PhyScene: Physically Interactable 3D Scene Synthesis for Embodied AI

Supplementary Material

A. Algorithm Details

A.1. Details of Parameters

We introduce the details of $\hat{\alpha}_t$ in Eq. (1). Given a data sample $\mathbf{x}_0$, we can define a forward diffusion process by adding noise. Each forward diffusion process adds Gaussian noise with variance β_t on $\mathbf{x}_{t-1}$, resulting in a new variable $\mathbf{x}_t$ with distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1})$. This process can be formulated as:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \boldsymbol{\Sigma}_t = \beta_t \mathbf{I}).$$

Then we can formulate the diffusion process with

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}),$$

where $q(\mathbf{x}_{1:T})$ means we apply q repeatedly from timestep 1 to T. To simplify this process, we define $\alpha_t = 1 - \beta_t$, $\hat{\alpha}_t = \prod_{s=0}^{t} \alpha_s$, and $\boldsymbol{\epsilon}, \boldsymbol{\epsilon}_0, ..., \boldsymbol{\epsilon}_{t-1} \sim \mathcal{N}(0, \mathbf{I})$. After reparameterizing with $\hat{\alpha}_t$, we have:

$$\begin{aligned}
\mathbf{x}_t =& \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}_{t-1} \\
=& \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\
=& \sqrt{\alpha_t}(\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}_{t-2}) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\
=& \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\boldsymbol{\epsilon} \\
=& ... \\
=& \sqrt{\alpha_t\alpha_{t-1}...\alpha_1}\mathbf{x}_0 + \sqrt{1-\alpha_t\alpha_{t-1}...\alpha_1}\boldsymbol{\epsilon} \\
=& \sqrt{\hat{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\hat{\alpha}_t}\boldsymbol{\epsilon}.
\end{aligned}$$

This reflects the derivation between x_t and x_0 in Eq. (1).

A.2. Details of Reachability Guidance

As mentioned in Sec. 3, we provide the detailed algorithm for calculating the reachability guidance in Algorithm 2.

```
Algorithm 2: Reachability Guidance
Module: Reachability guidance function $\varphi_{\text{reach}}(\cdot|\mathcal{F})$,
        search algorithm $\mathbf{A}^*(\cdot)$, indicator function $\mathbb{1}(\cdot)$.
Input: Floor plan $\mathcal{F}$, 3D object bboxes $\{\boldsymbol{b}_1, ..., \boldsymbol{b}_N\}$
       where N is the number of objects, embodied agent
       's width $\boldsymbol{d}$.
//Generate gaussian cost map
$W = \mathbb{1}(\mathcal{F})$ //Init walkable area
$C = \neg\mathbb{1}(\mathcal{F}) \cdot \text{MAX\_VALUE}$ //Init cost map
for $i = 1, \cdots, N$ do
    $\boldsymbol{b}_i^{2D} = \textbf{MapTo2D}(\boldsymbol{b}_i)$
    $W = W - \mathbb{1}(\textbf{Dilate}(\boldsymbol{b}_i^{2D}, \boldsymbol{d}/2))$
    //Add Gaussian cost for each object
    $C = C + \textbf{Gaussian}(\boldsymbol{b}_i^{2D})$
end
//$A^*$ shortest path search
$\{\boldsymbol{c}_1, ..., \boldsymbol{c}_M\} = \textbf{FindConnectedArea}(W)$
$\{\boldsymbol{p}_1, ..., \boldsymbol{p}_M\} = \textbf{FindCenter}(\{\boldsymbol{c}_1, ..., \boldsymbol{c}_M\})$
//Randomly choose $\boldsymbol{p}_{start}$ and $\boldsymbol{p}_{end}$
$\textbf{Path}_{\text{shortest}} = \boldsymbol{A}^*(C, \boldsymbol{p}_{start}, \boldsymbol{p}_{end})$
$\{\boldsymbol{b}_j^{\text{agent}}\}_{j=1}^{L} = \textbf{GetAgentBox}(\textbf{Path}_{\text{shortest}})$
// Reachability Guidance
$\varphi_{\text{reach}}(\mathbf{x}|\mathcal{F}) = -\sum_{i=1}^{N}\sum_{j=1}^{L} \textbf{IoU}_{3D}(\boldsymbol{b}_i, \boldsymbol{b}_j^{\text{agent}})$
return $\varphi_{\text{reach}}(\mathbf{x}|\mathcal{F})$
```

B. Data Processing

B.1. 3D-FUTURE

The original 3D-FUTURE dataset contains object CAD models that are not watertight, which can not be used for calculating collision directly. To solve this problem for evaluating physical collision between objects, we re-mesh each object model in Blender to compute the collision rate. Some examples of re-meshed models are shown in Fig. A.1, where models on the left are original CAD models in 3D-FUTURE and those on the right are the re-meshed models. Despite the perceptual similarity between models provided and re-meshed, most provided samples contain hollows inside that forbid collision calculation.

B.2. GAPartNet

To simulate the interaction between robots and articulated objects, we build upon the object CAD models and URDF files provided in GAPartNet. Specifically, we generate the articulated object's states from close to open according to the URDF file and record the sequential process into an integrated mesh. As shown in Fig. A.2, we show the original object CAD model on the left and the integrated mesh covering articulated object states on the right. In our experiments, we use the integrated mesh to compute the collision rate between articulated objects and also use this integrated mesh to compute the opening size of articulated objects for guidance calculation.

B.3. Retrieval Categories

As our method still primarily depends on retrieving object models for generating the final scene, we combine assets from the 3D-FUTURE and GAPartNet datasets for retrieval. In Fig. A.4 we show the utilized categories in the 3D-FUTURE dataset with their corresponding asset numbers. We build a mapping between the 3D-FUTURE ob-

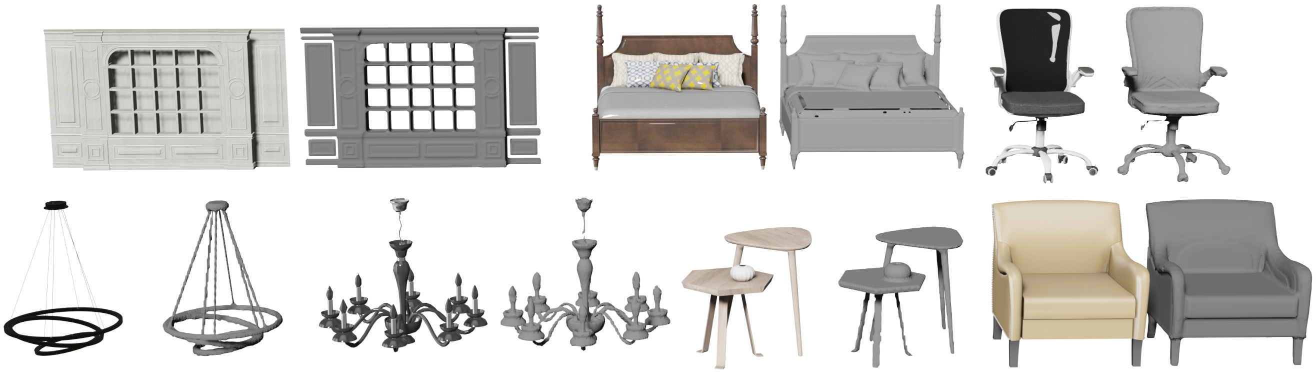


Figure A.1. **Original 3D-FUTURE model v.s. re-meshed model.** We show examples of re-meshed models. Models on the left model are the original CAD model in 3D-FUTURE, and on the right are the re-meshed models. Despite the perceptual similarity, the re-meshed models fill in the hollow area for collision calculation.

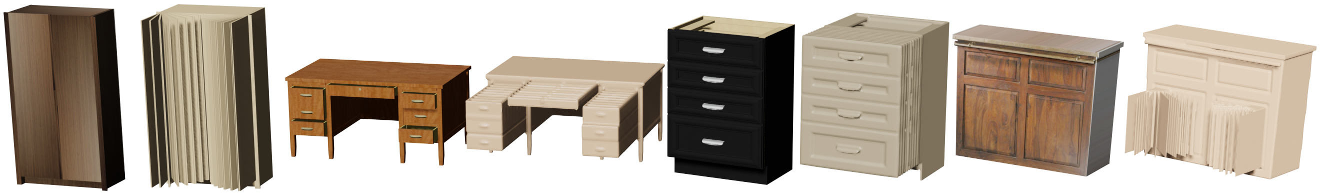


Figure A.2. **Original GAPartNet model v.s. sequential model.** The original CAD models are always in closed status. To simulate the interactive situation, we open the furniture and record the sequential process in an integrated mesh. The left model shows the original furniture, while the right one is the sequential model. We use the sequential model to compute the collision rate of articulated objects.

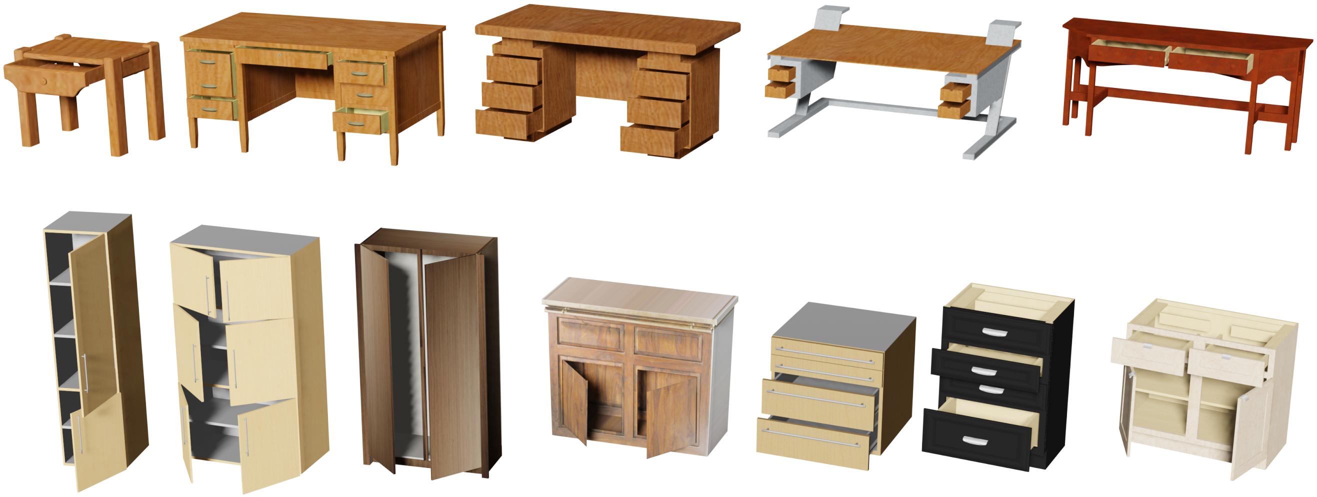


Figure A.3. **Examples of articulated objects in GAPartNet dataset.** We visualize some models of *StorageFurniture* and *Table*. The articulated models have various appearances and different joint types such as revolute and prismatic. Each piece of furniture has several joints for interaction.

ject assets and GAPartNet to align interactive categories between two datasets, such as *wardrobe* in the 3D-FUTURE, shown in orange, for the category of *StorageFurniture* in the GAPartNet. Fig. A.5 shows the category distribution of GAPartNet models, where *StorageFurniture* and *Table* take the largest proportion of this dataset. For example, the number of *StorageFurniture* is 324 out of the whole dataset number 1045. The articulated models have various appearances and different joint types such as revolute and prismatic. Each piece of furniture has several joints for inter-

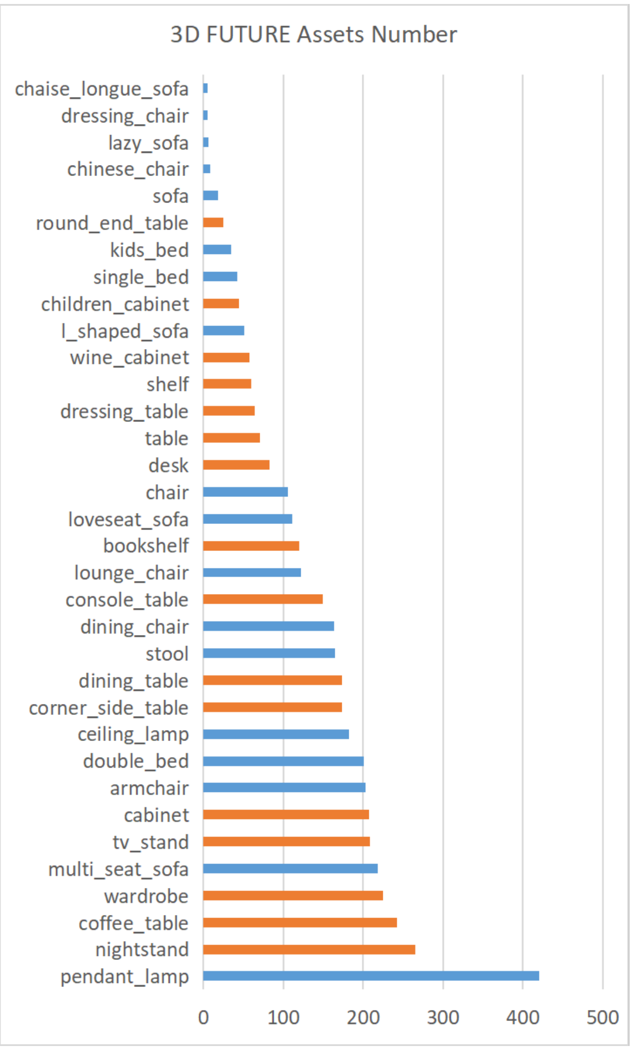


Figure A.4. **Category distribution in 3D-FUTURE dataset.** We show the utilized categories in 3D-FUTURE dataset with asset numbers. We choose interactive categories such as *wardrobe*, shown in orange, to retrieve GAPartNet model.

action. We visualize some models of *StorageFurniture* and *Table* in GAPartNet in Fig. A.3.

C. Additional Results

C.1. Physical Implausible Scenes in 3D-FRONT

As briefly discussed in Tab. 1, we provide further qualitative visualizations on the violation of physical plausibility in 3D-FRONT scene data in Fig. A.6. As shown from the visualizations, some of the scenes used for learning exhibit significant violations of physical plausibility, including object collisions and object-out-of-room scenarios.

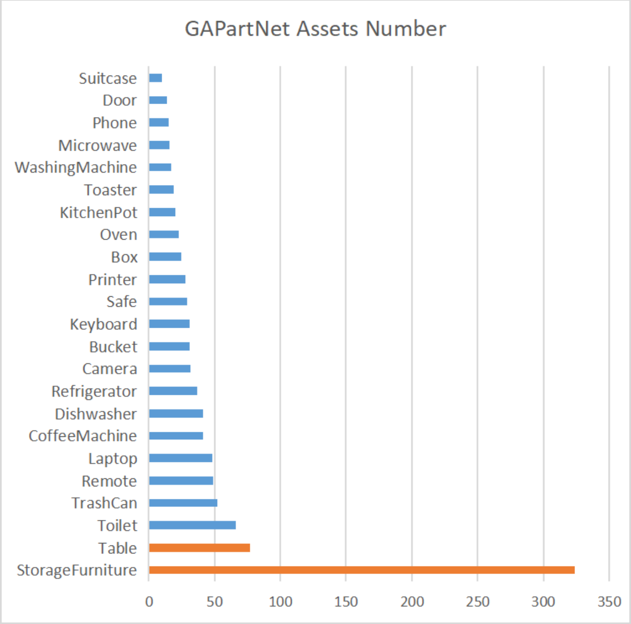


Figure A.5. **Category distribution in GAPartNet dataset.** We show the category distribution of GAPartNet model, where *StorageFurniture* and *Table* take the largest proportion of this dataset. These two categories, as shown in orange, are used to composite interactable scenes with cross-dataset retrieval.

Table A.1. **Comparison against the original 3D-FRONT dataset on collision rate**. Both ATISS and DiffuScene have higher collision rates than the 3D-FRONT dataset, while ours is lower than 3D-FRONT in most cases.

Data	Bedroom		Living Room		Dining Room	
	Col$_{obj}$	Col$_{scene}$	Col$_{obj}$	Col$_{scene}$	Col$_{obj}$	Col$_{scene}$
3D-FRONT	0.214	0.42	0.206	**0.625**	0.209	0.57
ATISS	0.248	0.46	0.316	0.85	0.591	0.96
DiffuScene	0.228	0.43	0.198	0.69	0.160	0.55
PhyScene(Ours)	**0.187**	**0.36**	**0.191**	0.63	**0.151**	**0.53**

C.2. Guidance on Different Agent Size

The reachability guidance is adaptive to different agent sizes. We use 0.2, 0.3, and 0.5 as the agent size separately, where the unit of size is the meter. We show guidance results with different agent sizes in each row and evaluate each guided result on different agent sizes (shown in each column). Here we show the guidance results in Fig. A.9 with the corresponding walkable map. It shows guidance that size 0.2 is not suitable for agent size 0.5, where the agent can only reach half of the room. And guidance on size 0.5 expands the walkable area to suit the agent in size of 0.5 and make the whole room reachable.

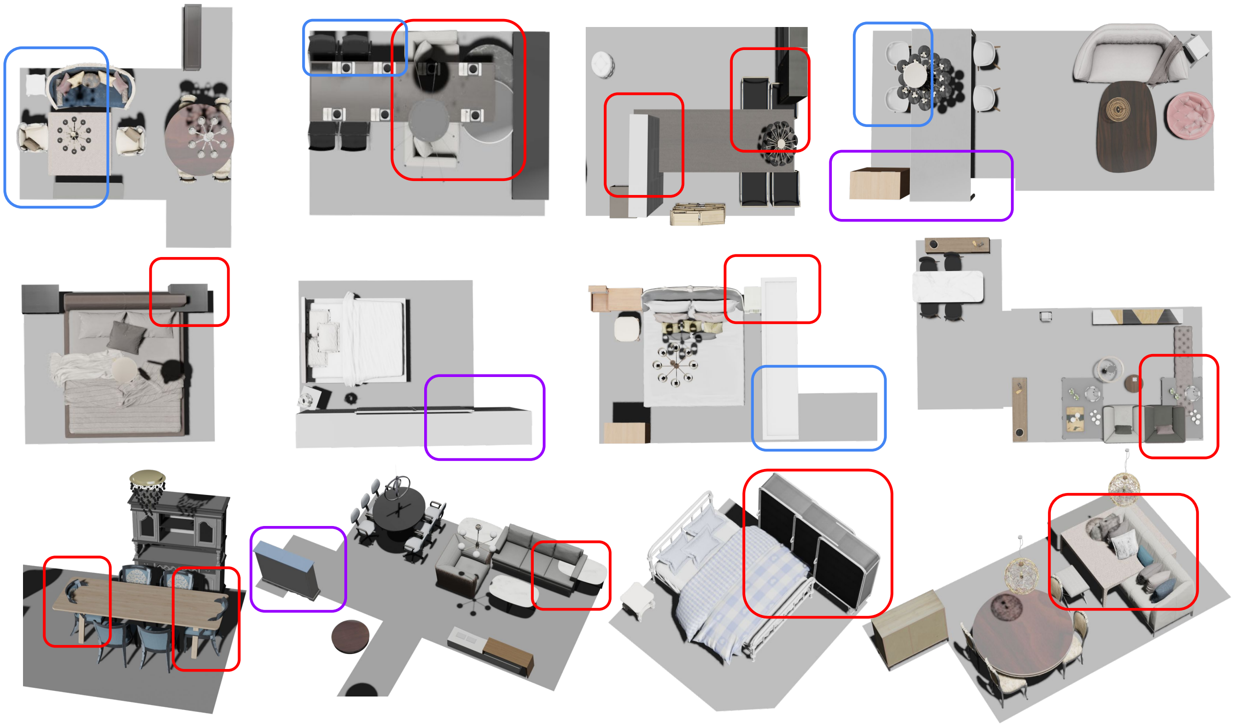
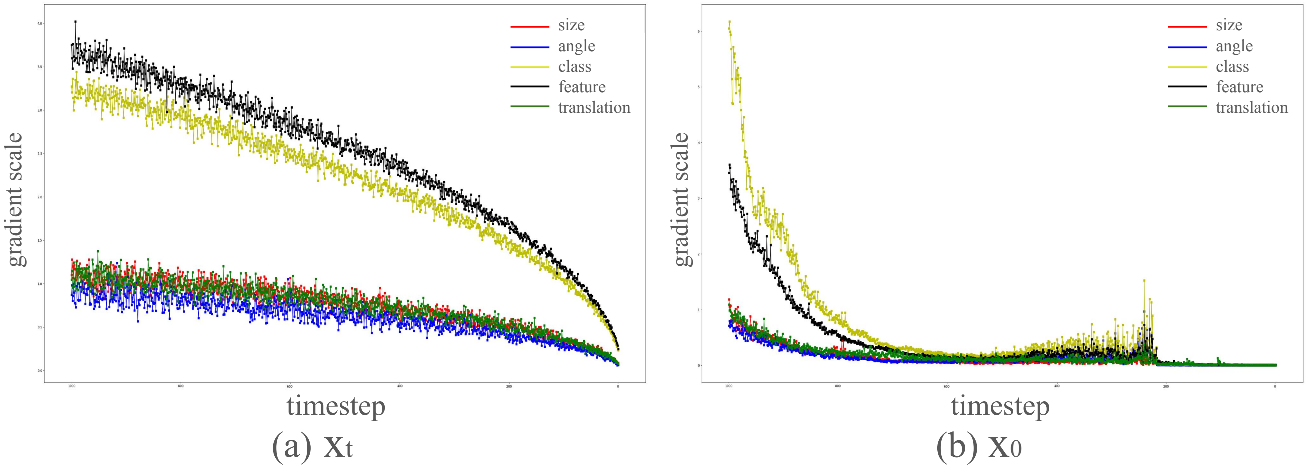


Figure A.6. **Visualization on physically implausible scenes in 3D-FRONT.** We show original 3D-FRONT scenes with physical and interactive failure cases. The **red**, **purple**, and **blue** boxes respectively indicate collisions between objects, objects outside the floor plan and unreachable areas to the embodied agent. Here we set the floor plan in gray color without texture.



(a) $\mathbf{x}_t$

(b) $\mathbf{x}_0$

Figure A.7. **Gradient scale varying with the denoising step.**

D. Comparison with 3D FRONT

Meanwhile, in Tab. A.1 we show models training on 3D-FRONT dataset can not get rid of the collision prior existed in the training dataset. Both ATISS and DiffuScene have higher collision rates on three types of rooms than 3D-FRONT. However, our PhyScene performs lower scores than 3D-FRONT. The result shows posterior optimization, such as physical and interactive guidance, is necessary to dismiss the unreasonable prior such as collision.

E. Guidance Details

We visualize the gradient scale of each denoising step in Fig. A.7. The gradient of $\mathbf{x}_t$ decreases continuously during the denoising process, while the gradient of $\mathbf{x}_0$ (predicted

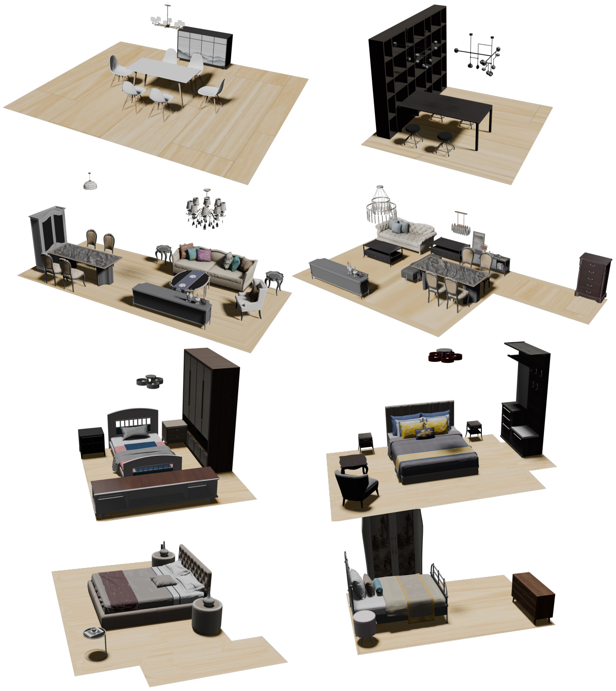


Figure A.8. **Visualization results of PhyScene on 3D Front.** The first two rows and the last two rows are the scene synthesis results of the Bedroom and Dining Room respectively.

at each step) has a rapid decline at the beginning and intensively changes in the middle stage. We visualize the layout trajectory at each step and find the layout shrinks to the vicinity of the floor plan at the beginning stage and changes from chaos to order in the middle stage. The layout finetunes itself with slight changes at the final steps. According to this observation, we add guidance on the final steps. The results also confirm that adding guidance on the final steps performs the best.

When adding guidance to the data, our guidance is calculated by bounding box, including object size, location, and angle. The purpose is to make the layout more physically plausible and interactable. So we only calculate the gradient of location and angle for guidance to move objects into a more intractable position. Noting that guiding on size will lead to rather small sizes (thickness) of objects.

F. Collision with Finer 3D Representations

In the collision guidance, we calculate the guidance objective on 3D bounding boxes of objects in Eq. (6). We have also considered other finer representations (*e.g.*, occupancy field). As the generation pipeline involves a non-differentiable object retrieval process from the generated object metadata (*i.e.*, location, scale, *etc.*), using these finer 3D representations introduces non-trivial difficulty in model optimization. Nevertheless, we tried to use bounding boxes

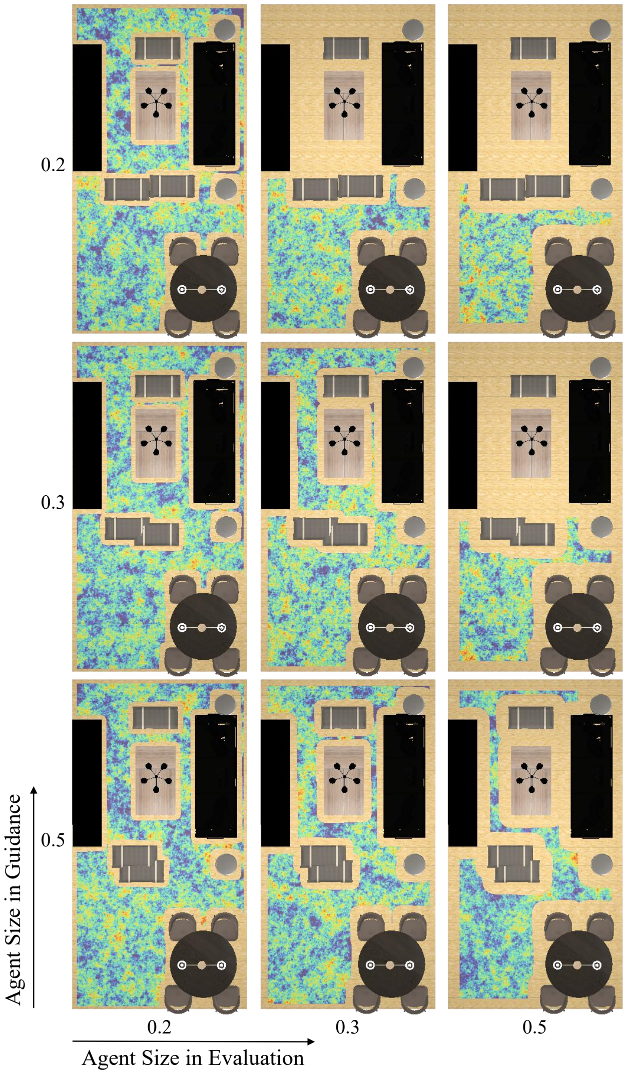


Figure A.9. **Reachability guidance results with different agent sizes.** We show the effectiveness of reachability guidance and the influence of the agent size. We compare walkable maps of different agent sizes both in guidance and in evaluation, which are 0.2, 0.3, and 0.5 separately. The unit of size is the meter.

as representations for optimization while occupancy field collisions as indicators for loss calculation, *i.e.*, using the following guidance function:

$$\varphi_{\text{coll}}(\boldsymbol{x}) = -\sum_{i,j,i\neq j} \mathbf{IoU}_{3D}(\boldsymbol{b}_i, \boldsymbol{b}_j)\mathbb{1}(\mathbf{OF}(\boldsymbol{o}_i, \boldsymbol{o}_j)),$$

where $\mathbb{1}(\mathbf{OF}(\boldsymbol{o}_i, \boldsymbol{o}_j))$ checks if two objects have collided occupancy fields. This objective penalizes bounding box collisions only for objects that are collided in their corresponding occupancy fields.

As shown in Fig. A.10, using occupancy fields as indi-

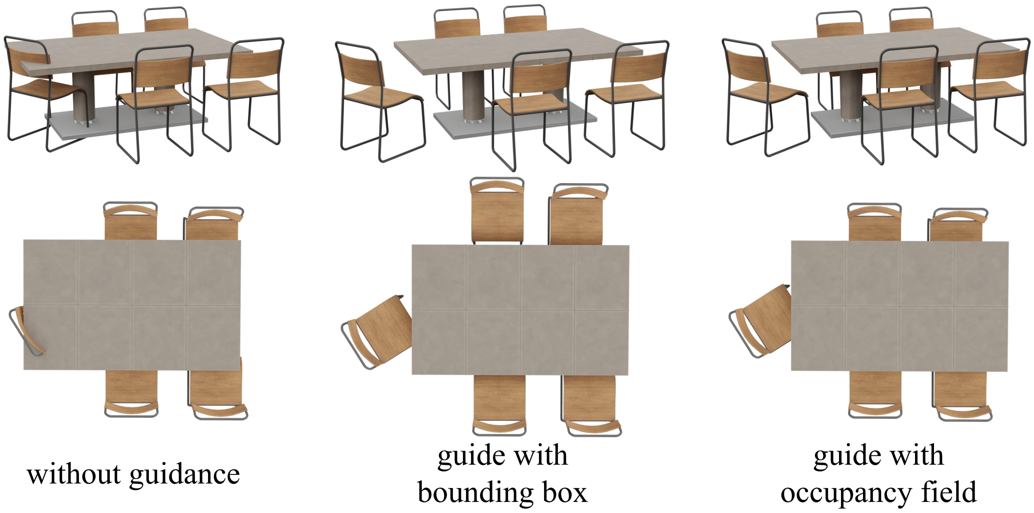


Figure A.10. **Comparison of different 3D representations in collision guidance.**

cators can slightly improve the granularity of collision considered. However, as guidance calculation is required in multiple diffusion steps, computing the collision for two occupancy fields significantly increases the computation overhead (55 times slower). Therefore, we leave this exploration to find a better balance between speed and granularity using finer 3D representations as an important future work.

G. Agent Interaction

In the reachability guidance introduced in Sec. 3.3, we only consider the walkable area as it is hard to unify guidance functions for object interactions, especially with various planners/modules required for different purposes (*e.g*., grasping, motion planning). However, as a preliminary attempt, we can extend the current pipeline to incorporate interaction constraints with proper simplifications. To ensure the articulated object interaction, we can use the same reachability guidance function while now 1) enlarging object bounding boxes to the maximum degree (fully opened) for recalculating the walkable map, 2) planning the shortest path from a walkable position to the end position of interactable object parts (*e.g*., drawer handles), and 3) applying the guidance to move the obstacle objects on this path. Similarly, we can model other interactions with rigid objects (*e.g*., sit) by planning the shortest path to the interactive areas (*e.g*., space in front of the chair) correspondingly in the guidance function.

With this simplified estimate, we can improve the *interactiveness rate* (measured by whether robots could reach the end position of object parts when being maximum interacted) from 0.101 to 0.143. Given our flexible synthesize-with-guidance designs, we believe more fine-grained and effective constraints could be seamlessly integrated into the generation pipeline and will continue to explore this topic in the future.

H. Diffusion v.s. Transformer

ATISS uses an autoregressive model with an end vector to stop predicting new furniture, while we find the object number might be very large, such as predicting 33 objects in a bedroom. In contrast, the diffusion model uses a fixed number of vectors and generates the objects' layout together. The predicted objects are embedded with overall information about the entire scene as well as inter-object relationships.

I. Additional Visualization

We provide additional qualitative visualization for the effectiveness of guidance functions in Fig. A.11. We also conduct experiments with basic floor plans (*i.e*., rectangles) in rooms from ProcTHOR and generate scenes with articulated objects. We provide the visualization of the generated results in Fig. A.12.

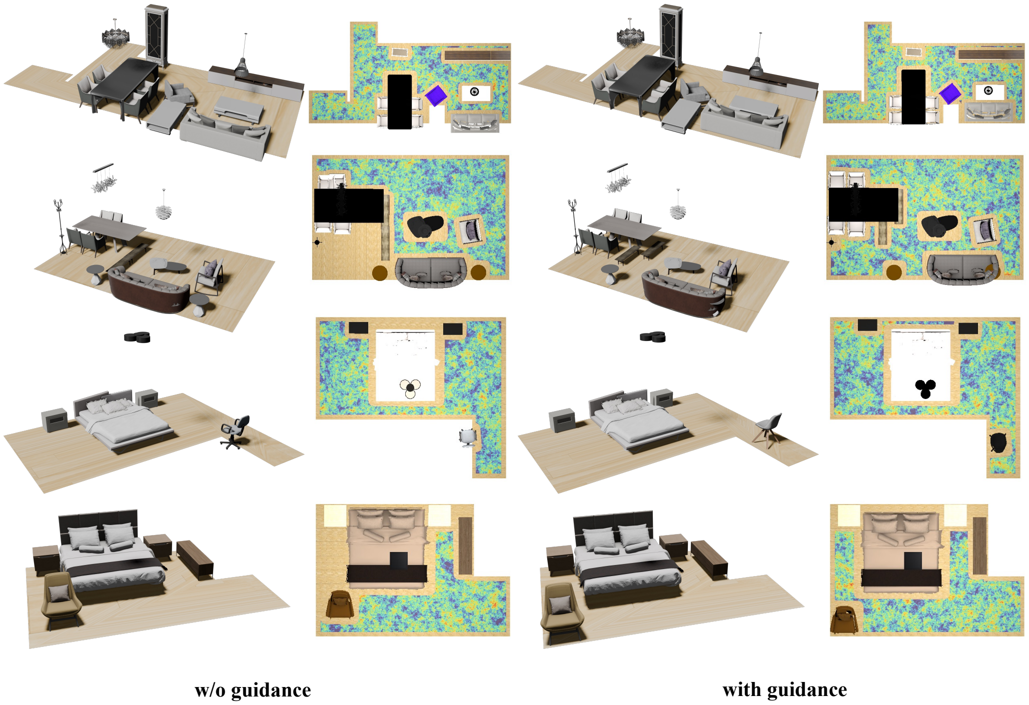


Figure A.11. **Comparison of PhyScene synthesis without and with guidance.** The first two columns and the last two columns are the scene synthesis results without and with guidance respectively.

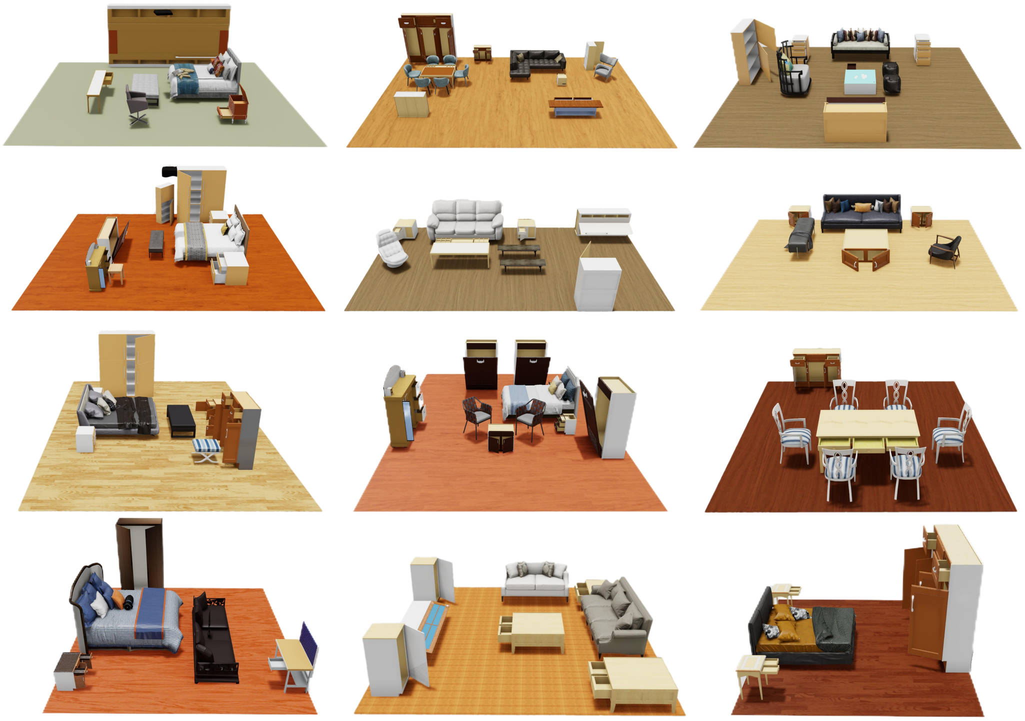


Figure A.12. **Generated scenes with articulated objects.** We show scene synthesis results with diverse layouts and random floor textures. Each scene is embedded with several articulated objects.