A Appendix

A.1 Societal Impact

OSRT enables efficient 3D scene decomposition and novel view synthesis without the requirement of collecting object or segmentation labels. While we believe that OSRT will accelerate future research efforts in this area, our current investigation is still limited to synthetically generated 3D scenes, populated with relatively simple 3D scanned objects from everyday environments. As such, it has no immediate impact on general society.

In the longer term, we expect this class of methods to be more broadly applicable and that methods in this area—compared to supervised approaches for scene or object understanding—will provide several advantages in terms of reliability and interpretability. Indeed, individual learned object variables can be inspected by analyzing or even rendering the respective *slot*. In terms of annotator bias, no human labels are required to learn object representations, and thus annotation bias cannot leak into the learned model (while dataset selection bias and related biases, however, still can). Better understanding these alternative forms of bias—in the absence of human labels—and their impact on model behavior will prove important for mitigating potential negative societal impacts that could otherwise arise from this line of work.

A.2 Baselines

A.2.1 ObSuRF [3]

ObSuRF checkpoints for MSN-Easy and CLEVR-3D were provided by the authors. For MSN-Hard, we train ObSuRF using Adam with hyperparameters similar to those used for MSN-Easy. The following hyperparameters were modified to accommodate MSN-Hard: the number of slots was increased from 5 to 32 to account for up to 31 objects per scene, the positional encoding frequency range varies from 2^{-7} to 2^{11} to account for the larger scene size, and the number of scenes per batch was reduced from 64 to 32 to account for the larger peak memory usage. We also project 3D query points onto the reference camera to gather image-derived feature vectors generated by ResNet-18. Lastly, we delay loss occlusion annealing until step 180,000, linearly increasing the loss weight from 0 to 1 over 80,000 steps. This is because we found that increasing the occlusion loss weight earlier causes the model to converge towards a local minimum where the entire scene is explained by a single slot. We terminate training after 800 k steps as we observed no further change to the loss or FG-ARI. We train using $4\times$ A100 GPUs for 6 days.

A.2.2 uORF [5]

We trained uORF on MSN-Easy and MSN-Hard. For MSN-Easy, we used 5 slots with 64-dimensional embeddings per slot. The model managed to do a good job in getting the position of the objects, but failed to precisely capture their shapes, achieving an FG-ARI of only 0.216. Note that we kept the input size to the default of 128×128, which results in metrics giving us a sense of quality, while not being directly comparable with the ones measured on ObSuRF and OSRT. After many attempts to train the model on MSN-Hard and feedback from the original authors, we did not manage to obtain a model capturing the objects, but only the background and horizon. We believe this to be a result of uORF's already limited capacity, which we had to further constrain by reducing the slot dimensionality to 10 to fit a 32-slot model into an Nvidia A100 GPU with 40 GB of VRAM. We also tried further alternatives, such as allowing the model only 8 slots, but instead of larger dimensionality, without success.

A.3 Model and training details

Unless stated otherwise, we use the exact same model with identical hyper parameters for all experiments across all datasets. The full OSRT model has 81 M params.

SRT Encoder. We mainly follow the original SRT Encoder, parameters are therefore identical to the description provided by Sajjadi et al. [2]. We list our changes to the SRT Encoder below.

Figure 1: SRT as published by Sajjadi et al. [2] (left image in each pair) and SRT with our tweaks as described in Appendix A.3 (right image in each pair).

Poses. We use a simple absolute parametrization of space rather than relative poses, and we drop camera ID and 2D position embeddings, as we found this to slightly improve reconstruction quality without further drawbacks. The only exception is UpOSRT, as this models relative poses by definition.

CNN. We remove the last block of the CNN in the encoder, thereby speeding it up and reducing its number of parameters considerably, while increasing the number of tokens in the SLSR by a factor of 4. While this would slow down SRT's inference a bit due to the larger SLSR, Slot Mixer operates on the constant-size SlotSR, so its rendering speed remains unaffected by this change.

Encoder Transformer. Instead of 10 post-normalization transformer layers with cross-attention, we use 5 pre-normalization layers [4] with self-attention.

Altogether, our tweaks reduce SRT's model size from 74 M to only 41 M parameters. Nevertheless, reconstruction quality is vastly improved, pushing PSNR from 23.33 to 25.92 on MSN-Hard. Qualitative results can be inspected in Fig. 1.

Slot Attention. The architecture of the Slot Attention module follows Locatello et al. [1]. Slots and embeddings in attention layers have 1536 dimensions. The MLP doubles the feature size in the hidden layer to 3072. We use a single Slot Attention iteration on all experiments except MSN-Easy, where we found 3 iterations to perform best.

Slot Mixer. The *Allocation Transformer* is based on SRT's Transformer Decoder [2]. However, before the positional encoding of the query rays (with 180 dimensions) is passed to the transformer, we apply a small MLP with 1 hidden layer of 360 dimensions with a ReLU activation. We found this to improve results and stabilize training. The attention layer in *Mixing Block* uses 1536 dimensions for the embeddings and a single attention head to provide scalar w_i. The *Render MLP* is a standard MLP with ReLU activations and 4 layers with 1536 hidden dimensions.

Training. We follow the exact training procedure described by Sajjadi et al. [2] for SRT, with the difference that we do not train for 4 M, but only 3 M steps. We trained OSRT for 16 h on CLEVR-3D and MSN-Easy and OSRT (1) and (5) for roughly 4 and 7 days respectively on MSN-Hard on 64 TPUv2 chips, always with batch size 256. Similar to Sajjadi et al. [2], we observed that OSRT's performance on MSN-Hard slowly improved beyond our training time (both in terms of PSNR and FG-ARI), though conclusions and comparisons are not affected by prolonged training.

A.4 Failure Cases

Fig. 2 shows an example result from prior experiments where a variation of OSRT would not segment the scene into objects, but into spatial locations. Notably, the decomposition is still 3D-consistent, but it is largely independent of object positions.

A.5 Results

We provide a convenient overview of all results for all datasets in Tabs. 1 to 3.

A.6 Videos

We provide video renders of OSRT in the supplementary material. For convenience, we have included an html file for easier viewing.

Table 1: **Complete quantitative results on CLEVR-3D**.

CLEVR-3D	PSNR	FG-ARI	FG-ARI ratio
ObSuRF	33.69	0.978	0.999
OSRT	39.98	0.976	0.995

Table 2: **Complete quantitative results on MSN-Easy**.

MSN-Easy	PSNR	FG-ARI	FG-ARI ratio
ObSuRF	27.96	0.792	0.993
ObSuRF w/o $\mathcal{L}_O$	27.41	0.940	0.997
OSRT	29.74	0.954	0.996

Table 3: **Complete quantitative results on MSN-Hard**.

MSN-Hard	PSNR	FG-ARI	FG-ARI ratio
ObSuRF	16.50	0.280	0.707
OSRT (1)	20.52	0.619	0.940
OSRT (3)	22.75	0.794	0.987
OSRT (5)	23.54	0.812	0.987
OSRT SB Decoder	23.35	0.801	0.987
OSRT SRT Decoder	24.40	0.330	0.880
VOSRT	21.38	0.767	0.984
UpOSRT	22.42	0.798	0.987
2D Reconstruction	28.14	0.198	0.671
SRT [2]	23.33	N/A	N/A
SRT (ours)	25.93	N/A	N/A

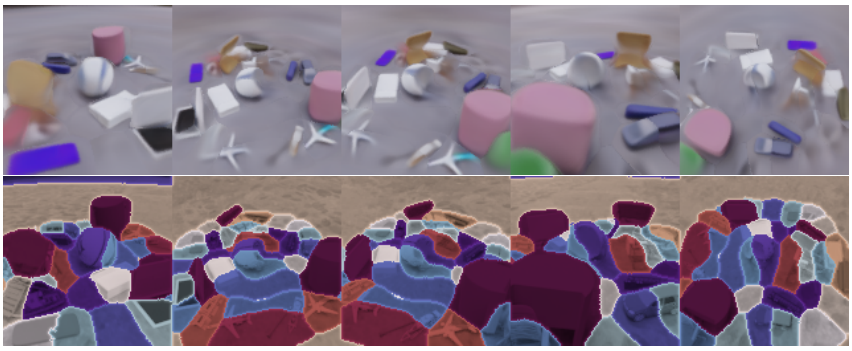

Figure 2: **Failure case** – Example of bad scene decompositions from an earlier OSRT experiment, showing prediction and slot assignments. While 3D-consistent, the decomposition cuts through objects and appears like a spatial partitioning of the scene.

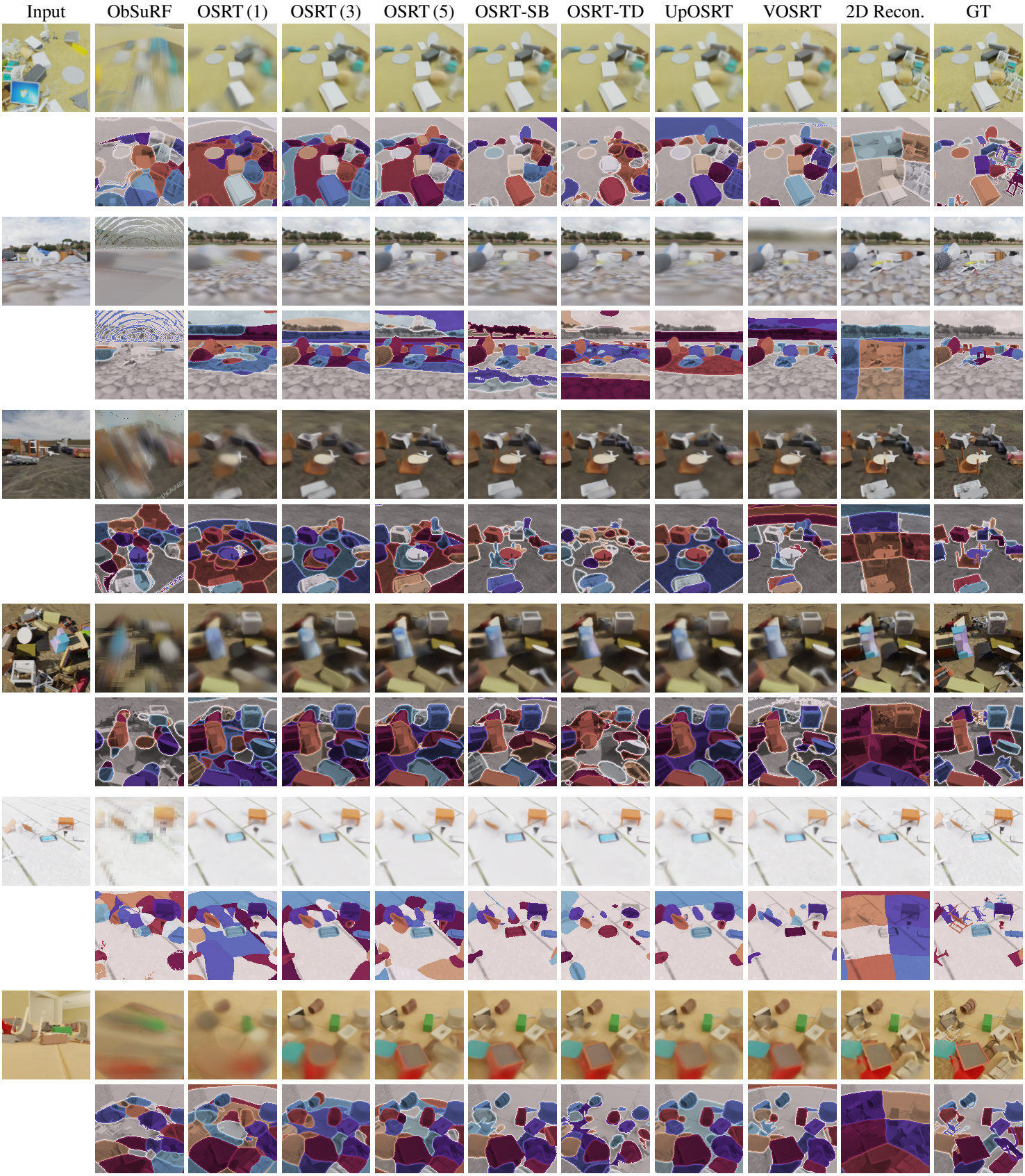

Figure 3: **Qualitative results on MSN-Hard** – Left to right: ObSuRF [3], OSRT with 1, 3, and 5 input views, SB Decoder, Transformer Decoder, unposed and volumetric OSRT, and finally OSRT trained on target view reconstruction (2D).

References

[1] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *NeurIPS*, 2020.

[2] Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lucic, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas Funkhouser, and Andrea Tagliasacchi. Scene Representation Transformer: Geometry-Free Novel View Synthesis Through Set-Latent Scene Representations. In *CVPR*, 2022.

[3] Karl Stelzner, Kristian Kersting, and Adam R Kosiorek. Decomposing 3d scenes into objects via unsupervised volume segmentation. *arXiv preprint arXiv:2104.01148*, 2021.

[4] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *ICML*, 2020.

[5] Hong-Xing Yu, Leonidas J Guibas, and Jiajun Wu. Unsupervised discovery of object radiance fields. In *ICLR*, 2022.