Self-Supervised Object-Wise 3D Decomposition of Images using Shape Priors

Cathrin Elich[1,2,3], **Martin R. Oswald**[3], **Marc Pollefeys**[3,4], **Jörg Stückler**[1]
[1]Max Planck Institute for Intelligent Systems, Tuebingen, Germany
[2]Max Planck ETH Center for Learning Systems
[3]Department of Computer Science, ETH Zurich
[4]Microsoft Mixed Reality and AI Zurich Lab
{cathrin.elich, joerg.stueckler}@tuebingen.mpg.de
{martin.oswald, marc.pollefeys}@inf.ethz.ch

Abstract

Object-wise scene representations are essential for scene understanding and decision making by intelligent agents. In this paper, we present our approach for learning multi-object scene representations from images which model the 3D scene layout and the shape and texture of objects. Our deep recurrent architecture encodes single images into an object-wise latent representation with 3D shapes, poses and texture of the objects. We confine the space of possible shapes using a pre-trained shape prior represented continuously in function-space as signed distance functions. The scene representation is rendered back into images in a differentiable way to facilitate self-supervised learning of scene decomposition. We evaluate our approach on scenes generated with ShapeNet models.

1 Introduction

Object-wise scene decomposition and inference of object properties such as 3D shape and texture from single views is a remarkable capability of human perception. In intelligent systems similar capabilities could facilitate object-level description, abstract reasoning and high-level decision making. Previous work on learning-based scene representations focused on single-object scenes [1], neglected to model the 3D geometry of the scene and the objects explicitly [2, 3, 4, 5, 6, 7, 8], or did not regress scene representations from input views [9]. In our work, we propose a multi-object scene representation network which learns to decompose scenes into objects and represents the 3D shape and texture of the objects explicitly. Shape, pose and texture are embedded in a latent representation which our model decodes into textured 3D geometry using differentiable rendering. This allows for training our scene representation network in a semi-supervised way. Our approach jointly learns the tasks of object instance detection, instance segmentation, object pose estimation and inference of 3D shape and texture in single RGB images. Inspired by [10, 11, 1], we represent 3D object shape and texture continuously in function-space as signed distance and color values at continuous 3D locations. The scene representation network infers the object poses and its shape and texture encodings from the input RGB image. Our differentiable renderer efficiently generates color and depth images as well as instance masks from the object-wise scene representation. For an overview on current work on differentiable renderer, we refer the reader to [12, 13]. By this, our model facilitates to generate new scenes by altering an interpretable latent representation (see Fig. 1). Our network is trained in two stages: In a first stage, we train an auto-decoder subnetwork of our full pipeline to embed a collection of meshes in continuous SDF shape embeddings as in DeepSDF [10]. With this pre-trained shape space, we train the remaining parts of our full multi-object network to decompose and describe the scene by multiple objects in a self-supervised way from RGB-D images. No ground truth of object pose, shape, texture, or instance segmentation is required for the training on multi-object scenes.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

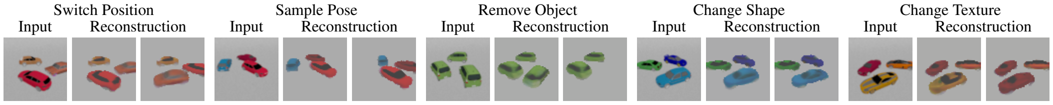

Figure 1: **Example scenes with object manipulations using our learned object-wise scene representation.** For each example, we input the left images and compute the middle one as standard reconstruction. After the manipulation in the latent space, we obtain the respective right image.

We denote our learning approach semi-supervised due to the supervised pre-training of the shape embedding and the self-supervised learning of the scene decomposition. We evaluate our approach on synthetic scene datasets with images composed of multiple objects.

2 Method

We propose an autoencoder architecture which learns object-wise scene representations from single images. Each object is explicitly described by its 3D pose, shape and textural appearance. By decoding the images back from the object-wise latent representation through differentiable rendering, our model can be trained in a self-supervised way from RGB-D images.

Scene Encoding. We use a recurrent encoder architecture to simultaneously decompose the scene and infer the objects' latent representation $\mathbf{z}_i = (\mathbf{z}_{i,sh}^\top, \mathbf{z}_{i,tex}^\top, \mathbf{z}_{i,ext}^\top)^\top \in \mathbb{R}^d, i \in \{1, \ldots, N\}$. A shared encoder regresses object latents sequentially from the original input image I as well as the difference image from the reconstructed object composition $\Delta\widehat{I}_{1:i-1}$ and combined mask $\widehat{M}_{1:i-1}$ from previously found objects. These decoded object composition images are required to guide the encoder to regress the latent representation of a yet undetected object. The object encoding $\mathbf{z}_i$ explicitly disentangles information about the object's 3D shape, textural appearance and extrinsics. Each shape and texture decoding $\mathbf{z}_{i,sh}, \mathbf{z}_{i,tex}$ similarly parametrizes respective autodecoders based on [10] which evaluate signed distance functions (SDF) and color values on continuous 3D points. Object position $\mathbf{p}_i = (x_i, y_i, z_i)^\top$, orientation θ_i and scale s_i are regressed with the extrinsic encoding $\mathbf{z}_{i,ext}$. We assume the objects to be placed upright, model rotations around the vertical axis, as well as a limited scaling range. Scenes are assumed to consist of a constant and known number of objects. An additional background encoder regresses the uniform color of the background plane. Plane extrinsics and camera parameters are assumed to be known.

Scene Decoding. Given our object-wise scene representation, we use differentiable rendering to generate individual images of objects based on their geometry and appearance and compose them into scene images. An object-wise renderer determines color image I_i, depth image D_i and occlusion mask M_i from each object encoding independently. Inspired by [14], we trace the SDF zero-crossing by evaluating the SDF function network $\mathbf{\Phi}$ on sampled points on the rays through each pixel $\mathbf{u} \in \mathbb{R}^2$. For this, the sampled points are transformed from camera to object coordinate system with respect to the known camera intrinsics parameters and view pose onto the scene as well as the estimated object pose on the plane. The SDF network is parametrized by the inferred shape latent of the object. The algorithm finds the zero-crossing along a ray at the first pair of samples with a sign change of the SDF. In this case, the depth $D_i(u)$ is determined through linear interpolation of the corresponding points' depth regarding their SDF values and the binary occlusion mask $M_i(u)$ is set to 1 for the respective pixel u. Otherwise, the depth is set to a large constant and occlusion is set to 0. For an occupied pixel $\mathbf{u}$, the color decoder network $\mathbf{\Psi}$ is evaluated at the corresponding estimated 3D point in object coordinates regarding the inferred texture latent $\mathbf{z}_{i,tex}$. The scene images, depth images and occlusion masks are composed from the individual objects and the background through z-buffering. We initialize them with the decoded background color, depth image of the empty plane and empty mask. For each pixel $\mathbf{u}$, we set the pixel's values in $\widehat{I}_{1:N}, \widehat{D}_{1:N}, \widehat{M}_{1:N}$ according to the occluding object i with the smallest depth at the pixel.

Training. We train our network architecture in two stages. In a first stage, we learn the SDF function network from a collection of meshes similar to [10]. The second stage uses the pre-trained SDF models to learn the remaining components for the object-wise scene decomposition and rendering network. Our multi-object network architecture is trained self-supervised from RGB-D images containing example scenes composed of multiple objects. To this end, we minimize the loss function

$L_{total} = \lambda_I L_I + \lambda_D L_D + \lambda_{gr} L_{gr} + \lambda_{sh} L_{sh}$, which is a weighted sum of the following sub-losses:

$$L_I = \frac{1}{|\Omega|} \sum_{\mathbf{u} \in \Omega} \left\| G\left(\widehat{I}_{1:N}\right)(\mathbf{u}) - G(I_{gt})(\mathbf{u}) \right\|^2 \qquad L_D = \frac{1}{|\Omega|} \sum_{\mathbf{u} \in \Omega} \left\| G\left(\widehat{D}_{1:N}\right)(\mathbf{u}) - G(D_{gt})(\mathbf{u}) \right\|$$

$$L_{gr} = \sum_i \max(0, -z_i) + \max(0, -\phi_i(z_i')) \qquad L_{sh} = \sum_i \|\mathbf{z}_{i,sh}\|^2$$

In particular, deviations of the reconstruction from both ground-truth color image I_{gt} and depth D_{gt} are penalized. We denote the set of image pixels by Ω. We apply Gaussian smoothing $G(\cdot)$ with decreasing standard deviation over time which improves object localization. L_{sh} regularizes the shape encoding to stay within the training regime of the SDF network. Lastly, L_{gr} favors objects to reside above the ground plane with z_i being the coordinate of the object in the world frame and $\phi_i(\mathbf{x}_k)$ the SDF value at corresponding projection onto the ground plane. We use a CNN with subsequent fully connected layers for both the object and the background encoder. Similar to [10], we use multi-layer fully-connected neural networks for the shape decoder $\mathbf{\Phi}$ and texture decoder $\mathbf{\Psi}$.

3 Experiments

We evaluate our approach on synthetic scenes based on the Clevr dataset [15] with objects from ShapeNet [16]. These scenes contain images with three objects on a planar single-colored background. We separately consider objects from the categories cars and armchairs. Specifically, we select 25 models per setting which we use both for pre-training the DeepSDF as well as for the generation of the multi-object datasets. Each dataset consists of 25K (train:18K, val:2K, test:5K) images with size 64×64 pixels. Objects are randomly rotated and placed in a range of $[-1.5, 1.5]^2$ on the ground plane. While we ensure that any two objects do not intersect, they might be completely hidden behind another one. Additionally to the RGB images, we also generate depth maps for training as well as instance masks for evaluation. The evaluation is performed on two different test sets: (1) with known shapes and (2) with new objects.

Evaluations Metrics. We evaluate the task of learning object-level 3D scene representations using measures for instance segmentation, image reconstruction, and pose estimation. To evaluate the image decomposition, we report average precision (AP$_{0.5}$), average recall (AR$_{0.5}$), F1$_{0.5}$-score for predicted and ground truth masks with intersection-over-union (IoU) of at least $\tau = 0.5$ as well as the mean AP over thresholds in range $[0.5, 0.95]$ with stepsize 0.05 [17]. Only objects that occupy at least 25 pixels are taken into account. Furthermore, we list the ratio of scenes where all visible objects were found w.r.t. $\tau = 0.5$ (allObj). Next, we evaluate the quality of both the RGB and depth reconstruction obtained from the generated objects. To assess the image reconstruction, we report *Root Mean Squared Error* (RMSE), *Structural SIMilarity Index* (SSIM) and *Peak Signal-to-Noise Ratio* (PSNR) scores. For the object geometry, we compute similar to [18] the *Absolute Relative Difference* (AbsRD), *Squared Relative Difference* (SqRD), as well as the RMSE for the predicted depth. Furthermore, we report the error on the estimated objects' position (mean) and rotation (median, sym.: up to symmetries) for objects with a valid match w.r.t. $\tau = 0.5$. We show results over five runs per configuration and report the mean and best result.

ShapeNet Dataset Evaluation. In Fig. 2 (left), we show reconstructed images and normal maps on our ShapeNet dataset [16]. The network correctly decomposes scenes into the constituent objects and

Table 1: Results for scenes containing objects from different categories. We differentiate between scenes that consist of shapes that were seen during training and novel objects.

			Instance Reconstruction					Image Reconstruction			Depth Reconstruction			Pose Estimation	
			mAP↑	AP$_{0.5}$ ↑	AR$_{0.5}$ ↑	F1$_{0.5}$ ↑	allObj ↑	RMSE ↓	PSNR ↑	SSIM ↑	RMSE ↓	AbsRD ↓	SqRD ↓	Err$_{pos}$ ↓	Err$_{rot}$ [sym.] ↓
cars	*seen*	best	0.750	0.991	0.991	0.991	0.979	0.064	24.092	0.898	0.158	0.006	0.004	0.144	23.67° [3.29°]
		mean	0.738	0.990	0.990	0.990	0.975	0.064	23.979	0.894	0.160	0.006	0.005	0.146	22.09° [3.07°]
	unseen	best	0.639	0.980	0.980	0.980	0.955	0.077	22.442	0.843	0.210	0.010	0.008	0.183	24.24° [4.53°]
		mean	0.632	0.977	0.977	0.977	0.944	0.077	22.454	0.842	0.208	0.010	0.008	0.184	24.25° [4.41°]
chairs	*seen*	best	0.432	0.897	0.871	0.881	0.640	0.086	21.576	0.803	0.829	0.040	0.117	0.308	43.64° [9.13°]
		mean	0.329	0.642	0.638	0.640	0.188	0.102	20.137	0.772	1.021	0.058	0.196	0.296	55.12° [7.25°]
	unseen	best	0.377	0.852	0.821	0.833	0.534	0.092	20.994	0.778	0.890	0.052	0.137	0.395	58.79° [10.66°]
		mean	0.278	0.613	0.607	0.609	0.158	0.106	19.740	0.746	1.068	0.069	0.213	0.372	68.29° [9.28°]

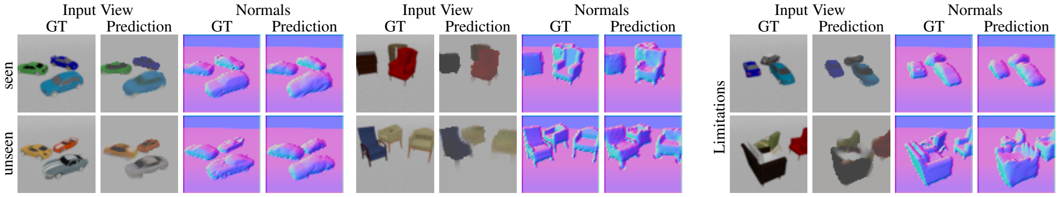

Figure 2: **Qualitative Results. Left:** We show reconstructions for scenes with either cars or chairs. Our model is able to decouple all objects from the background, and estimates plausible pose and shape of individual objects and learns to decode basic textures. **Right:** Due to ambiguities in a self-supervised setting, the model sometimes favors to adapt shape over the rotation estimate.

generates plausible reconstructions. Basic texture and shape of the objects are well recognized. We report mean and best results over five training runs in Tab. 1, where the best run is chosen according to F1 score on the validation set. Evaluation is performed on two different testsets: scenes containing (1) object instances with shapes and textures used for training and (2) unseen object instances.

We found that training performance depends on the chosen object category. While our model yields consistent high performance in all runs for car objects, training on chairs converges to local minima more easily. We further observed that our self-supervised model has difficulties to differentiate pseudo shape symmetries and would favor to adapt the texture over rotating the object in such a case (Fig. 2, right). This is more visible for chairs than for cars through deformed shape and image reconstructions. The median rotation error indicates better than chance prediction for the correct orientation for both categories. On scenes with instances that were not seen during training, our model often approximates the shapes with similar training instances.

We observed that training the shape decoder fully self-supervised in our pipeline is not straightforward and does not produce good results without further augmentations. Presumably this is due to the random initialization of the neural SDF shape representation which does not yield meaningful shapes for rendering at the beginning of training.

Our 3D scene model further naturally facilitates generation and manipulation of scenes by altering the latent representation. In Fig. 1, we show example operations like switching the positions of two objects, changing their shape, or removing an entire object. The explicit knowledge about 3D shape also allows us to reason about object penetrations when generating new scenes. Further qualitative results are provided in the supplementary material.

4 Discussion & Conclusion

Self-supervised learning is a promising field of research since it avoids the expensive collection of human-labeled data. Without regularizing assumptions, this setting typically leads to ill-conditioned problems. We propose a deep learning approach for multi-object scene representation learning and parsing. Our model jointly learns the tasks of object instance detection, instance segmentation, object pose estimation, and inference of 3D shape and texture in a single RGB image in a semi-supervised way. It infers a 3D scene representation by recursively parsing the image for shape, texture and poses of the objects. We pre-train the shape decoder network in a supervised way to confine the space of possible shapes. Besides this, no further ground truth data is required to learn the multi-object decomposition from RGB-D images itself in a self-supervised way. This requires a differentiable renderer which generates images based on the latent scene representation. In our semi-supervised approach, ambiguities can arise due to the decoupling of shape and texture. For instance, the network can choose to occlude the background partially with the shape but fix the image reconstruction by predicting background color in these areas. Rotations can only be learned up to a pseudo-symmetry by self-supervision when object shapes are rotationally similar and the subtle differences in shape or texture are difficult to differentiate in the image (Fig. 2, right). In such cases, the network can favor to adapt shape and texture over rotating the shape. Depending on the complexity of the scenes and the complex combination of loss terms, training can run into local minima in which objects are moved outside the image or fit the ground plane. Despite all these difficulties, our experiments demonstrate that our model well achieves scene parsing without explicit supervision on this task. We believe our approach provides an important step towards self-supervised learning of object-level 3D scene representations of complex scenes from real images.

References

[1] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[2] Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. MONet: Unsupervised scene decomposition and representation, 2019.

[3] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[4] S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E. Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3233–3241, Red Hook, NY, USA, 2016. Curran Associates Inc.

[5] Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[6] Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. SPACE: Unsupervised object-oriented scene representation via spatial attention and decomposition. In *Accepted for International Conference on Learning Representations (ICLR)*, 2020.

[7] Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. GENESIS: Generative scene inference and sampling with object-centric latent representations. In *Accepted for International Conference on Learning Representations (ICLR)*, 2020.

[8] Yanchao Yang, Yutong Chen, and Stefano Soatto. Learning to manipulate individual objects in an image. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 6557–6566. IEEE, 2020.

[9] Yiyi Liao, Katja Schwarz, Lars Mescheder, and Andreas Geiger. Towards unsupervised learning of generative models for 3d controllable image synthesis. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[10] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[11] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *Proceedings IEEE International Conf. on Computer Vision (ICCV)*, 2019.

[12] A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner, R. Pandey, S. Fanello, G. Wetzstein, J.-Y. Zhu, C. Theobalt, M. Agrawala, E. Shechtman, D. B Goldman, and M. Zollhöfer. State of the art on neural rendering. *Computer Graphics Forum (EG STAR 2020)*, 2020.

[13] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey, 2020.

[14] R. Wang, N. Yang, J. Stueckler, and D. Cremers. Directshape: Photometric alignment of shape priors for visual vehicle pose and shape estimation. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[15] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017.

[16] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *ArXiv*, abs/1512.03012, 2015.

[17] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, 2010.

[18] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2366–2374, Cambridge, MA, USA, 2014. MIT Press.

Supplementary Material

A Overview

In this supplementary material, we present further evaluation results. In particular, we present more qualitative results on ShapeNet scenes [16] in Fig. 4 and Fig. 5. We provide rotation error histograms in Fig. 3.

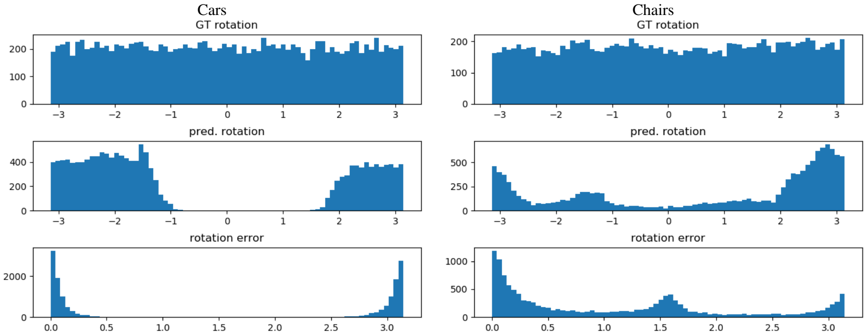

Figure 3: **Rotation Prediction on ShapeNet dataset [16].** From top to bottom: GT and predicted rotation angles for each dataset and resulting rotation angles. While values for GT rotation are naturally uniformly distributed over the entire range of $[-\pi, \pi]$ for all scenes, we found that predicted rotation estimates were often spread over a smaller sub-range. Peaks in the histogram for cars ($\sim \pi$) and chairs ($\sim \frac{\pi}{2}, \sim \pi$) indicate that the model got stuck in local minimum were it predicts a rotation up to a pseudo-symmetry.

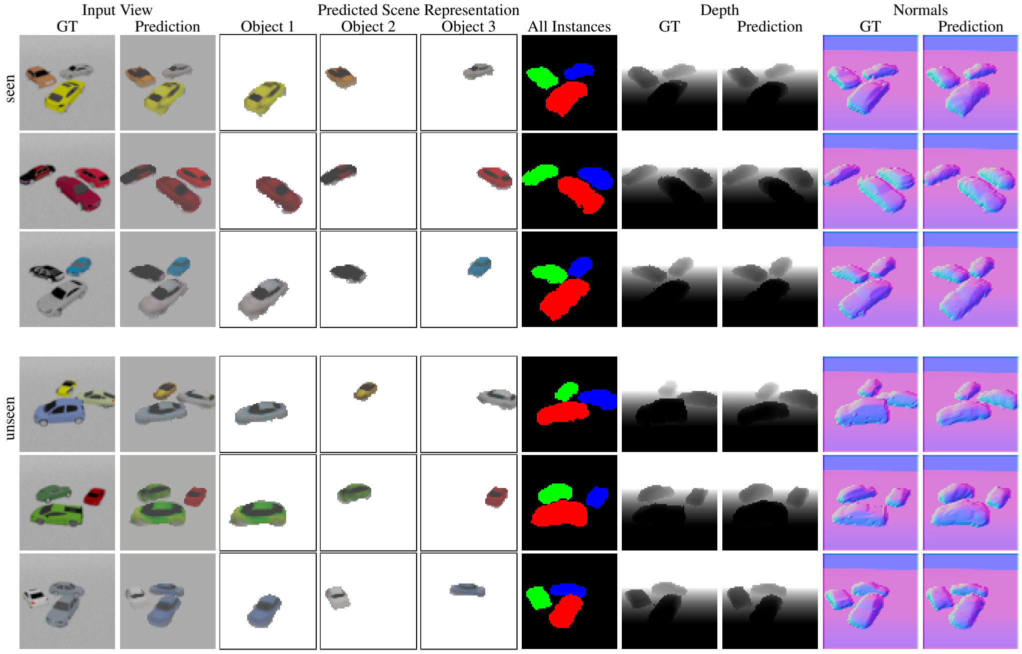

Figure 4: **Qualitative results on ShapeNet datasets [16] with car models.** Our model is able to decompose the scene into the individual objects and generates reasonable reconstruction for scenes with both seen (top) and unseen (bottom) object instances. For the latter case, it describes objects with similar shapes and textures is has seen in training.

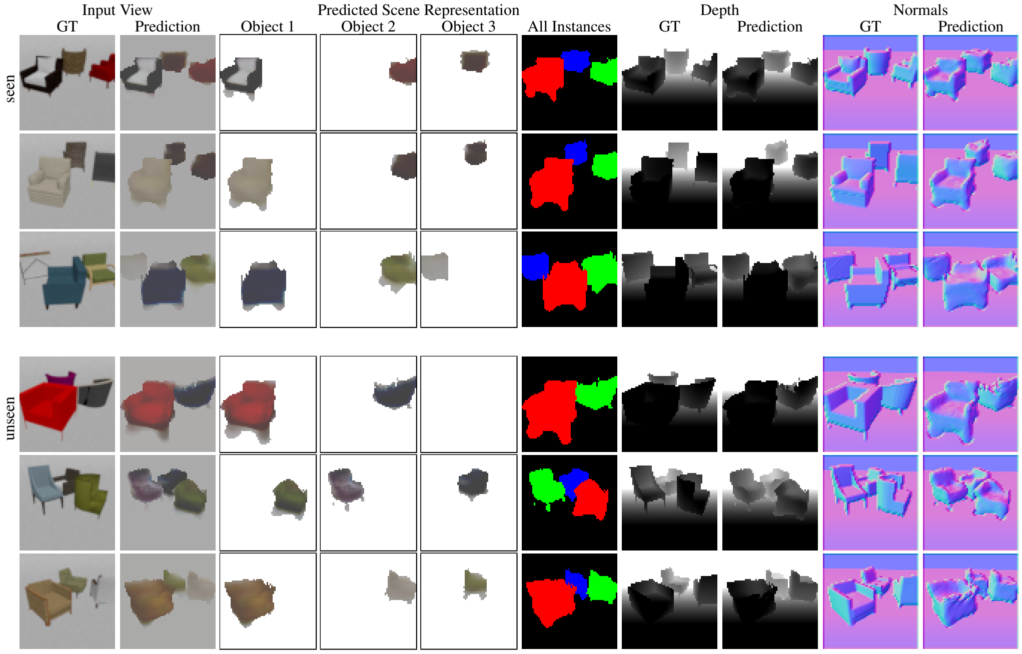

Figure 5: **Qualitative results on ShapeNet datasets [16] with chair models.** For the chair models it is more important to predict the correct rotation to infer a well matching shape than for other models in our datasets. The model still got easily trapped in local minima of 90-degree steps where it would rather adapt shape and texture reconstruction instead of the estimated rotation. Due to the low resolution as well as the discrete sampling by the renderer, our model is prone to miss fine structural elements like armrests or thin legs. We show examples with both seen (top) and unseen (bottom) objects.