Scene Synthesis via Uncertainty-Driven Attribute Synchronization – Supplementary Material

Haitao Yang[1] Zaiwei Zhang[1] Siming Yan[1] Haibin Huang[2] Chongyang Ma[2]
Yi Zheng[2] Chandrajit Bajaj[1] Qixing Huang[1]
[1]The University of Texas at Austin [2]Kuaishou Technology

Our supplementary material provides details of the functions in prior modeling and joint hyperparameter optimization in Section 1, additional details on datasets and network architecture in Section 2, as well as more experiment results and analysis in Section 3.

1. Details of Modeling the Objective Function for Synchronization

1.1. GGMM in Prior Modeling

We mainly focus on translation when modeling the prior $P_c(\boldsymbol{a}_v)$ and $P_{(c,c')}(\phi(\boldsymbol{a}_v, \boldsymbol{a}_{v'}))$ because we notice that the rotation and size predicted by the network are accurate and can be directly utilized in synchronization. We model the prior for each dimension of the translation separately. A 6-component GMM is utilized to model $M_{\mu_c}(\boldsymbol{a}_v)$ for each class.

$P_{(c,c')}(\phi(\boldsymbol{a}_v, \boldsymbol{a}_{v'}))$ is modeled by an 8-component GMM multiplied by a mask which equals to one when objects of class c and class c' could penetrate (e.g., a chair and a desk) and $I_{(c,c')}(\boldsymbol{a}_v, \boldsymbol{a}_{v'})$ otherwise, where

$$I_{(c,c')}(\boldsymbol{a}_v, \boldsymbol{a}_{v'}) = \exp(\max\{0, \frac{\boldsymbol{s}_v + \boldsymbol{s}_{v'}}{2} - |\boldsymbol{t}_{v'} - \boldsymbol{t}_v|\}) \tag{1}$$

We use a 2-component GMM to fit $P_c(\boldsymbol{z}_{\mathcal{V}_c})$ and a 4-component GMM to fit $P_{(c,c')}(\boldsymbol{z}_{\mathcal{V}_c}, \boldsymbol{z}_{\mathcal{V}_{c'}})$.

Figure 1 shows an example of the distribution of translation. Figures 3 and 4 show examples of the distribution of numbers of objects.

1.2. Joint Hyperparameter Optimization

The regularization term $l(\Phi)$ is designed to learn hyperparameters of the prior distribution from the training set $\mathcal{T}_{train} := \{\boldsymbol{t}_{\mathcal{V}_c}, \boldsymbol{t}_{\mathcal{E}_{(c,c')}}\}$, where $\boldsymbol{t}_{\mathcal{V}_c}$ collects absolute translation of vertices belonging to the class c, $\boldsymbol{t}_{\mathcal{E}_{(c,c')}}$ collects relative translation of vertex pairs belonging to class c and c'. To learn the translation prior term, we utilize $l_1(\Phi)$ based on maximum likelihood estimation:

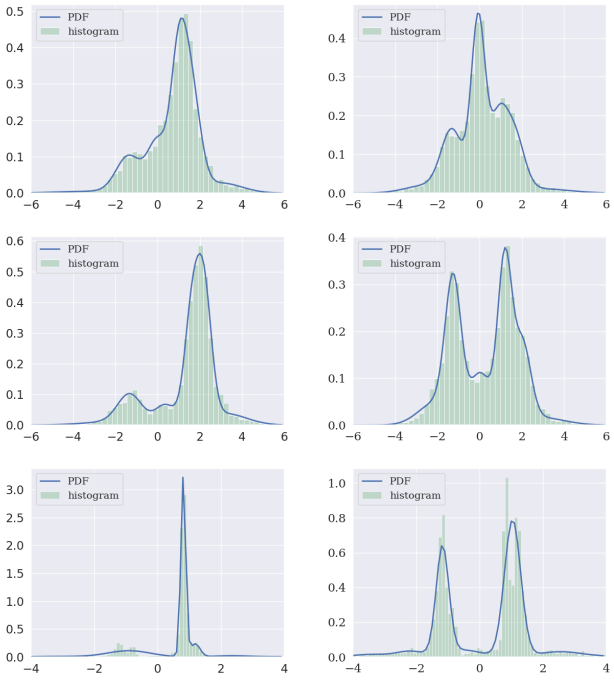

Figure 1: Distribution of translation in the SUNCG dataset. Top: absolute translation of the bed. Middle: absolute translation of the stand. Bottom: relative translation between the bed and the stand. Left: x coordinates. Right: y coordinates.

$$l_1(\Phi) = -\sum_{c\in\mathcal{C}}\sum_{\boldsymbol{t}_v\in\boldsymbol{t}_{\mathcal{V}_c}} \log M_{\mu_c}(\boldsymbol{t}_v) - \sum_{c,c'\in\mathcal{C}}\sum_{\boldsymbol{t}_e\in\boldsymbol{t}_{\mathcal{E}_{(c,c')}}} \log M_{\mu_{(c,c')}}(\boldsymbol{t}_e)$$

For object count prior, we first compute the discrete distribution p_c and $p_{(c,c')}$ from the dataset. Here

$$p_c(i) = n_{c,i}/n_{scenes}, \quad i \in \{0, 1, \cdots, N_c\}$$

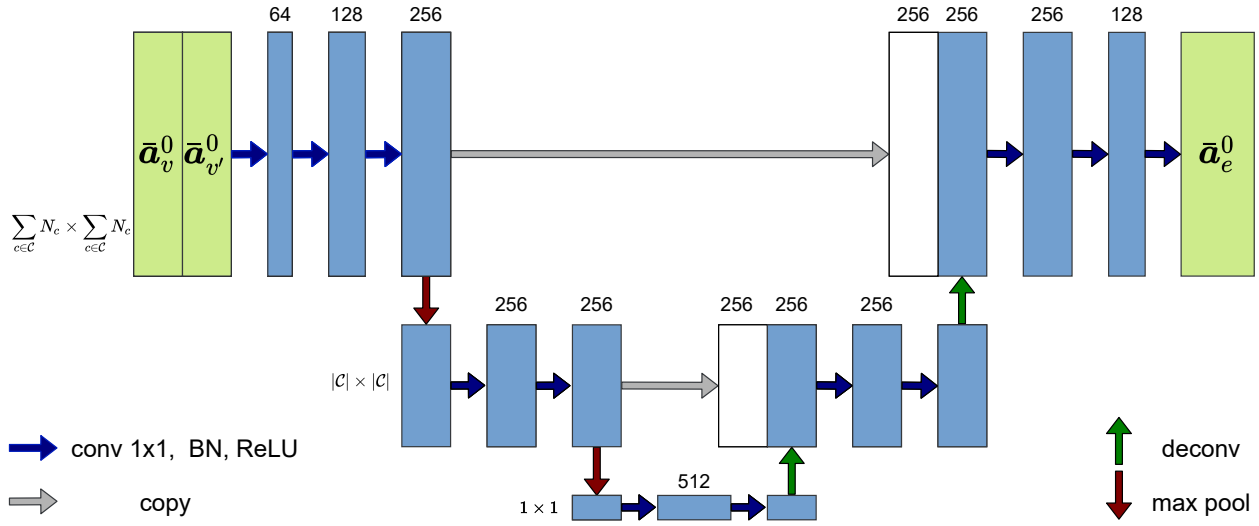

Figure 2: Architecture of the relative attribute prediction network.

Figure 3: Modeling $P_c(\boldsymbol{z}_{\mathcal{V}_c})$ with GMM in the SUNCG bedroom dataset. Left: the bed. Right: the stand. From the statistical distribution, it is most likely that there exist 1 bed and 2 stands in each scene.

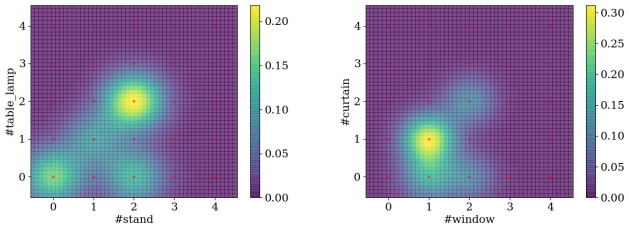

Figure 4: Modeling $P_{(c,c')}(\boldsymbol{z}_{\mathcal{V}_c}, \boldsymbol{z}_{\mathcal{V}_{c'}})$ with GMM in the SUNCG bedroom dataset. Left: the stand and the table lamp. It is most likely that the number of stands and that of table lamps are equal, i.e. they co-exist. Right: the window and the curtain.

where n_{scenes} is the total number of scenes, and $n_{c,i}$ is the total number of scenes with i objects of class c. Likewise,

$$p_{c,c'}(i,j) = n_{c,c',i,j}/n_{scenes}, \quad i,j \in \{0,1,\cdots,N_c\}$$

where $n_{c,c',i,j}$ is the number of scenes with i objects of class c and j objects of class c'. Then we fit the discrete distribution with continuous function using least square:

$$l_2(\Phi) = \sum_{c\in\mathcal{C}} \sum_{\{(i,p_c(i))\}} \|M_{\gamma_c}(i) - p_c(i)\|^2 + \sum_{c,c'\in\mathcal{C}} \sum_{\{((i,j),p_{(c,c')}(i,j))\}} \|M_{\gamma_{(c,c')}}((i,j)) - p_{(c,c')}(i,j))\|^2 \tag{2}$$

The total regularization term:

$$l(\Phi) = l_1(\Phi) + l_2(\Phi) \tag{3}$$

2. Dataset and Network Architecture

2.1. Preprocessing of Datasets

For the 3D-FRONT dataset, we select two scene types: bedroom (Bedroom, MasterBedRoom and SecondBedRoom) and living room (LivingRoom and LivingDivingRoom). We fisrt filter out scenes with width and length larger than 8 meters. Then for bedroom and living room, we remove scenes whose number of objects is smaller than 6 and 4 respectively. Finally, for both scene types, we randomly sample 4000 scenes for training and about 100 scenes for validation. For the SUNCG dataset, we perform joint scene alignment following [2] and use 1000 scenes for validation.

2.2. Network Architecture Details

The network architecture for the relative attribute prediction module is shown in Figure 2. Every pair of absolute attributes $\bar{a}_v^0$ and $\bar{a}_{v'}^0$ is concatenated as input to the network. The network outputs the refined relative attributes.

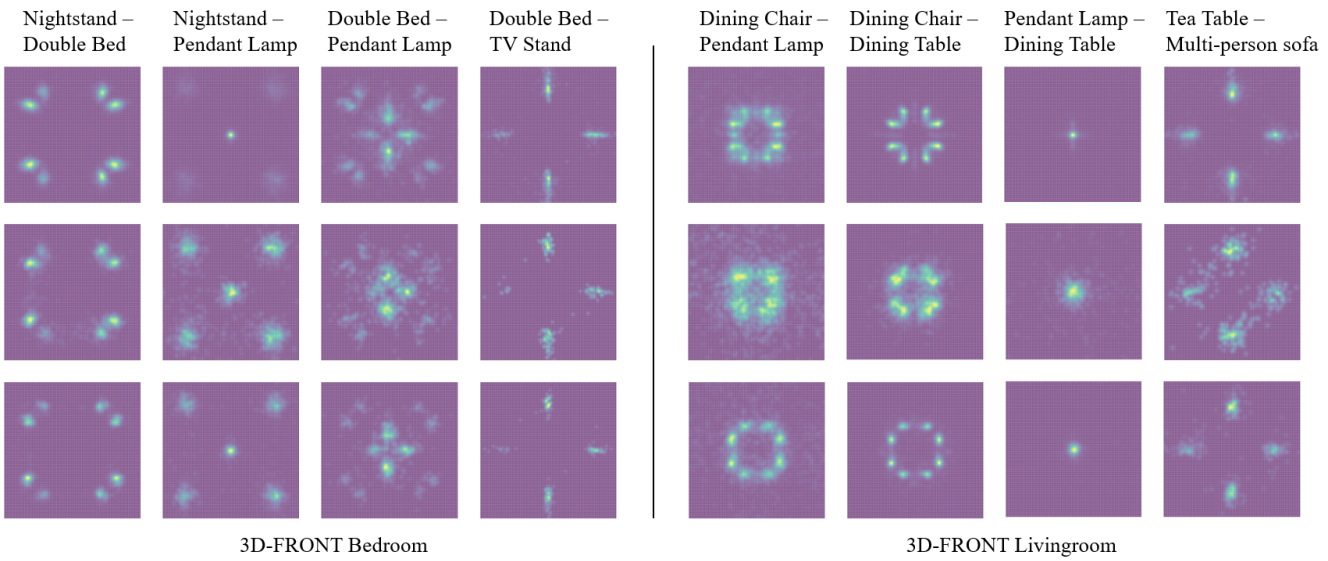

Figure 5: Distributions of relative translation in the 3D-FRONT dataset. Top: distribution of the training data. Middle: distribution derived from predicted absolute parameters. Bottom: distribution derived from the optimized absolute parameters after synchronization.

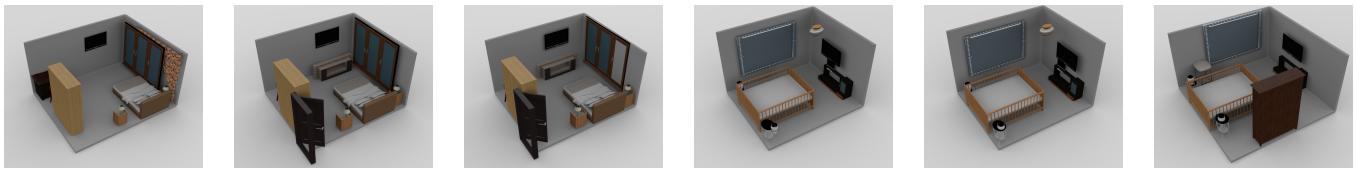

Figure 6: Left: output of the prediction module. Middle: scene optimization without using relative attribute predictions. Right: full pipeline. The relation between the bed and the nightstand, the TV and the TV stand are more realistic when utilizing the relative attributes in scene optimization.

Figure 7: Comparison between off-the-shelf scene optimization techniques. Left: output of the prediction module. Middle: scene optimization using off-the-shelf techniques. Right: our method.

3. More Experimental Results and Analysis

3.1. Visual Comparisons between Our Method and Baseline Approaches

Figure 8 and Figure 9 show more visual comparisons between our approach and baseline approaches. We can see that our approach generates more reasonable scenes than the baselines.

3.2. More Analysis on Distribution of Relative Attributes

Figure 5 shows the distributions of relative translation in the 3D-FRONT dataset. Again we can see the improvements of relative translation, which benefit from incorporating predicted relative attributes and prior modeling.

3.3. Importance of Utilizing Relative Attributes

Figure 6 shows the comparison between scene optimization with/without utilizing relative attributes. We can see from the results that merely using prior distribution for optimization easily gets stuck in local minimums and can not improve the relative position well.

3.4. Comparison between Off-the-shelf Scene Optimization Techniques

Figure 7 shows the comparison between our approach and off-the-shelf techniques [1]. Given the same initial scene, both approaches can achieve reasonable arrangements. However, by incorporating the prior distribution and relative attributes into the optimization pipeline, our approach can remove redundant objects (e.g. the laptop on the TV stand) and add diverse objects (e.g. the chair and the cabinet).

References

[1] Lap-Fai Yu, Sai-Kit Yeung, Chi-Keung Tang, Demetri Terzopoulos, Tony F. Chan, and Stanley J. Osher. Make it home: Automatic optimization of furniture arrangement. *ACM Trans. Graph.*, 30(4), July 2011. 3

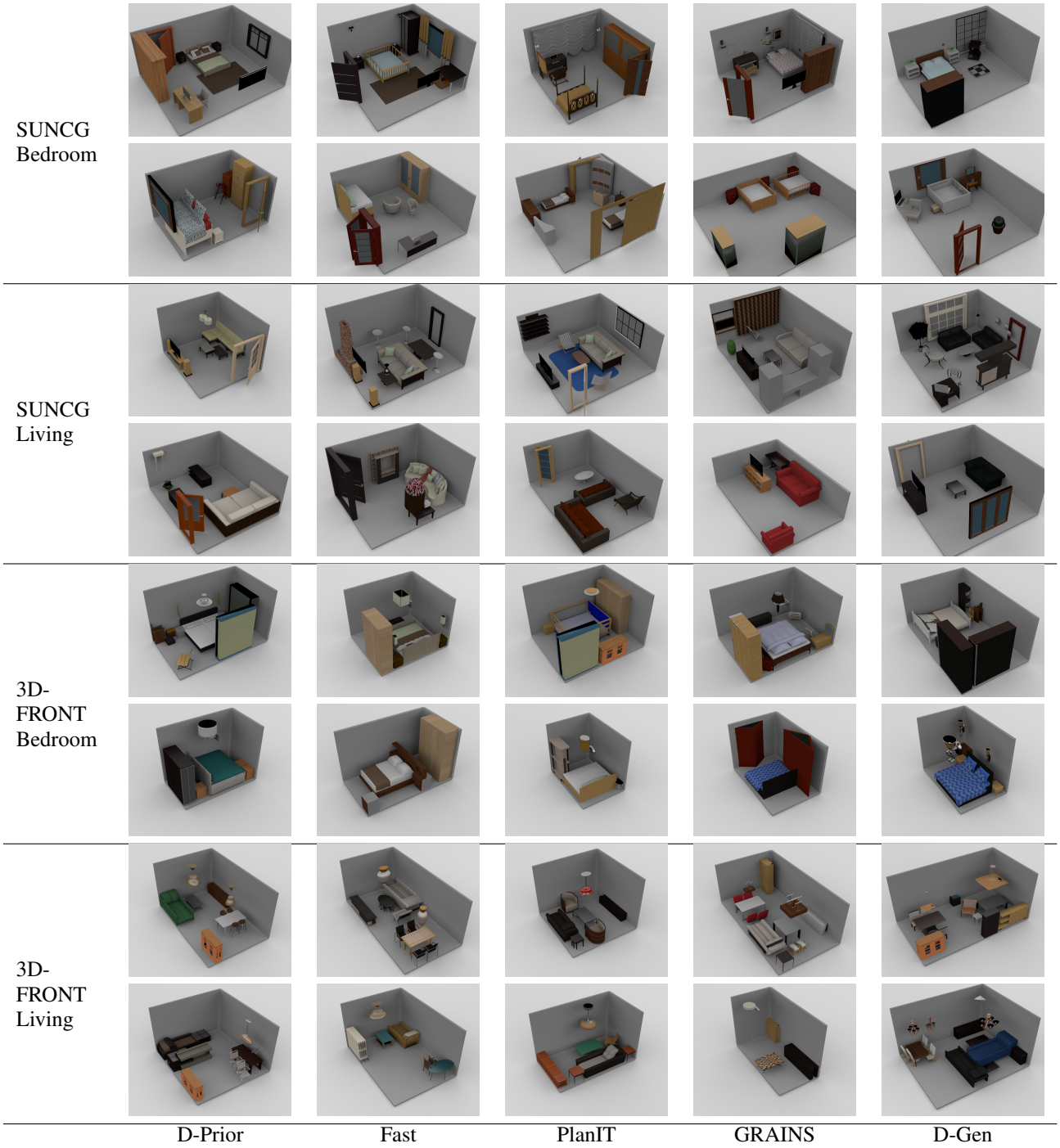

Figure 8: Scene synthesis results between ours and baseline methods. For each dataset, the first row: our results, the second row: baseline methods. For baseline methods, from left to right: D-Prior, Fast, PlanIT, GRAINS, D-Gen.

[2] Zaiwei Zhang, Zhenpei Yang, Chongyang Ma, Linjie Luo, Alexander Huth, Etienne Vouga, and Qixing Huang. Deep generative modeling for scene synthesis via hybrid representations. *ACM Trans. Graph.*, 39(2):17:1–17:21, 2020. 2

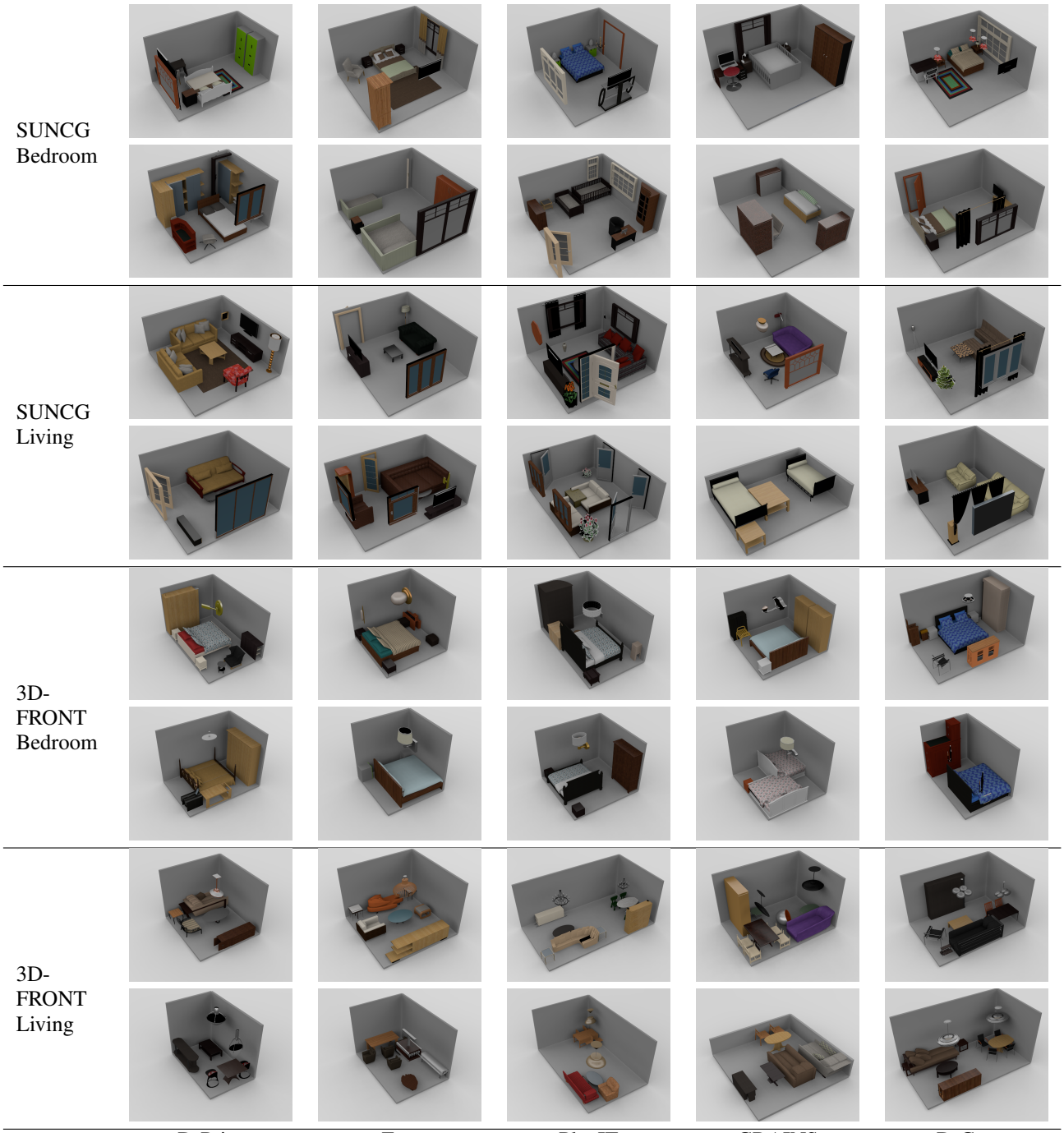

Figure 9: Scene synthesis results between ours and baseline methods. For each dataset, the first row: our results, the second row: baseline methods. For baseline methods, from left to right: D-Prior, Fast, PlanIT, GRAINS, D-Gen.