JRM: Joint Reconstruction Model for Multiple Objects without Alignment

Supplementary Material

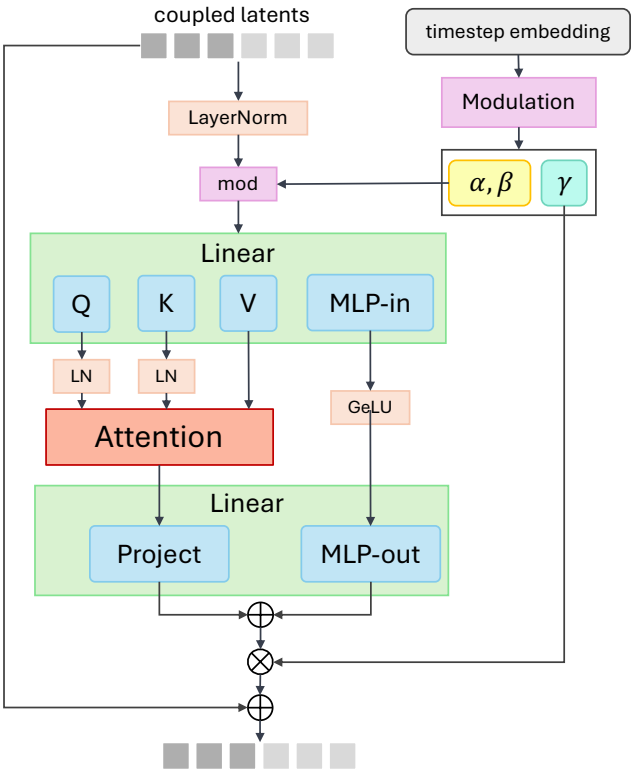

Figure 1. Illustration of a single-stream block taking coupled latents as input.

In this supplement, we provide additional details of our method (Section A) and experiments. We describe the details of synthetic data preparation for training and evaluation benchmarks (Section B). We present further ablation studies, discuss the benefits of joint reconstruction for better implicit segmentation, and qualitative results of JRM on additional real-world scenes (Section C).

A. Model Architecture

A.1. ShapeR: Robust Conditional 3D Shape Generation from Casual Captures

ShapeR [4] uses multiple egocentric input modalities and robust training strategies to achieve object-centric 3D reconstruction from image sequences. In this section, we briefly summarize how the inputs are processed.

Given a scan of an environment, an off-the-shelf visual-inertial SLAM technique [1] is used to extract a sparse 3D pointcloud and camera poses. Subsequently, object instances are identified using a 3D instance detection approach [6].

For each detected object, its sparse points, corresponding image crops, 2D point mask projected on images, and text prompt generated by a vision-language model [3] are extracted. Camera poses are embedded as Plücker ray encodings and further concatenated with image tokens.

These multimodal conditions guide a 3D rectified flow matching model, which denoises a latent VecSet using a mixed architecture of single- and double-stream blocks, and decodes it to produce the 3D meshes.

During training, ShapeR applies extensive augmentations to all modalities to simulate noisy and realistic inputs that further improve robustness. For images, the augmentations contain background compositing, occlusion overlays, visibility fog, resolution degradation, and photometric perturbations. For SLAM points, the augmentations simulate partial trajectories, a diverse range of point dropout strategies, Gaussian noise, and point occlusion.

It also leverages curriculum learning to ensure the robustness in real-world scenarios by employing two-stage training on object-level and scene-level data respectively. We refer the full details of ShapeR to a separate material, SHAPER.PDF.

A.2. Implementation Details.

Our model extends a base ShapeR architecture [4]. To make experimentation more feasible, we use a smaller variant of the ShapeR model as the basis for our experiments in this paper. This variant of ShapeR, FM, comprises 8 dual-stream and 16 single-stream blocks, each with 16 attention heads and a hidden width of 1024. It is trained for 300K steps on 128 NVIDIA H100 GPUs with an effective batch size of 1,536, progressively increasing the latent sequence length from 256 to 4,096.

The 3D VAE consists of an encoder of 8 transformer layers and a decoder of 16 layers, each with a hidden width of 768 and 12 attention heads. The VAE is trained for 200K steps with an effective batch size of 640 across 64 NVIDIA H100 GPUs.

We extend this architecture into a joint reconstruction model (JRM) by introducing coupled fusion blocks that alternate between single-stream blocks. We illustrate how the coupled attention mechanism builds upon a single-stream block in Fig. 1. Both JRM and FM use identical encoders for multimodal inputs.

The JRM is initialized from the pretrained checkpoint of FM. For the implementation of JRM, we offer two approaches: *replace* every other pretrained single-stream block with a fresh coupled fusion block, or *insert* a new

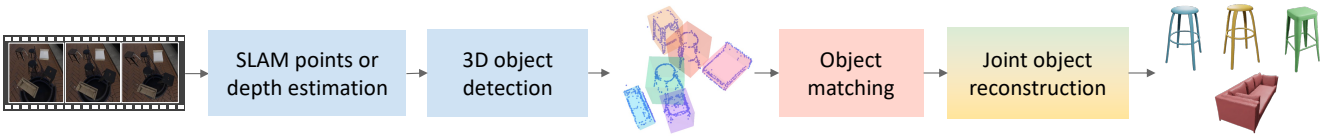

Figure 2. A simple flowchart of the complete pipeline of compositional 3D scene reconstruction from video inputs.

Table 1. Qualitative results on two JRM variants.

JRM	No Pair			Identical Pair			Similar Pair			Negative Pair		
	CD↓	NC↑	F1↑	CD↓	NC↑	F1↑	CD↓	NC↑	F1↑	CD↓	NC↑	F1↑
Replace	2.61	82.26	87.99	2.49	82.93	88.70	2.72	82.09	86.90	3.04	80.89	86.31
Insert	**2.37**	**82.96**	**89.62**	**2.18**	**83.80**	**91.15**	**2.65**	**82.07**	**88.34**	**2.56**	**82.90**	**88.69**

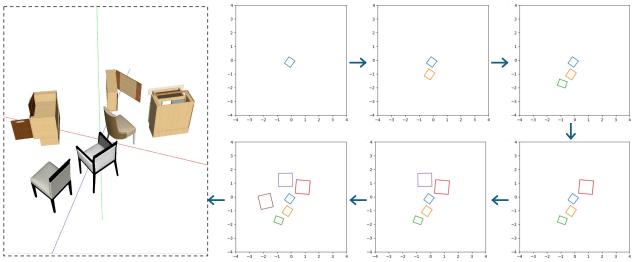

Figure 3. A top-down view illustration of how a synthetic scene is constructed with iterative object insertion by solving collision on 2D polygons (bounding boxes).

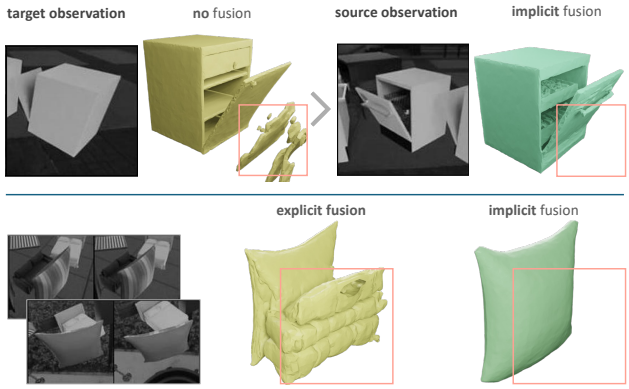

Figure 4. Implicit observation aggregation helps better segmentation during shape generation.

coupled fusion block following each single-stream block. The latter essentially enlarges the model capacity of the base ShapeR. We compare their performance differences in Sec. C.2. A complete pipeline of applying JRM to the raw video input is illustrated in Fig. 2.

JRM is trained using object pairs constructed from 80K 3D shapes. The initial learning rate is set at 1×10^{-5}, with a multi-step scheduler reducing it by 0.1 at 80K steps and by 0.05 at 160K steps. Training is conducted on 64 NVIDIA H100 GPUs with an effective batch size of 256, over 200K steps. Again for feasibility of experiments, we don't perform curriculum learning that further trains ShapeR on the scene-level data.

B. Data Preparation

B.1. Training Data

For each training object, we rescale it diagonally within a unit cube and place the center at the origin. We first sample a varied number of controlled viewpoints determined by the sampled radius of camera positions, calculated as $r_{cam} = \max(\text{bbox}_{obj}) + \delta_r$, where δ_r is a random increment within the range of 0.5 to 1. By default, the camera is oriented towards the origin. With fixed viewpoints, we then interpolate between viewpoints to produce a flying camera trajectory from a sequence of 100 camera viewpoints in total to render synthetic video capture.

With a short video clip produced, we can obtain sparse SLAM points [1] and extract video frames with strong data augmentation applied as our conditions. We use 2 randomly sampled frames of resolution 224×224 as the image condition for each object. In particular, sparse structured SLAM points better simulate real-world noisy points and generalize well to more dense depth-back-projected points.

B.2. Evaluation Benchmarks

We follow a heuristic algorithm proposed in Wu et al. [7] to construct the synthetic benchmarks. A synthetic scene is constructed by iteratively placing a new object to an existing arrangement that mimics real-world occlusion and clutter. Examples of synthetic scenes are shown in Fig. 5.

Scene layout generation. We arrange objects by sequentially adding new 3D shapes to the existing layout, ensuring that their 2D bounding boxes projected on the top-

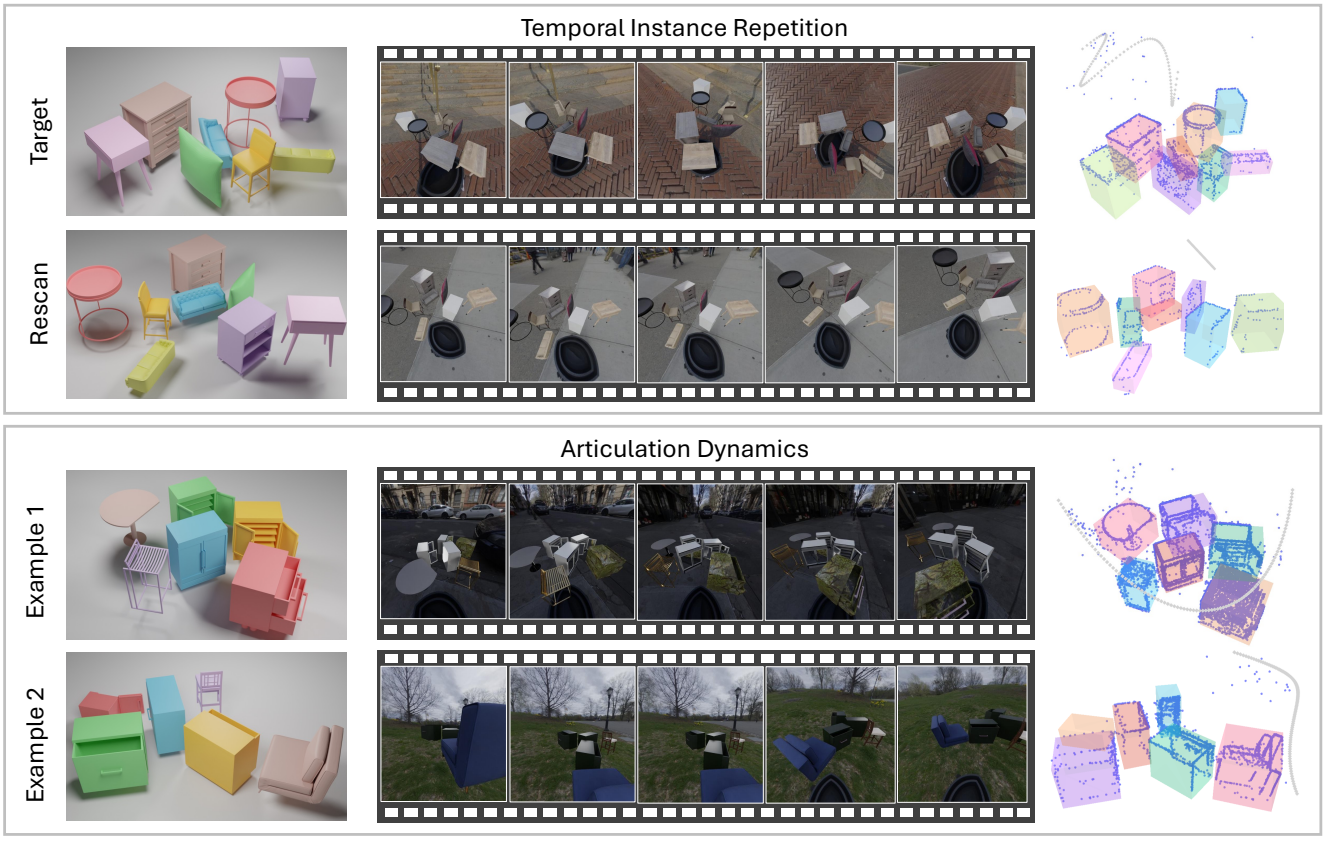

Figure 5. We show examples of generated synthetic scenes in scenarios of temporal instance repetition and articulation dynamics. In the last column, we visualize SLAM points (blue points), camera trajectory (gray diamonds) and 3D bounding boxes.

Table 2. Quantitative results of temporal instance repetition with *predicted object matching and alignment* by MORE^2. Numbers in gray color represent results of only applying the oracle object alignment to the condition inputs of FM.

Methods	Modality			No Rescan			1 Rescan			3 Rescans		
	Point	Image	Text	CD↓	NC↑	F1↑	CD↓	NC↑	F1↑	CD↓	NC↑	F1↑
MORE^2[9]	✓	✗	✗	10.43	74.45	32.25	10.08	74.12	32.94	10.24	73.40	33.36
FM	✓	✗	✗	**3.07**	**83.42**	**86.10**	3.82	78.87	80.05	4.68	72.42	71.45
							2.66	84.25	88.86	2.67	84.30	89.38
JRM	✓	✗	✗	3.46	83.26	85.20	**3.17**	**84.36**	**86.35**	**3.35**	**84.07**	**85.57**
FM	✓	✓	✓	3.12	**84.19**	**88.57**	3.54	81.67	84.08	3.85	79.34	79.85
							2.90	85.14	90.18	2.49	85.90	92.01
JRM	✓	✓	✓	**2.84**	81.75	86.74	**2.62**	**83.17**	**88.11**	**2.54**	**83.57**	**89.04**

down view do not intersect with those of previously placed shapes. The goal is to create a scene where objects are near enough to present occlusion in rendering but do not overlap (see Fig. 3).

For each 3D shape S_i to be placed, we normalize its scale within a unit cube and randomly rotate it around the vertical axis. If no shapes have yet been placed, its starting position $\boldsymbol{p}_i^0$ is set at the origin; otherwise, it is positioned at the mean location of the shapes placed previously. A unit vector $\boldsymbol{v}_i$ is randomly sampled for the direction of placement. The initial placement distance d_i^0 is calculated by summing the short sides of all positioned objects. Thus, the location of the new shape is determined by $d_i^0 \cdot \boldsymbol{v}_i + \boldsymbol{p}_i^0$.

In cases where the 2D bounding boxes of existing objects

Figure 6. Reconstruction with similar source objects.

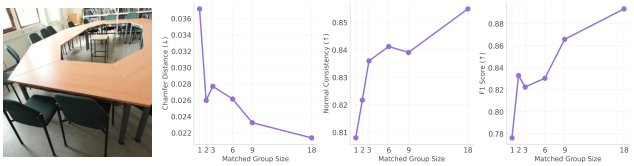

Figure 7. Reconstruction metrics for 18 chairs from ScanNet++ *Scan 4* as the size of the group for joint reconstruction increases.

Figure 8. Reconstruction with all objects in a single group.

intersect with a new object, we increment d_i^0 by 0.05 iteratively until, after N iterations, no intersections remain. The final shape position is calculated as $(d_i^0 + 0.05 \times N) \cdot \boldsymbol{v}_i + \boldsymbol{p}_i^0$. The vertical position of the shape is also calculated with a random deviation sampled from $[-0.5, 0.2]$.

We repeat the procedure until all shapes are placed in the scene. Note that we only consider intersections among 2D bounding boxes from the top-down view as a simplification instead of using more expensive physics-based collision checks.

Temporal instance repetition. To investigate the setting of temporal repetition of instances, we adopt the configuration setting of synthetic scenes from LivingScenes [9] where a living scene is considered as a changing 3D environment of multiple objects dynamically transformed over time.

We randomly sample a set of 3D shapes to build a scene from 6 categories, including "chair", "table", "sofa", "bed", "pillow" and "lamp". Each scene contains a varied number of objects, ranging from 4 to 8. For the same set of selected objects, we generate 4 distinct scene arrangements using the scene layout generation approach described above. One of the arrangements is considered the target reconstruction scene, while the other three arrangements simulate temporal changes of the target scene with rigid transformations at irregular intervals.

Spatial instance repetition. In the setting of spatial instance repetition, we only generate one possible scene arrangement for the selected objects.

We randomly sample a target object from 6 main categories, including "chair", "table", "sofa", "bed", "storage" and "others". The broad "others" category further contains 10 sub-categories with fewer instances each, including "flower pot", "vase", "fan", "lamp", "tent", "ladder", "pillow", "cart", "office appliance", and "exercise weight". We intentionally choose three distinct source objects for the target: one identical, one similar, and one negative.

In a manner akin to generating pair-wise training data, a similar instance is randomly selected using pre-computed DuoDuoCLIP [2] embeddings, ensuring the cosine similarity with the target object is greater than 0.65 and belongs to the same semantic category. A negative instance is randomly selected from different semantic categories. Two more objects are randomly sampled from all available categories as occluders to create occlusion in renderings.

Articulation. To study the generalization of JRM to dynamic articulated objects, we generated synthetic scenes with repetitive instances of an articulated object spawned into a scene at different motion states. We primarily concentrate on articulated furniture items intended for storage or functionality, like cabinets, dressers, dishwashers, microwaves, *etc*.

Procedural programs can swiftly create these objects by altering kinematic graphs at the part level and adjusting spatial parameters like positions and sizes. Initially, we select an articulated object and create three variations, each exhibiting distinct part-level deformations by employing the rest state and two random articulation states. As with spatial instance repetition, we introduce two random objects into the scene to act as occluders.

Synthetic scene rendering. We utilize a camera trajectory generation method similar to that applied for object-centric training data. The key difference is that the camera moves around in the entire scene rather than focusing on a single object. Moreover, the camera is directed towards a changing lookat target rather than the fixed origin. We use back-projected rendered depth points as the point condition to JRM for evaluation on all synthetic scene benchmarks.

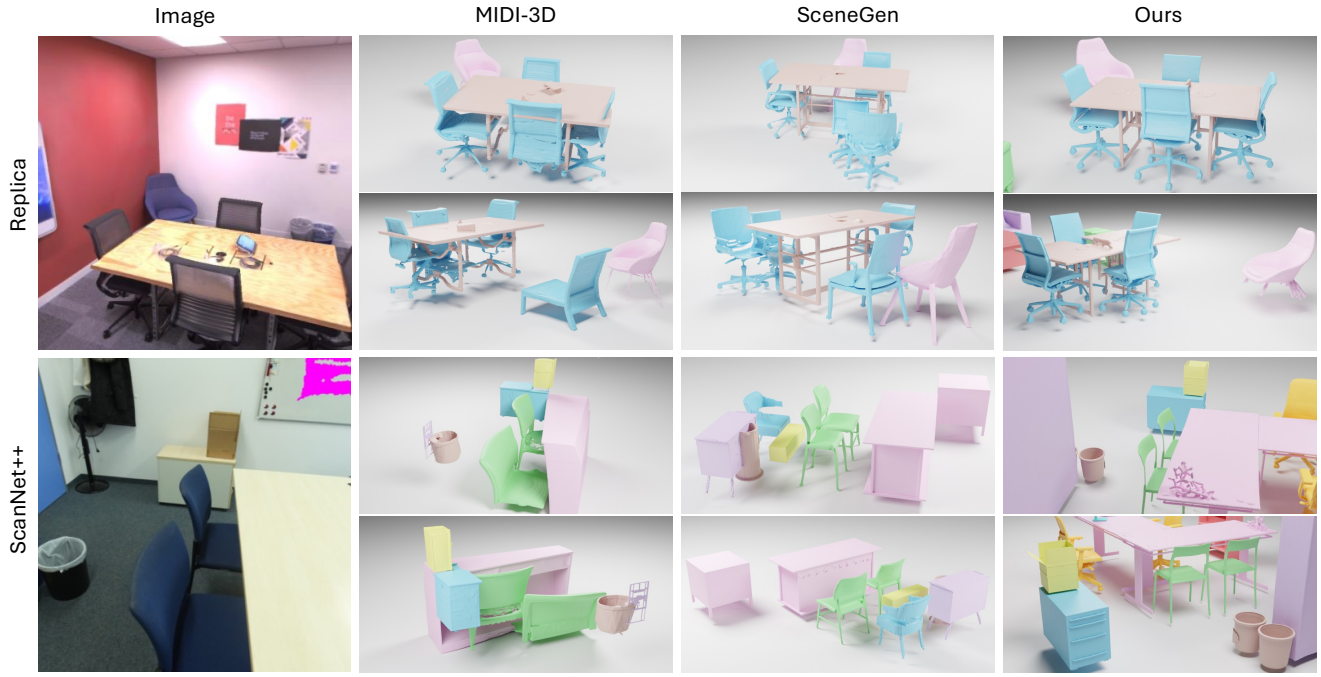

Figure 9. Comparison with MIDI-3D and SceneGen, both are conditioned on the single image displayed. Best viewed zoomed.

C. Additional Experiments

C.1. Additional Results on Temporal Instance Repetition

In Tab. 2, we present additional quantitative results on temporal instance repetition with predicted object matching by MORE^2 [9]. This differs from Table 2 in the main paper, which used ground truth object matching instead of a predicted matching. Therefore, the results for "No Rescan" remain the same as before, but change with further rescans where ideal object matching is no longer assumed.

For these experiments, we evaluate two variants of FM and JRM. We train a version of FM and JRM conditioning solely on the point condition, and build on a curated training dataset that matches the size of the MORE^2 training data, offering a fair comparison with MORE^2 [9]. We also present results of FM and JRM trained on our full data and conditioned on three modalities.

In these results, we see a continuation of the trends observed in the main paper. Even with both imperfect matching and alignment, JRM is able to improve reconstruction accuracy with addition source scans. However, the additional noise introduced by imperfect matching causes the baseline methods' reconstructions to deteriorate as the environment is re-observed.

C.2. Coupled Blocks Insertion Ablation

We study how JRM performs with different implementations of the framework. In addition to *replacing* every other single-stream block with a coupled fusion block in the original FM architecture, we can instead *insert* a coupled fusion block after each single-stream block, which results in a larger model capacity, 40 attention layers in total.

The results in Tab. 1 show that both implementations fulfill the goal of observation fusion, but that additional model capacity yields further performance improvements across different paired objects.

C.3. Additional Qualitative Results

Robustness to dissimilarity. We clarify that JRM does not "force" dissimilar objects to look alike, nor do we filter for near-identical training pairs. During training, the model is exposed to pairs of distinct objects with a probability of 0.1, ablated in Paper Tab. 5. In Fig. 6, we show JRM respects instance-specific identity; it correctly preserves unique features like throw cushions or specific leg geometries on one while omitting them from others. Paper Tab. 2 quantifies this robustness where "similar" matches with CLIP scores between 0.65 and 0.9. The degradation from non-identical matches is significantly less severe than the baseline, showing JRM balances shared features between objects rather than blindly collapsing them.

Implicit Segmentation. A common failure mode that we observe in our experiments is the generation of additional

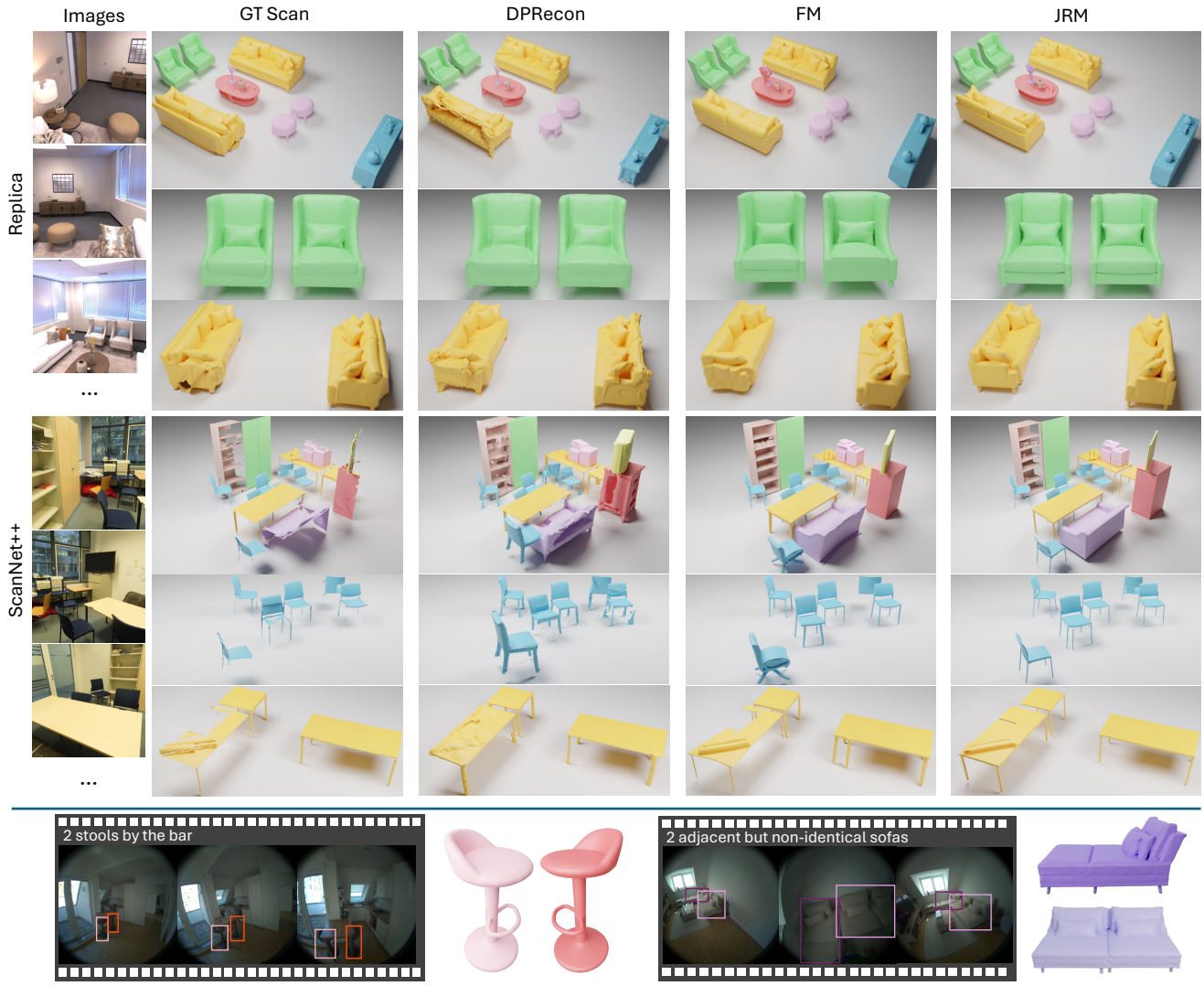

Figure 10. Additional qualitative results on real-world scenes, including ScanNet++ and Replica. We use the same color for objects within the same match group. For each scene, we produce two close-up views focusing on the matched objects.

geometry that does not belong to the target object. We refer to these cases as failures of implicit segmentation, as the reconstruction method is unable to isolate the target from distractor content, *e.g.* from occluding objects, in the conditioning inputs.

We show in Fig. 4 that joint reconstruction with implicit observation fusion also improves object segmentation during geometry generation. In the first case, taking extra unaligned source observations helps to eliminate artifacts due to a restricted perspective of the target observation. In the second case, simply combining observations using explicit alignment can result in worse reconstruction. However, joint reconstruction with implicit aggregation is capable of avoiding such performance degradation.

Generalisation to larger group sizes. We investigate the generalisation of JRM to larger batches of joint reconstruction, from pairs, as seen during training to larger groups. Specifically, we use *Scan 4* from ScanNet++, which has 18 similar chairs. Figure 7 shows chair reconstruction accuracy. Although trained on pairs, metrics consistently improve as the size of the joint group increases.

Comparison with MIDI-3D and SceneGen. We perform qualitative comparisons on Replica and ScanNet++ in Fig. 9. While these prior works are conditioned on a single image, JRM leverages multi-view inputs; ground-truth segmentation is provided to all methods. Some baseline errors, such as pose and scale inaccuracies, may stem from the monocular input. However, more fundamental issues exhibited by both MIDI-3D and SceneGen, *e.g.* implausible geometry and inconsistent styles, are largely avoided by JRM.

We attribute this to the significantly larger scale of data accessible by our object-centric training approach compared to scene-limited alternatives.

Full-scene group. Using the same scenes as Fig. 9, we evaluate a baseline with all objects in a single group in Fig. 8. There is a regression compared to class-based groups, but the reconstructions remain representative.

Real-world Scenes. We present extra qualitative results on real-world scenes in Fig. 10, including sources of Replica [5], ScanNet++ [8] and a real apartment scanned using Aria glasses [1].

References

[1] Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, et al. Project aria: A new tool for egocentric multi-modal ai research. *arXiv preprint arXiv:2308.13561*, 2023. 1, 2, 7

[2] Han-Hung Lee, Yiming Zhang, and Angel X Chang. Duoduo CLIP: Efficient 3D understanding with multi-view images. In *International Conference on Learning Representations (ICLR)*, 2025. 4

[3] Meta AI. Introducing LLaMA 4: Advancing Multimodal Intelligence, 2025. 1

[4] Yawar Siddiqui, Duncan Frost, Samir Aroudj, Armen Avetisyan, Henry Howard-Jenkins, Daniel DeTone, Pierre Moulon, Qirui Wu, Zhengqin Li, Julian Straub, et al. ShapeR: Robust conditional 3D shape generation from casual captures. *arXiv preprint arXiv:2601.11514*, 2026. 1

[5] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 7

[6] Julian Straub, Daniel DeTone, Tianwei Shen, Nan Yang, Chris Sweeney, and Richard Newcombe. EFM3D: A benchmark for measuring progress towards 3D egocentric foundation models. *arXiv preprint arXiv:2406.10224*, 2024. 1

[7] Qirui Wu, Daniel Ritchie, Manolis Savva, and Angel X Chang. Generalizing single-view 3D shape retrieval to occlusions and unseen objects. In *International Conference on 3D Vision (3DV)*, pages 893–902. IEEE, 2024. 2

[8] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023. 7

[9] Liyuan Zhu, Shengyu Huang, and Iro Armeni Konrad Schindler. Living scenes: Multi-object relocalization and reconstruction in changing 3D environments. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3, 4, 5