I-Scene: 3D Instance Models are Implicit Generalizable Spatial Learners

Supplementary Material

1. Overview

To better illustrate our method, the supplementary material is organized into two main parts: **Method** and **Experiments**.

Method.
- **Scene-context attention.** We provide a mathematical derivation showing that when the scene and instance inputs are identical, the proposed scene-context attention (SCA) is equivalent to a standard self-attention layer.
- **Collision-free random layout.** We describe the details of our collision-free random layout generation procedure.

Experiment setting.
- Model implementation details.
- Training datasets.
- Metrics. We provide the full specification of our customized robust ICP metric and related details.

Qualitative results.
- Qualitative comparison with all baselines on synthetic scenes. We either (i) show interactive 3D scene comparisons on the web page, or (ii) include two representative rendered views per scene for each method.
- Qualitative comparison with all baselines on real-world and stylized images, following the same visualization protocol as above.
- Additional qualitative results for the **ablation** study.
- **Failure cases**. We report the failure cases when instance mask is small. The figure including the input image, the predicted scene, and the predicted low-quality instance examples.

2. Method

scene-context attention. In this section, we show the mathematic proof that when the K_i/V_i is the same with the scene K_s/V_s, meaning when the scene and instance input is exact the same, the SCA gives equivalent result of self-attention layers.

Let $Q_i \in \mathbb{R}^{t\times d}$ be the (row-stacked) instance queries, $K_i, K_\mathrm{s} \in \mathbb{R}^{n\times d}$ the keys, and $V_i, V_\mathrm{s} \in \mathbb{R}^{n\times d_v}$ the values. We concatenate along the token dimension,

$$\tilde{K}_i = [K_i;\, K_\mathrm{s}] \in \mathbb{R}^{2n\times d}, \qquad \tilde{V}_i = [V_i;\, V_\mathrm{s}] \in \mathbb{R}^{2n\times d_v},$$

and define scene-context attention (SCA) by

$$\mathrm{SCA}(Q_i, \tilde{K}_i, \tilde{V}_i) = \mathrm{softmax}\Big(\tfrac{Q_i \tilde{K}_i^\top}{\sqrt{d}}\Big)\, \tilde{V}_i,$$

where $\mathrm{softmax}(\cdot)$ is applied *row-wise* across the key dimension. We show that when the scene and instance inputs coincide, i.e., $K_i = K_\mathrm{s}$ and $V_i = V_\mathrm{s}$, SCA reduces exactly to standard self-attention:

$$\mathrm{SCA}(Q_i, \tilde{K}_i, \tilde{V}_i) = \mathrm{softmax}\Big(\tfrac{Q_i K_\mathrm{s}^\top}{\sqrt{d}}\Big) V_\mathrm{s}.$$

Proposition. If $K_i = K_\mathrm{s}$ and $V_i = V_\mathrm{s}$, then

$$\mathrm{SCA}(Q_i, [K_i;\, K_\mathrm{s}], [V_i;\, V_\mathrm{s}]) = \mathrm{softmax}\Big(\tfrac{Q_i K_\mathrm{s}^\top}{\sqrt{d}}\Big) V_\mathrm{s}.$$

Proof. Under $K_i = K_\mathrm{s}$ and $V_i = V_\mathrm{s}$, write $K \coloneqq K_\mathrm{s}$ and $V \coloneqq V_\mathrm{s}$. Then $\tilde{K}_i = [K;\, K]$ and $\tilde{V}_i = [V;\, V]$. Let

$$Z \coloneqq \frac{Q_i K^\top}{\sqrt{d}} \in \mathbb{R}^{t\times n}.$$

By block structure,

$$\frac{Q_i \tilde{K}_i^\top}{\sqrt{d}} = \frac{Q_i [K;\, K]^\top}{\sqrt{d}} = \begin{bmatrix} Z & Z \end{bmatrix} \in \mathbb{R}^{t\times 2n}.$$

Consider any row $z \in \mathbb{R}^n$ of Z. The row-wise softmax over the concatenation $[z, z] \in \mathbb{R}^{2n}$ yields

$$\mathrm{softmax}([z, z]) = \Big[\tfrac{1}{2}\,\mathrm{softmax}(z) \;\; \tfrac{1}{2}\,\mathrm{softmax}(z)\Big],$$

because for each coordinate j, with $s \coloneqq \sum_{\ell=1}^n e^{z_\ell}$ we have

$$\frac{e^{z_j}}{\sum_{\ell=1}^n e^{z_\ell} + \sum_{\ell=1}^n e^{z_\ell}} = \frac{e^{z_j}}{2s} = \tfrac{1}{2}\ \mathrm{softmax}(z)_j.$$

Applying this row-wise to $[Z, Z]$ gives

$$\mathrm{softmax}([Z, Z]) = \begin{bmatrix} \frac{1}{2}S & \frac{1}{2}S \end{bmatrix}, \quad \text{where}$$

$$S \coloneqq \mathrm{softmax}(Z) \in \mathbb{R}^{t\times n}.$$

Therefore,

$$\begin{aligned} \mathrm{softmax}([Z, Z])\, \tilde{V}_i &= \begin{bmatrix} \frac{1}{2}S & \frac{1}{2}S \end{bmatrix} [V;\, V] \\ &= \tfrac{1}{2}SV + \tfrac{1}{2}SV \\ &= SV \\ &= \mathrm{softmax}\Big(\tfrac{Q_i K^\top}{\sqrt{d}}\Big)\, V. \end{aligned} \tag{1}$$

This is precisely the output of a standard self-attention layer evaluated on (Q_i, K, V). ∎

Collision-free Random Layout. When we randomly create a layout, we first randomly sample N instances from the instance object dataset (e.g. Objaverse). Then we generate collision-free layouts in a Poisson noise pattern by treating each object i as a 2D disc on the ground with radius $r_i = s_i \hat{r}_i$, where $\hat{r}_i$ is computed from the mesh x/z extents and s_i is the sampled scale. Centers are sampled with a variable-radius Poisson-disk routine that places larger radius first. We define a global clearance $\text{gap} = \bar{r} \cdot s$, where $\bar{r}$ is the mean of r_i and s is a randomly sampled scaling factor that controls the layout density. A candidate center $\mathbf{x}_i$ is accepted only if $\|\mathbf{x}_i - \mathbf{x}_j\|_2 \geq r_i + r_j + \text{gap}$ for all placed j. We accelerate checks with a uniform grid of cell size $(\min_i r_i + \text{gap})/\sqrt{2}$, retry up to a fixed budget, and expand the sampling region by 10% if needed. For the stacked object, we slice the table mesh just below its top to extract the top polygon, use its centroid as the anchor, and choose a scale that fits inside based on the distance to the nearest edge, then place the stacked object on the table.

3. Experiment

Model implementation. All experiments are conducted on a cluster with 8 NVIDIA H100 GPUs. We train *I-Scene* for 130K steps using the AdamW optimizer with a learning rate of $5e^{-5}$ and a batch size of 8 per GPU. During inference, we adopt 25 sampling steps with the classifier-free guidance set to $\omega = 3.0$ for both the sparse structure generation and structured latents generation.

Training dataset The 3D-FRONT training dataset is the same as MIDI processed 3D-FRONT dataset, where there are roughly 24K different scenes. Using the random layout generation algorithm discussed above, randomly generate 15K and 25K scenes. Each scene has minimum 2 object and maximum 12 objects.

For each scene, we use Blender to render 150 views that evenly distributed from all directions looking at the scene center. Then we transform each instance into view centric space, voxelize and encode each instance geometry using TRELLIS (with 64^3 voxel grid) sparse structure encoder as ground truth. Note, in some view some object is fully occluded and not visible, we discard this render view to avoid the model memorize and hallucinate invisible objects.

Robust ICP. Previous work MIDI [2] uses ICP for alignment before calculating the metrics. We notice this ICP is not robust and leads to many bad alignment, making the metric calculation not reliable. Instead, we propose a robust ICP to make the metric results more reliable.

The main reason ICP is not robust is it is very easy to get stuck into local minimum. To handle this challenge, we used several ways to escape local minimum.

Initial transform search. We perform a yaw-sweep global initialization about a designated up axis ($a \in \{x, y, z\}$). For a candidate set of yaw angles (default $\{0, 45, 90, 135, 180, 225, 270, 315\}$ degrees), we rotate the source downsampled point cloud about a, and pre-score each angle using a trimmed symmetric Chamfer distance with trim ratio τ=0.2 on up to 2000 sampled points per cloud. We keep the top three yaw candidates by this pre-score and run a short seed ICP on the downsampled clouds for each candidate using a point-to-point estimator (optionally with isotropic scale if enabled). We select the initialization T_0 that minimizes rmse $+ \lambda(1 - \text{fitness})$ with λ set to the voxel size (λ=v).

Coarse-to-fine. *Shared normalization.* For numerical stability and consistency across stages, we apply the same uniform normalization to source and target: let c be the midpoint of the axis-aligned bounding box over both clouds and σ the maximum side length. We work in normalized coordinates $x' = (x - c)/\sigma$ for the remainder of the registration.

Downsampling and normals. We voxel downsample both normalized clouds with voxel size $v = 0.03$ (relative to the normalized extent) for the coarse stage, and estimate normals on both downsampled and full-resolution clouds when available.

Estimators. The coarse stage uses a point-to-point objective; if isotropic scale is enabled and supported, we estimate a global uniform scale jointly with the rigid motion. The fine stage uses point-to-plane with a Tukey robust loss (scale $k = 1.5v$); if normals or robust losses are unavailable, we fall back to point-to-point.

Iteration budgets and thresholds. We split the budget evenly with safeguards: $T_{\text{coarse}} = \max(10, \lfloor 0.5\, T \rfloor)$ and $T_{\text{fine}} = \max(10, T - T_{\text{coarse}})$. Distance thresholds are $2.5v$ (coarse) and v for point-to-plane fine refinement (or $1.5v$ if point-to-point is used).

Execution. Coarse ICP is run on the downsampled pair from T_0, yielding $T^{\star}_{\text{coarse}}$. If isotropic scale was estimated, we pre-apply $T^{\star}_{\text{coarse}}$ to the full-resolution source and run fine ICP from identity, then compose the results; otherwise, we run fine ICP on the full-resolution pair initialized with $T^{\star}_{\text{coarse}}$.

Pose projection and validation. We project the 3×3 rotation block to the nearest element of $\text{SO}(3)$ via SVD. If reflections are disallowed, we enforce a positive determinant. We then perform sanity checks on the transform in normalized space (finite entries, reasonable determinant when rigid, translation magnitude not excessive). On detection of anomalies we fall back to the coarse solution or identity, whichever is valid.

Practical notes. As different methods generate in different space with various scale, this introduce an subtle bias. Even if we align the space by min/max values, the scene size potentially still affects the results. For example, rotating an extreme long scene a little bit potentially has different scene scale after normalization. So we apply the robust ICP alignment in this way, say we have two space: (i). min/max normalize space: just scale the scene space by the default output range, e.g. TRELLIS outputs into [-0.5, 0.5], then we just scale the scene isotropically by 2; (ii). AABB re-centered normliaze: we move the scene into zero center and then normalize AABB min/max into [-1.0, 1.0]. We apply robust ICP and obtain $transform_1$ and $transform_2$. We then pick the best metric results from the two transformations.

Spatial or physical metrics. The Main Tab. 1 reports spatial metrics including scene IoU/CD/F-score, which quantify global coherence. We add object IoU to quantify fine-grained spatial relation and physical metrics (object/scene collision rate defined in Scenethesis) to the Table 1.

Table 1. Object IoU and object/scene collision rates (Col-O/S).

	Gen3DSR	PartCrafter	SceneGen	MIDI	***I-Scene***
IoU-O ↑	0.07	0.04	0.09	0.24	**0.47**
Col-O ↓	10.5%	14.5%	55.3%	30.3%	**6.1%**
Col-S ↓	30%	45%	95%	65%	**15%**

4. Qualitative results.

We present the qualitative results for the following items:

- Figure 1 and Figure 2 present multi-view qualitative comparison between *I-Scene* and all baselines using real-world /stylized images as input.
- Figure 3 presents multi-view qualitative comparison between *I-Scene* and all baselines using synthetic scene images as input.
- The visualization of ablation study can be found in Figure 4, where adding each component (SCA, VC, and NS) in *I-Scene* would significantly improve the layout coherent and instance quality.
- We also present the failure cases in Figure 5. As shown in the figure, instance quality is bad when the input instance mask is small in the image.

References

[1] Andreea Ardelean, Mert Özer, and Bernhard Egger. Gen3dsr: Generalizable 3d scene reconstruction via divide and conquer from a single view. *arXiv preprint arXiv:2404.03421*, 2024. 4, 5, 6

[2] Zehuan Huang, Yuan-Chen Guo, Xingqiao An, Yunhan Yang, Yangguang Li, Zi-Xin Zou, Ding Liang, Xihui Liu, Yan-Pei Cao, and Lu Sheng. Midi: Multi-instance diffusion for single image to 3d scene generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23646–23657, 2025. 2, 4, 5, 6

[3] Yuchen Lin, Chenguo Lin, Panwang Pan, Honglei Yan, Yiqiang Feng, Yadong Mu, and Katerina Fragkiadaki. Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers. *arXiv preprint arXiv:2506.05573*, 2025. 4, 5, 6

[4] Yanxu Meng, Haoning Wu, Ya Zhang, and Weidi Xie. Scenegen: Single-image 3d scene generation in one feedforward pass. *arXiv preprint arXiv:2508.15769*, 2025. 4, 5, 6

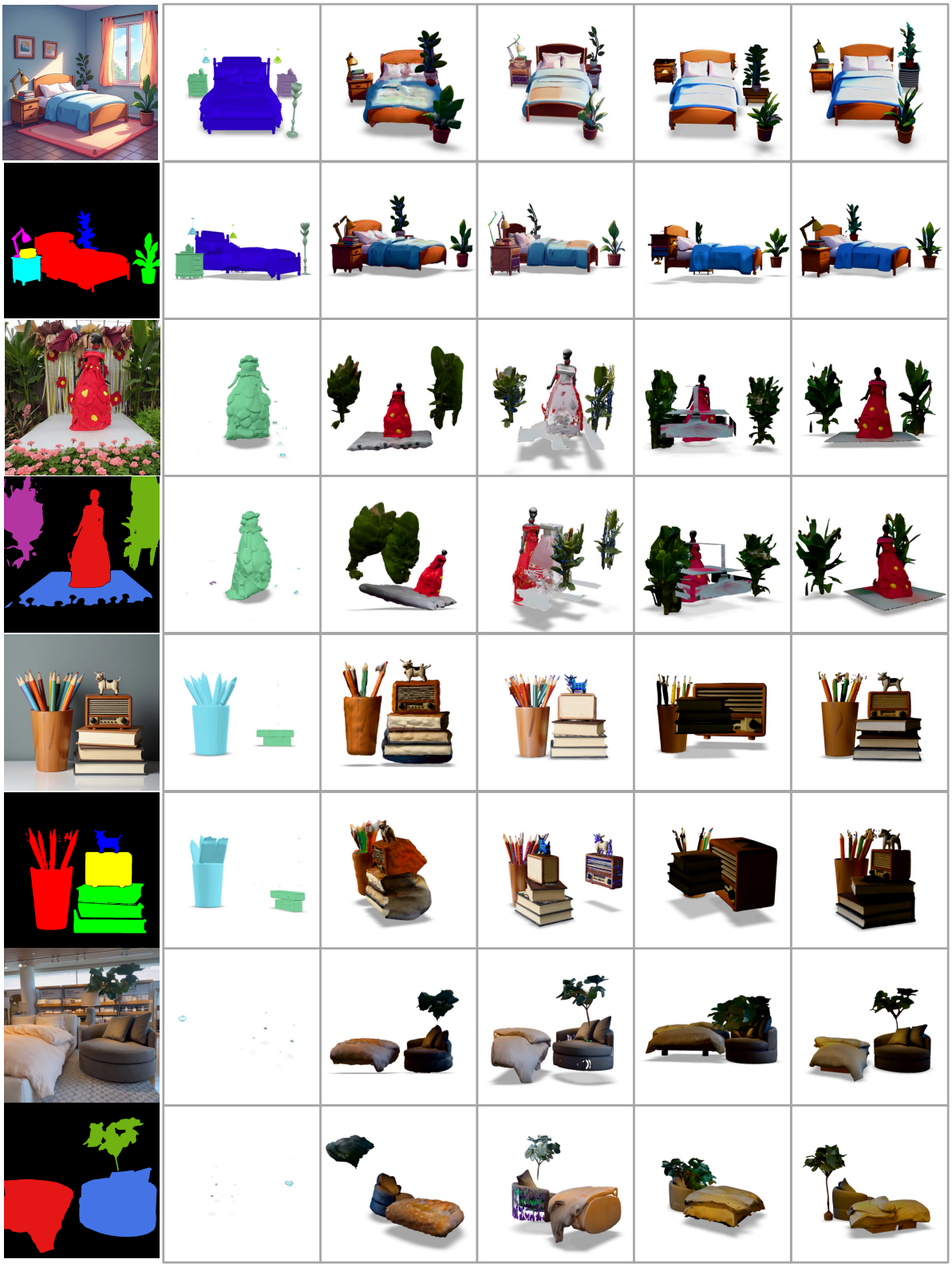

Figure 1. We present the multi-view illustration results using real-world /stylized image as input (example 1). The testing scenes contain various layouts including *front, back, right, left, small on large, behind*, and etc. Baselines includes PartCfrater [3], Gen3DSR [1], MIDI-3D [2], and SceneGen [4]

Figure 2. We present the multi-view illustration results using real-world /stylized image as input (example 2). The testing scenes contain various layouts including *front, back, right, left, small on large, behind*, and etc. Baselines includes PartCfrater [3], Gen3DSR [1], MIDI-3D [2], and SceneGen [4]

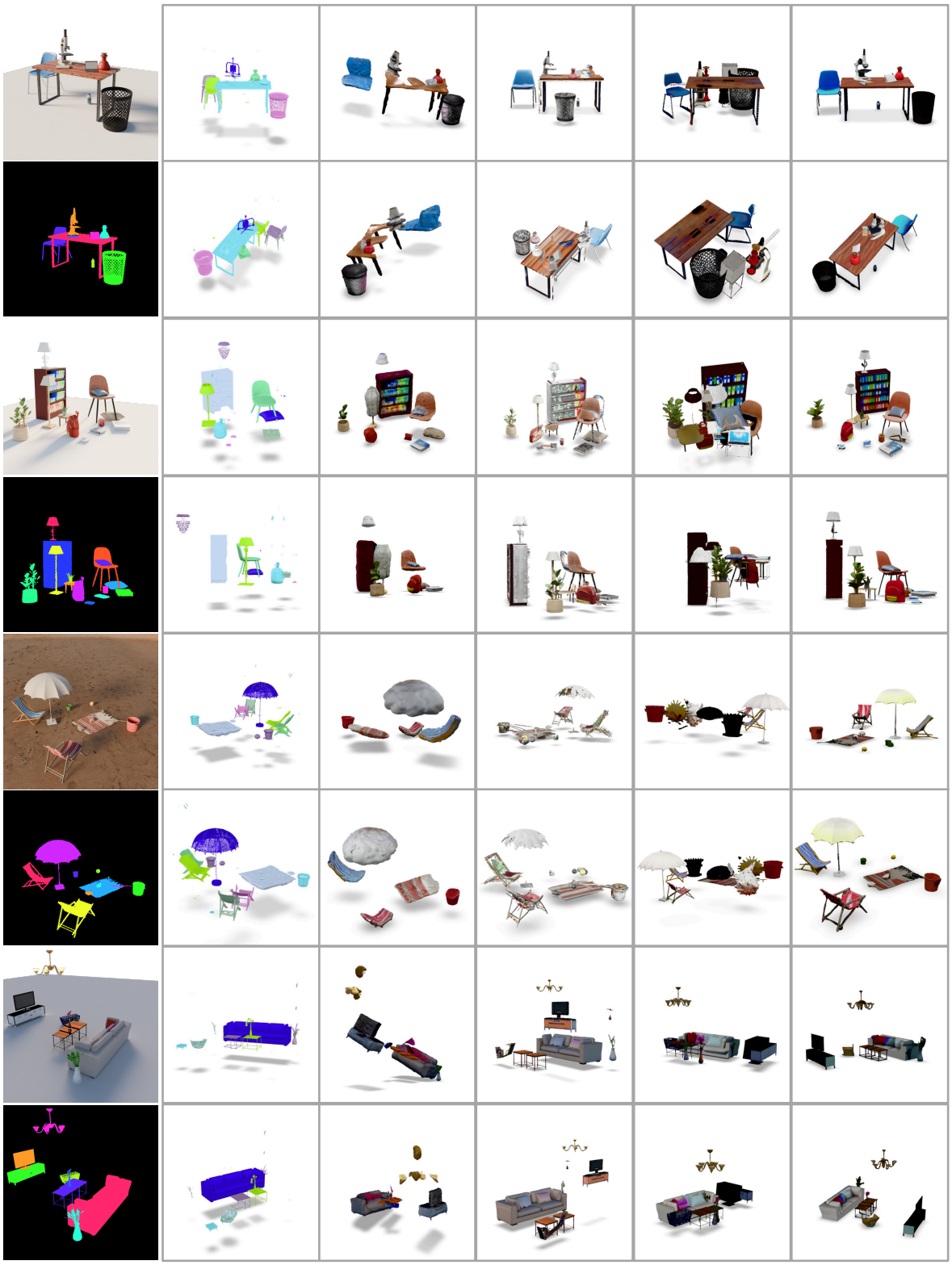

Figure 3. We present the multi-view evaluation results using synthetic scenes as input. The testing scenes contain various layouts including *front, back, right, left, small on large, behind*, and etc. Baselines includes PartCfrater [3], Gen3DSR [1], MIDI-3D [2], and SceneGen [4]

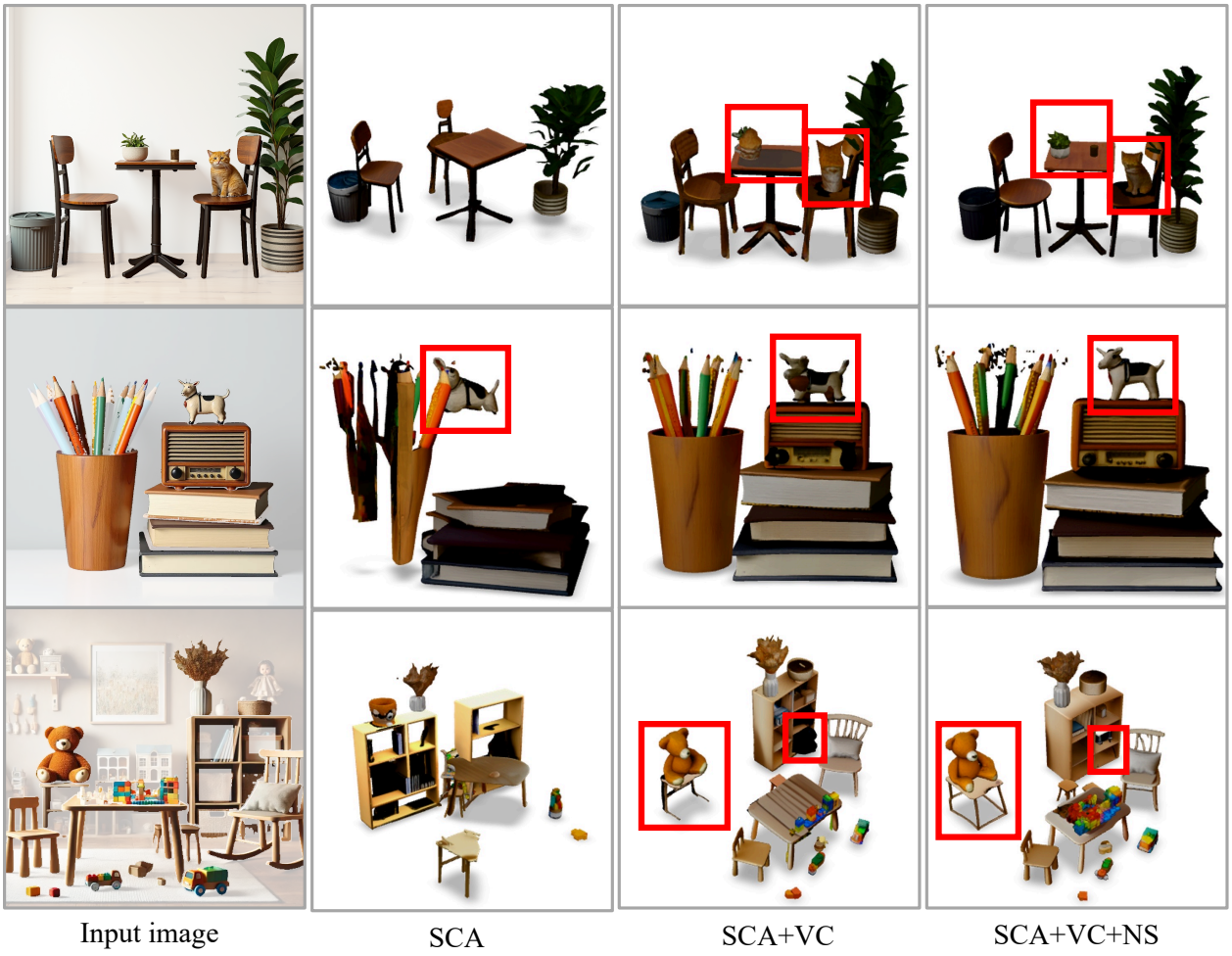

Figure 4. Ablation study on scene-context attention (SCA), view-centric space (VC), and non-semantic scenes (NS). With only SCA, the model often fails to maintain a coherent global layout and produces frequent object collisions. Adding the VC component substantially improves the overall arrangement of objects, but instance quality remains limited, as seen in the distorted cat, cow, and the chair under the toy bear. Incorporating NS further enhances both instance fidelity and global layout coherence.

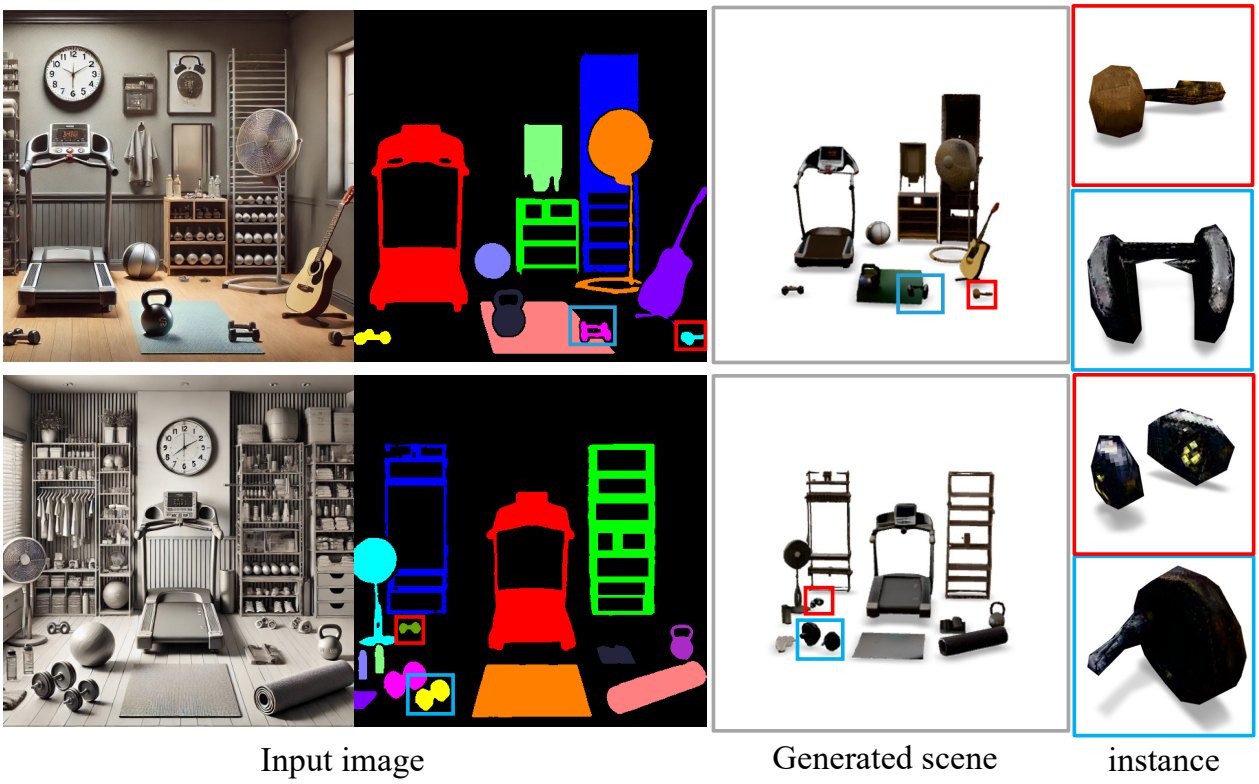

Figure 5. instance quality is bad when the input instance mask is small in the image.