sec. 5, and user studies in Sec. 6.

1. Implementations

1.1. Shape Auto-Encoder

We adopt a pre-trained shape auto-encoder to extract a set of latent shape codes for CAD models from the 3D-FUTURE [1] dataset. The network architecture of the shape auto-encoder is shown in Fig. 1. It is a variational auto-encoder, similar to FoldingNet [11]. Specifically, a point cloud $\mathbf{P}_{in}$ of size 2,048 is fed into a graph encoder based on PointNet [4] with graph convolutions [7] to extract a global latent code of dimension 512, which is used to predict the mean μ and variance σ of a low-dimensional latent space of size 32. Subsequently, a compressed latent is sampled from $\mathcal{N}(\mu, \sigma)$. Finally, the compressed latent is mapped back to the original space and passed to the FoldingNet decoder to recover a point cloud $\mathbf{P}_{rec}$ of size 2,025. The used training objective is a weighted combination of Chamfer distance (*i.e.* CD) and KL divergence.

$$L_{vae} = \mathrm{CD}(\mathbf{P}_{in}, \mathbf{P}_{rec}) + \omega_{kl} * \mathrm{KL}(\mathcal{N}(\mu, \sigma) || \mathcal{N}(\mathbf{0}, \mathbf{I})), \quad (1)$$

where ω_{kl} is set to 0.001. The latent compression and KL regularization lead to a compact and structured latent space, focusing on global shape structures. The shape autoencoder is trained on a single RTX 2080 with a batch size of 16 for 1,000 epochs. The learning rate is initialized to $lr = 1\mathrm{e}{-4}$ and then gradually decreases with the decay rate of 0.1 in every 400 epochs.

1.2. Shape Code Diffusion

We use the extracted latent codes to train shape code diffusion. While we apply KL regularization, the value range of latent codes is still unbound. To make it easier to diffuse, we scale the latent codes to $[-1, 1]$ by using the statistical minimum and maximum feature values over the whole set. During inference, we rescale generated shape codes.

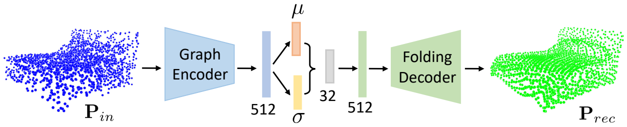


Figure 1. **Shape Auto-encoder.**

1.3. Shape Retrieval

During inference, we use shape retrieval as the post-processing procedure to acquire object surface geometries for generated scenes. Concretely, for each instance, we perform the nearest neighbor search in the 3D-FUTURE [1] dataset to find the CAD model with the same class label and the closest geometry feature.

2. Dataset

Preprocessing The dataset preprocessing is based on the setting of ATISS [3]. We start by filtering out those scenes with problematic object arrangements such as severe object intersections or incorrect object class labels, e.g., beds are misclassified as wardrobes in some scenes. Then, we remove those scenes with unnatural sizes. The floor size of a natural room is within $6m \times 6m$ and its height is less than $4m$. Subsequently, we ignore scenes that have too few or many objects. The number of objects in valid bedrooms is between 3 and 13. As for dining and living rooms, the minimum and maximum numbers are set to 3 and 21 respectively. Thus, the number of objects is $N = 13$ in bedrooms and $N = 21$ in dining and living rooms. In addition, we delete scenes that have objects out of pre-defined categories. After pre-processing, we obtained 4,041 bedrooms, 900 dining rooms, and 813 living rooms.

For the semantic class diffusion, we have an additional class of 'empty' to define the existence of an object. Combining with the object categories that appeared in each room type, we have $L = 22$ object categories for bedrooms, and $L = 25$ object categories for dining and living rooms in total. The category labels are listed as follows.

```
# 22 3D-Front bedroom categories
['empty', 'armchair', 'bookshelf', 'cabinet',
'ceiling_lamp', 'chair', 'children_cabinet',
'coffee_table', 'desk', 'double_bed',
'dressing_chair', 'dressing_table', 'kids_bed',
'nightstand', 'pendant_lamp', 'shelf',
'single_bed', 'sofa', 'stool', 'table',
'tv_stand', 'wardrobe']

# 25 3D-Front dining or living room categories
['empty', 'armchair', 'bookshelf', 'cabinet',
'ceiling_lamp', 'chaise_longue_sofa',
'chinese_chair', 'coffee_table', 'console_table',
'corner_side_table', 'desk', 'dining_chair',
'dining_table', 'l_shaped_sofa', 'lazy_sofa',
'lounge_chair', 'loveseat_sofa',
'multi_seat_sofa', 'pendant_lamp',
'round_end_table', 'shelf', 'stool',
'tv_stand', 'wardrobe', 'wine_cabinet']
```

Text Prompt Generation We follow the SceneFormer [6] to generate text prompts describing partial scene configurations. Each text prompt contains one to three sentences. We explain the details of text formulation process by using the text prompt 'The room has a dining table, a pendant lamp, and a lounge chair. The pendant lamp is above the dining table. There is a stool to the right of the lounge chair.‘ as an example. First, we randomly select three objects from a scene, get their class labels, and then count the number of appearances of each selected object category. As such, we can get the first sentence. Then, we find all valid object pairs associated with the selected three objects. An object pair is valid only if the distance between two objects is less than a certain threshold that is set to 1.5 in our method. Next, we calculate the relative orientations and translations, from which we can determine the relationship type of the valid object pair from the candidate pool: 'is above to‘, 'is next to‘, 'is left of‘, 'is right of‘, ' surrounding‘, 'inside‘, 'behind‘, 'in front of‘, and 'on‘. In this way, we can acquire some relation-describing sentences like the second and third sentences in the example. Finally, we randomly sampled zero to two relation-describing sentences.

3. Baselines

DepthGAN DepthGAN [10] adopts a generative adversary network to train 3D scene synthesis using both semantic maps and depth images. The generator network is built with 3D convolution layers, which decode a volumetric scene with semantic labels. A differentiable projection layer is applied to project the semantic scene volume into depth images and semantic maps under different views, where a multi-view discriminator is designed to distinguish the synthesized views from ground-truth semantic maps and depth images during the adversarial training.

Sync2Gen Sync2Gen [9] represents a scene arrangement as a sequence of 3D objects characterized by different attributes (e.g., bounding box, class category, shape code). The generative ability of their method relies on a variational auto-encoder network, where they learn objects' relative attributes. Besides, a Bayesian optimization stage is used as a post-processing step to refine object arrangements based on the learned relative attribute priors.

ATISS ATISS [3] considers a scene as an unordered set of objects and then designs a novel autoregressive transformer architecture to model the scene synthesis process. During training, based on the previously known object attributes, ATISS utilizes a permutation-invariant transformer to aggregate their features and predicts the location, size, orientation, and class category of the next possible object conditioned on the fused feature. The original version of ATISS [3] is conditioned on a 2D room mask from the top-down orthographic projection of the 3D floor plane of a scene. To ensure fair comparisons, we train an unconditional ATISS without using a 2D room mask as input, following the same training strategies and hyperparameters as the original ATISS.

4. Ablation Studies

In main paper, we investigated the effectiveness of each design in our DiffuScene, including network architecture, loss function, and geometry feature diffusion. We present more implementation details of each method variant.

What is the effect of UNet-1D+Attention as the denoiser? We advocate the use of UNet-1D with attention layers as the denoising network. The self-attention layers within this architecture effectively aggregate all object features and explore inter-object relationships, facilitating the learning of a global context that aids in distinguishing different objects within the scene. An alternative choice is to use a pure transformer network, like the one adopted in DALLE-2 [5]. However, our comparisons revealed a marginal degradation in performance metrics such as FID, KID, SCA, and CKL. It demonstrates that UNet-1D with attention layers is more adept at capturing accurate scene distributions than networks solely composed of transformation layers.

What is the effect of multiple prediction heads in the denoiser? In our denoiser architecture, we employ three distinct encoding and prediction heads tailored for specific object properties, including bounding box parameters, semantic class labels, and geometry codes. By utilizing multiple diffusion heads with individual loss functions for each attribute (e.g., bouding box, class, geometry), we mitigate the risk of bias towards any single attribute within a single encoding and prediction head. This approach ensures that our denoiser effectively captures and processes diverse object properties without favoring one over the others. The consistent improvement in each evaluation metric verifies the effectiveness of multiple prediction heads.

What is the effect of the IoU loss? In scene diffusion models, we employ noise prediction loss as the primary supervision, focusing on attribute denoising of individual object instances. However, this loss does not address object intersections within a scene. To alleviate the issue, we augment it with pair-wise bounding box IoU loss. Quantitative comparisons indicate that incorporating IoU loss results in the synthesis of scenes with improved symmetry and enhanced plausibility, as evidenced by lower FID, KID, SCA, PIoU, and higher Sym.

What is the effect of geometry feature diffusion? To evaluate our method's performance without geometry feature diffusion, we eliminate the geometry feature encoding and prediction heads from our denoiser network. Consequently, this method only produces bounding boxes and class labels for objects within a scene. During inference, for each generated object, we conduct shape retrieval in the 3D-FUTURE [1] dataset to find the CAD model with the same class label and the closest 3D bounding box sizes. Fig. 5 of the main paper shows that our model can find symmetric nightstands by beds due to the geometry awareness of the diffusion process and shape retrieval. Table 3 in the main paper presents the comparison in the formation of symmetric pairs: 0.72 (w/ shape diffusion) vs. 0.50 (w/o shape diffusion). This highlights the effectiveness of geometry feature diffusion in achieving symmetric placements and semantically coherent arrangements. Improved plausibility in synthesis results is reflected in lower FID, KID, and SCA evaluations. Additionally, the decrease in CKL suggests that the joint diffusion of geometry code and object layout facilitates learning more similar object class distributions.

5. Additional Results

Diversity Analysis. The qualitative comparisons in Fig. 7 of the main paper and Fig. 6 illustrate that our diffusion-based method can produce more diverse results than the baseline methods. Following ATISS and LEGO, we use FID and KID to quantitatively evaluate the result diversity. We compare both the mean and covariance of generated and reference scene distribution. Additionally, we include Precision / Recall commonly used to evaluate generative models [2]. Precision is the probability that a randomly generated scene falls within the support of real scene distribution. Recall is the probability that a random scene from the datasets falls within the generated scene distribution. Tab. 1 shows that our approach outperforms all baselines in both metrics, which demonstrates better diversity, plausibility, and mode coverage.

Method	Bedroom		Dining		Living	
	Precision	Recall	Precision	Recall	Precision	Recall
DepthGAN	58.05	31.66	70.16	15.77	81.30	12.08
Sync2Gen*	55.10	67.57	70.90	47.16	75.20	52.01
Sync2Gen	59.00	67.74	76.15	33.19	77.77	48.79
ATISS	72.80	77.08	77.70	64.17	76.50	62.64
Ours	**82.31**	**77.93**	**82.80**	**78.83**	**79.30**	**70.53**

Table 1. The Precision [%] of generated scenes and Recall [%] of reference scenes. For both metrics, the higher the better.

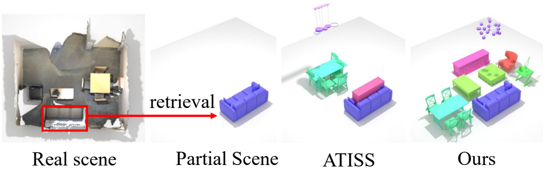


Figure 2. Scene completion of a real scene. We select a sofa and perform CAD retrieval to obtain a partial scene as input.

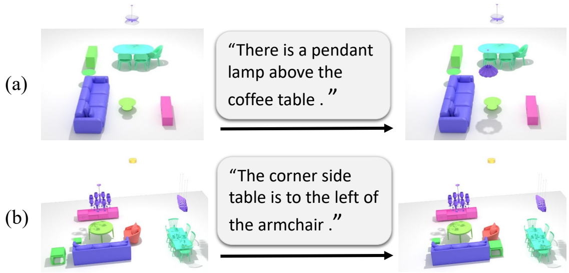

(a)
"There is a pendant lamp above the coffee table ."
(b)
"The corner side table is to the left of the armchair ."

Figure 3. Text-guided (a) object suggestion (b) scene editing.

Unconditional Scene Synthesis In Fig. 4, we provide additional qualitative comparisons against state-of-the-art methods on the unconditional scene synthesis model. Also, more visualization results of our unconditional scene synthesis model are presented in Fig. 5.

Scene Arrangement We visualize additional qualitative comparisons on the task of scene arrangement in Fig. 7. LEGO [8] aims to predict 2D object locations and orientations, taking the input of a floor plane, object semantics, and geometries. It does not handle objects like lamps that could hang from the ceiling. In contrast, DiffuScene is a scene-generative model that predicts 3D instance properties from random noise, including 3D locations and orientations, semantics, and geometries. Compared to ATISS and LEGO, our method generates various object placement options with better plausibility and more symmetries.

Scene Completion We present more qualitative comparisons on the task of scene completion in Fig. 6. Also, the quantitative results are shown in Tab. 2. Compared to ATISS, our method produced more diverse completion results with higher fidelity. Our method can consistently outperform ATISS in all listed metrics.

Room	Method	FID ↓	KID ↓	#Sym.	PIoU
Bed	ATISS	30.54	2.38	0.01	0.84
	Ours	**27.32**	**1.92**	**0.47**	**0.61**
Dining	ATISS	42.65	8.32	1.42	1.73
	Ours	**40.99**	**6.31**	**2.57**	**0.84**
Living	ATISS	43.30	5.22	0.16	0.87
	Ours	**40.49**	**4.59**	**2.24**	**0.58**

Table 2. Quantitative comparisons on the task of **scene completion** on 3D-FRONT bedrooms, dining rooms, and living rooms. Only 3 objects are given in the partial scenes.

Real-world Scene Generalization While trained on synthetic datasets, our method can be evaluated on real-world scenes without finetuning, e.g. for scene completion as shown in Fig. 2. Compared to ATISS, our method produces a more favorable scene.

Text-conditioned Scene Synthesis We provide additional qualitative comparisons on the text-conditioned scene synthesis in Fig. 8. As observed, in the first and third rows, ATISS has object intersection issues while ours does not. In the second row, our method can correctly generate a corner side table on the left of the armchair. However, ATISS generates a corner side table on the right of the armchair. In the fourth row, our method can generate four dining chairs that are consistent with the text description, but ATISS can only generate two dining chairs.

Scene editing via texts. In Fig. 3, we show that our method can support text-guided object suggestion and scene editing, without changing the attributes of other objects.

6. User Study

We conducted a perceptual user study to evaluate the quality of our method against ATISS on the application of text-conditioned scene synthesis. As shown in Fig. 9, we provide the visualization of a ground-truth scene used to generate a text prompt as a reference. For each pair of results, a user needs to answer "which of the generated scenes can better match the text prompt?" and "Which of the generated scenes is more reasonable and realistic?". We collect the answers of 225 scenes from 45 users and calculate the statistics. 62% of the user answers prefer our method to ATISS in realism. 55% of answers think our method is more consistent with the text prompt.

References

[1] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 129:3313–3337, 2021. 1, 3

[2] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[3] Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. Atiss: Autoregressive transformers for indoor scene synthesis. *Advances in Neural Information Processing Systems*, 34:12013–12026, 2021. 1, 2, 5, 6, 7, 8

[4] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1

[5] 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

[6] Xinpeng Wang, Chandan Yeshwanth, and Matthias Nießner. Sceneformer: Indoor scene generation with transformers. In *2021 International Conference on 3D Vision (3DV)*, pages 106–115. IEEE, 2021. 2

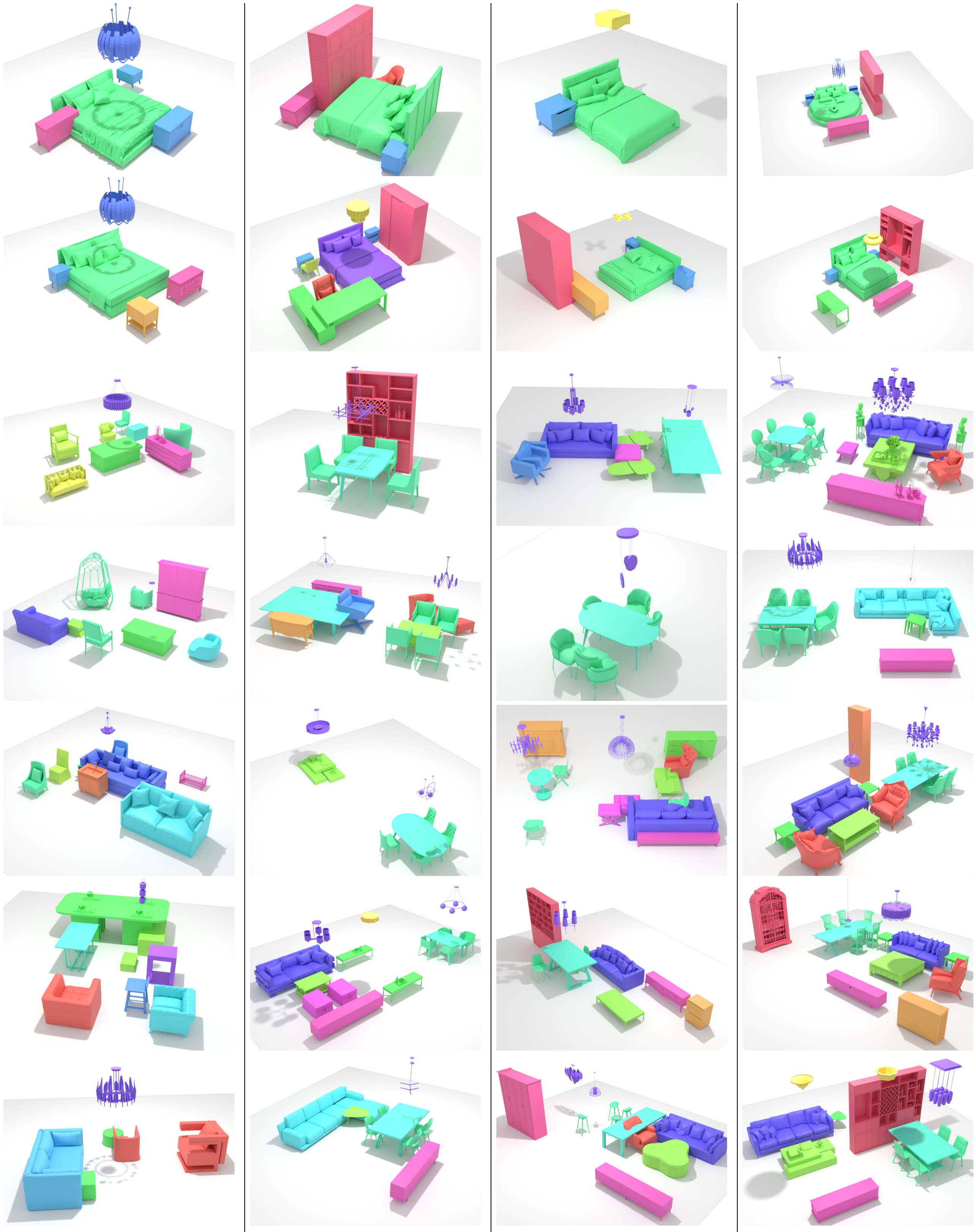
[7] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 1

[8] Qiuhong Anna Wei, Sijie Ding, Jeong Joon Park, Rahul Sajnani, Adrien Poulenard, Srinath Sridhar, and Leonidas Guibas. Lego-net: Learning regular rearrangements of objects in rooms. *arXiv preprint arXiv:2301.09629*, 2023. 3, 7

[9] Haitao Yang, Zaiwei Zhang, Siming Yan, Haibin Huang, Chongyang Ma, Yi Zheng, Chandrajit Bajaj, and Qixing Huang. Scene synthesis via uncertainty-driven attribute synchronization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5630–5640, 2021. 2, 5

[10] Ming-Jia Yang, Yu-Xiao Guo, Bin Zhou, and Xin Tong. Indoor scene generation from a collection of semantic-segmented depth images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15203–15212, 2021. 2, 5

[11] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2018. 1



(a) DepthGAN [10]

(b) Sync2Gen [9]

(c) ATISS [3]

(d) Ours

Figure 4. **Additional results of unconditional scene synthesis**. We compare our method with the state-of-the-art by generating from random noises, where our results present higher diversity and better plausibility with fewer penetration issues and more symmetric pairs.

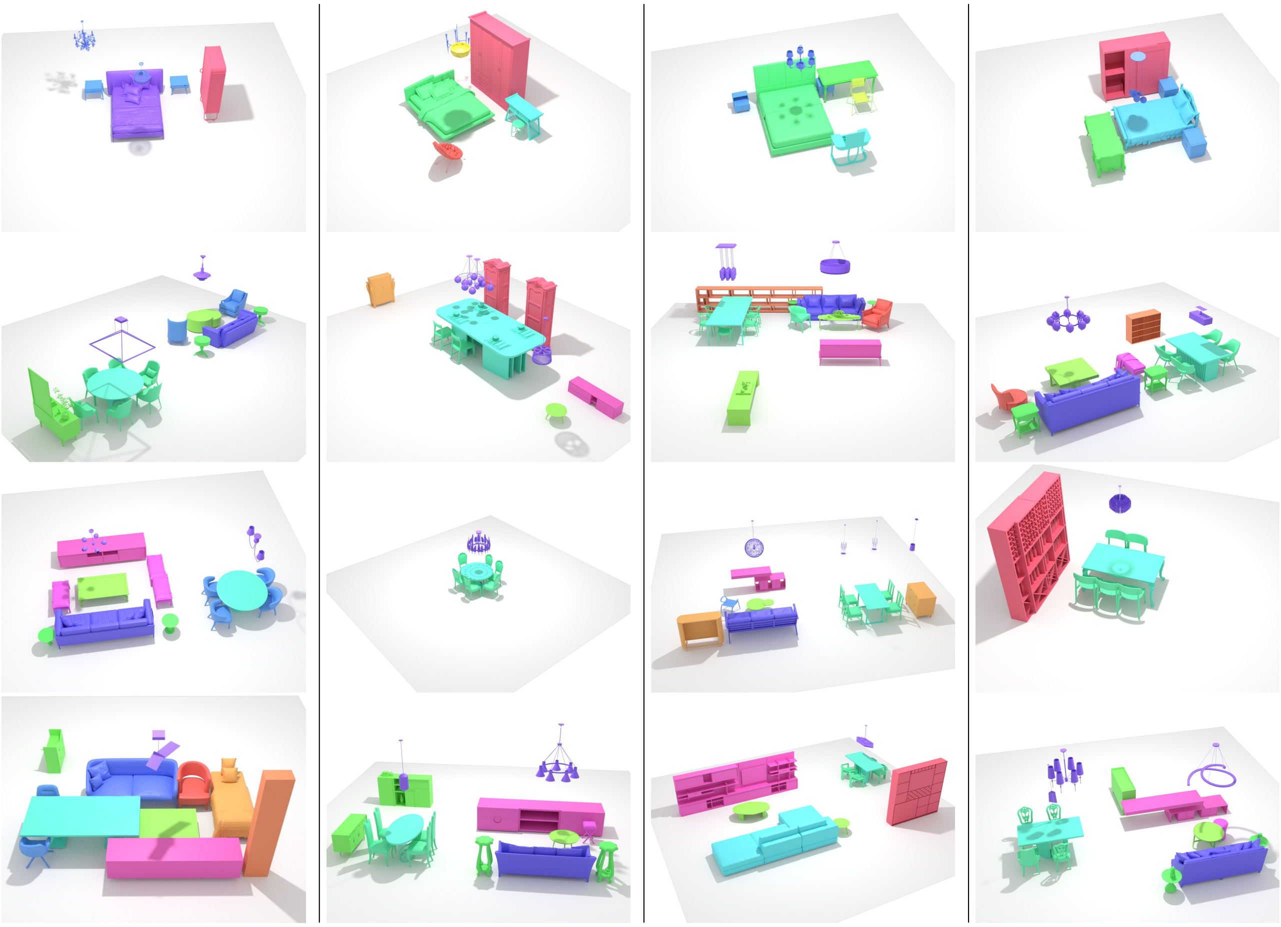
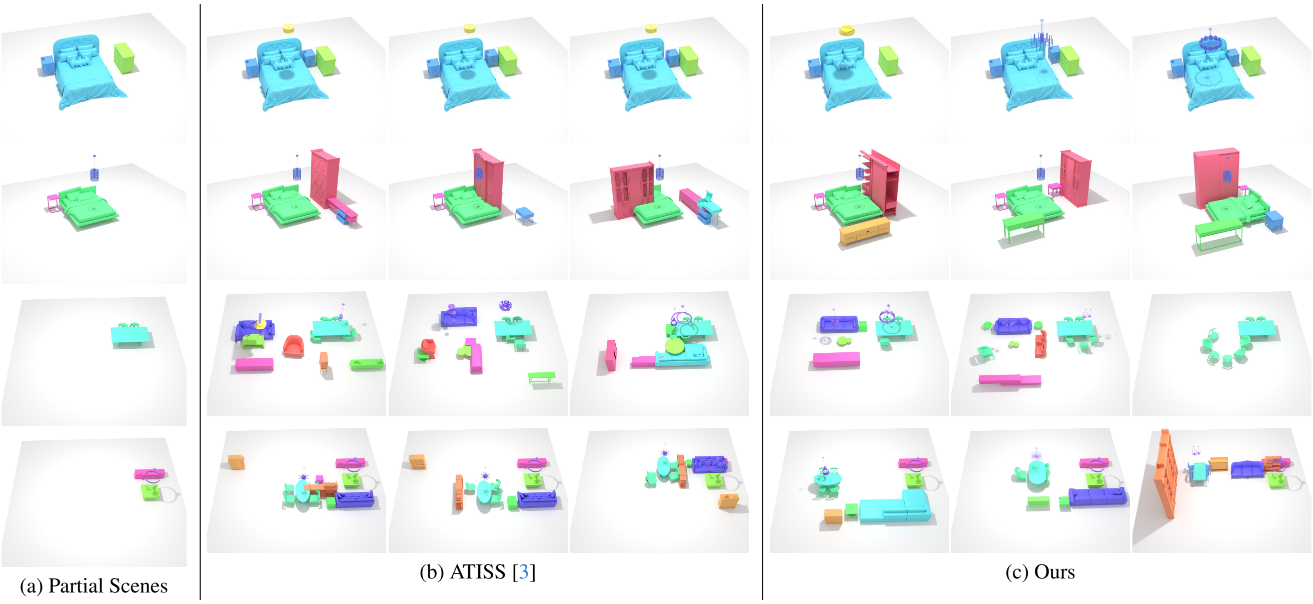


Figure 5. Diverse and plausible results of **unconditional scene synthesis from our method**.

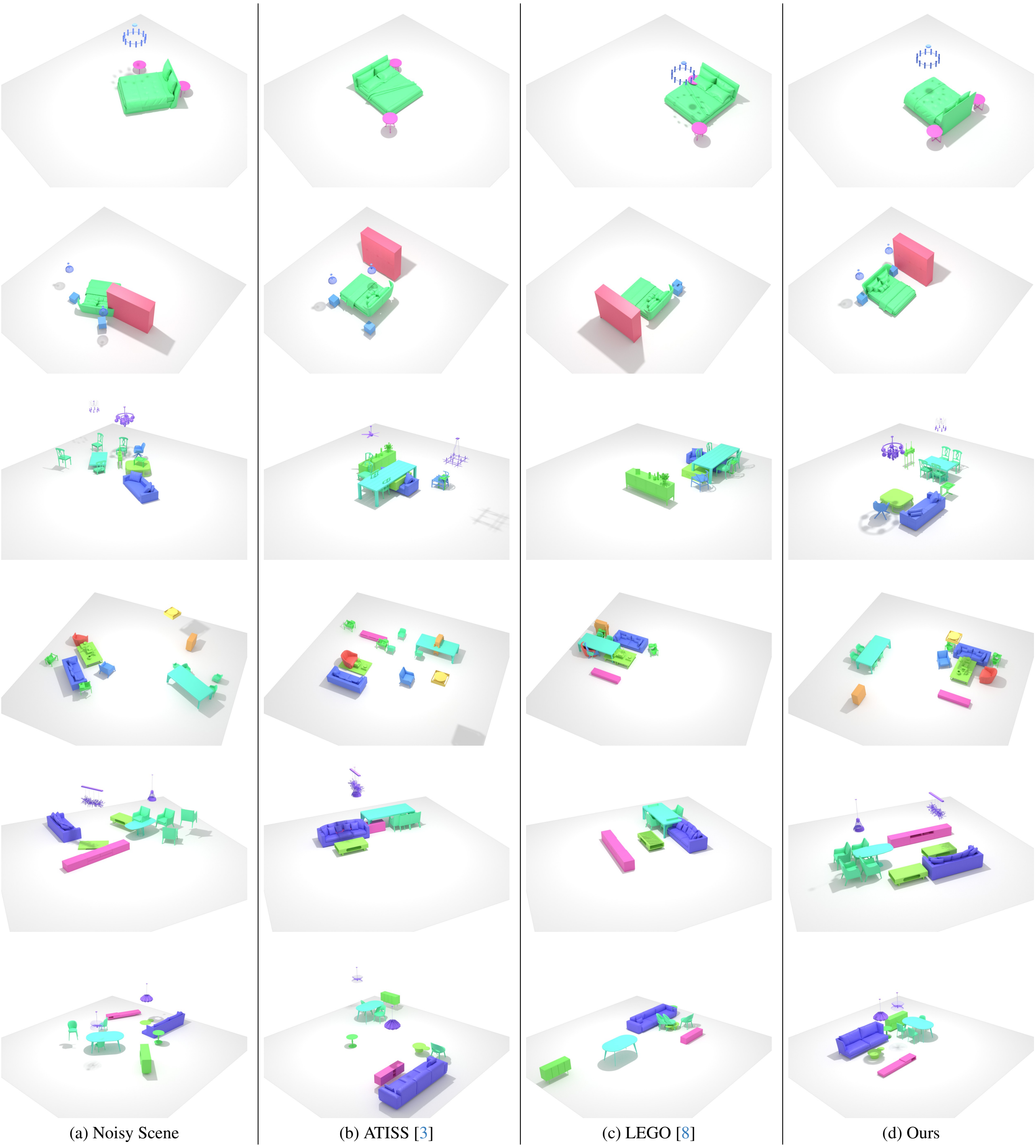


(a) Partial Scenes

(b) ATISS [3]

(c) Ours

Figure 6. **Scene completion** from partial scenes with only three objects given as inputs. Compared to ATISS, our method produced more diverse completion results with higher fidelity.



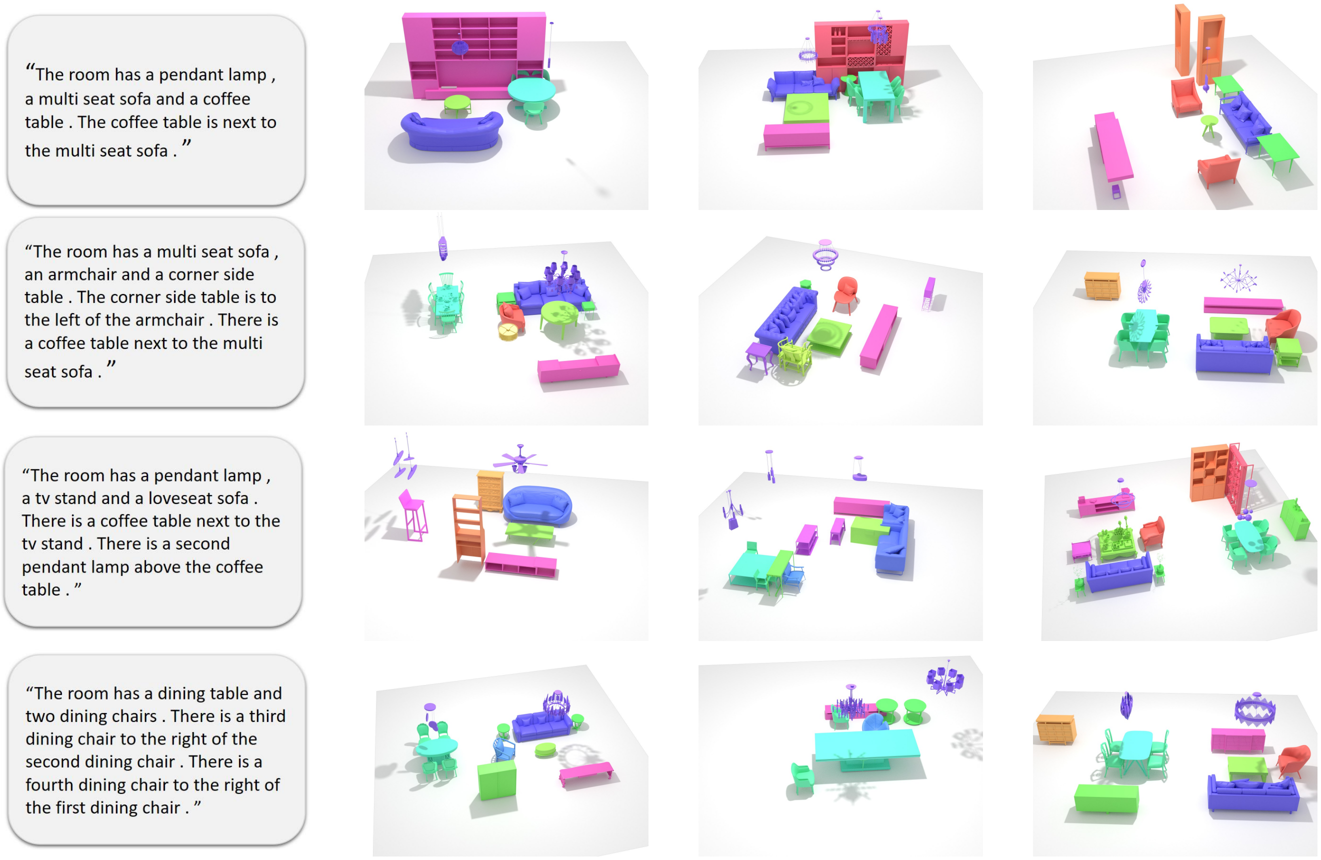
(a) Noisy Scene

(b) ATISS [3]

(c) LEGO [8]

(d) Ours

Figure 7. **Scene re-arrangements** of collections of random objects. Compared to ATISS and LEGO, our method generates various object placement options with better plausibility and more symmetries.



“The room has a pendant lamp , a multi seat sofa and a coffee table . The coffee table is next to the multi seat sofa . ”

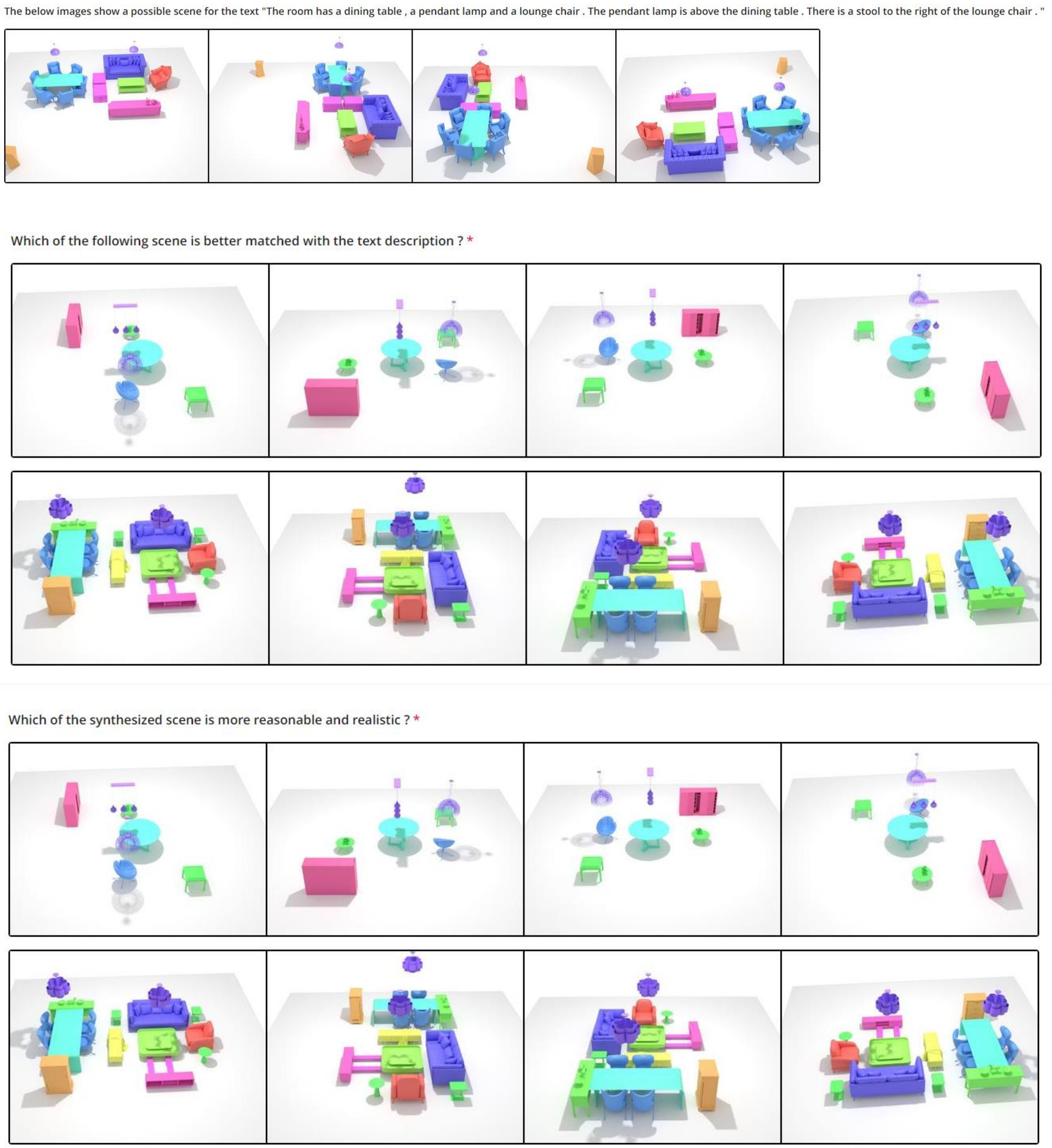
“The room has a multi seat sofa , an armchair and a corner side table . The corner side table is to the left of the armchair . There is a coffee table next to the multi seat sofa . ”

“The room has a pendant lamp , a tv stand and a loveseat sofa . There is a coffee table next to the tv stand . There is a second pendant lamp above the coffee table . ”

“The room has a dining table and two dining chairs . There is a third dining chair to the right of the second dining chair . There is a fourth dining chair to the right of the first dining chair . ”

(a) Input text (b) Reference (c) ATISS [3] (d) Ours

Figure 8. **Text-conditioned scene synthesis**. The input text describes only a partial scene configuration. Our method generates more plausible scenes matched with the texts.



The below images show a possible scene for the text "The room has a dining table , a pendant lamp and a lounge chair . The pendant lamp is above the dining table . There is a stool to the right of the lounge chair . "

Which of the following scene is better matched with the text description ? *

Which of the synthesized scene is more reasonable and realistic ? *

Figure 9. **User Study UI**. Based on the reference scene used to generate text prompts, users are asked which of the synthesized scenes is more matched with the text prompt and more realistic. Note that the results from ATISS and our method are randomly shuffled to avoid bias.