Semi-Supervised Learning of Multi-Object 3D Scene Representations

Cathrin Elich [1 2 3] **Martin R. Oswald** [2] **Marc Pollefeys** [2 4] **Joerg Stueckler** [1]

Abstract

Representing scenes at the granularity of objects is a prerequisite for scene understanding and decision making. We propose a novel approach for learning multi-object 3D scene representations from images. A recurrent encoder regresses a latent representation of 3D shapes, poses and texture of each object from an input RGB image. The 3D shapes are represented continuously in function-space as signed distance functions which we efficiently pre-train from example shapes in a supervised way. By differentiable rendering we then train our model to decompose scenes self-supervised from RGB-D images. Our approach learns to decompose images into the constituent objects of the scene and to infer their shape, pose and texture from a single view. We evaluate the accuracy of our model in inferring the 3D scene layout and demonstrate its generative capabilities.

1. Introduction

Humans have the remarkable capability to decompose scenes into its constituent objects and to infer object properties such as 3D shape and texture from just a single view. Providing intelligent systems with similar capabilities is a long-standing goal in artificial intelligence. Such representations would facilitate object-level description, abstract reasoning and high-level decision making. Moreover, object-level scene representations could improve generalization for learning in downstream tasks such as robust object recognition or action planning.

Previous work on learning-based scene representations focused on single-object scenes (Sitzmann et al., 2019) or neglected to model the 3D geometry of the scene and the objects explicitly (Burgess et al., 2019; Greff et al., 2019; Eslami et al., 2016). 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 detection, instance segmentation, object pose estimation and inference of 3D shape and texture in single RGB images. Inspired by (Park et al., 2019; Oechsle et al., 2019; Sitzmann et al., 2019), 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. We propose a novel differentiable renderer which efficiently generates color and depth images as well as instance masks from the object-wise scene representation. 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 to embed a collection of meshes in continuous signed distance function (SDF) shape embeddings as in DeepSDF (Park et al., 2019). 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. 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 to show its capabilities with shapes such as geometric primitives and vehicles and demonstrate the properties of our geometric and semi-supervised learning approach for scene representation.

In summary, we make the following **contributions**:
(1) We propose a novel model to learn representations of multi-object scenes. Our model describes the scene by explicitly encoding object poses, 3D shapes and texture. To the best of our knowledge, our approach is the first to jointly learn object instance detection, instance segmentation, object localization, and inference of 3D shape and texture in a

[1]Max Planck Institute for Intelligent Systems, Tuebingen, Germany [2]Department of Computer Science, ETH Zurich [3]Max Planck ETH Center for Learning Systems [4]Microsoft Mixed Reality and AI Zurich Lab. Correspondence to: Cathrin Elich <cathrin.elich@tuebingen.mpg.de>.

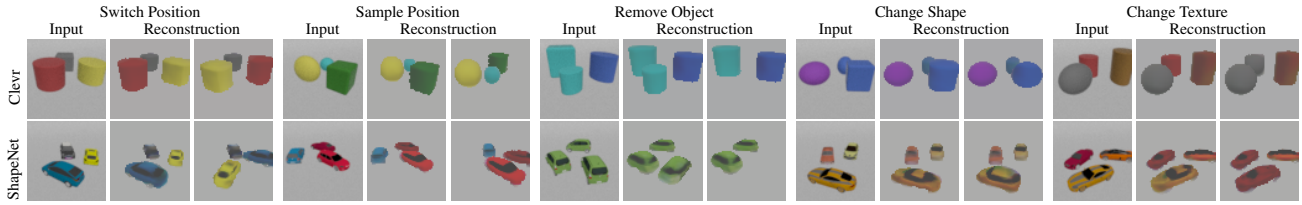

Figure 1. **Example scenes with object manipulation.** For each example, we input the left images and compute the middle one as standard reconstruction. After the manipulation in the latent space, we obtain the respective right image. Plausible new scene configurations are shown on the Clevr dataset (Johnson et al., 2017) (top) and on composed ShapeNet models (Chang et al., 2015) (bottom).

single RGB image via self-supervised scene decomposition. **(2)** Our model is trained via differentiable rendering to decode the latent representation back into images. We propose a novel differentiable renderer using sampling-based ray-casting for deep SDF shape embeddings which renders color and depth images as well as instance segmentation masks. **(3)** By representing 3D geometry explicitly, our approach naturally respects occlusions and collisions between objects and facilitates manipulation of the scene within the latent space. We demonstrate properties of our geometric model for scene representation and augmentation, and discuss advantages over multi-object scene representation methods which model geometry implicitly.

2. Related Work

Deep learning of single object geometry. Several recent 3D learning approaches represent single object geometry by implicit surfaces of occupancy or signed distance functions which are discretized in 3D voxel grids (Kar et al., 2017; Tulsiani et al., 2017; Wu et al., 2016; Gadelha et al., 2017; Qi et al., 2016; Jimenez Rezende et al., 2016; Choy et al., 2016; Shin et al., 2018; Xie et al., 2019). Voxel representations typically waste significant memory and computation resources in empty scene parts. This limits their resolution and capabilities to represent fine details. Other methods represent shapes with point clouds (Qi et al., 2017; Achlioptas et al., 2018), meshes (Groueix et al., 2018), deformations of shape primitives (Henderson & Ferrari, 2019) or multiple views (Tatarchenko et al., 2016). In continuous representations, neural networks are trained to directly predict signed distance (Park et al., 2019; Xu et al., 2019; Sitzmann et al., 2019), occupancy (Mescheder et al., 2019; Chen & Zhang, 2019), or texture (Oechsle et al., 2019) at continuous query points. We use such representations for individual objects.

Deep learning of multi-object scene representations. Self-supervised learning of multi-object scene representations from images recently gained significant attention in the machine learning community. MONet (Burgess et al., 2019) presents a multi-object network which decomposes the scene using a recurrent attention network and an object-wise autoencoder. It embeds images into object-wise latent representations and overlays them into images with a neural decoder. (Yang et al., 2020) improve upon this work. (Greff et al., 2019) use iterative variational inference to optimize object-wise latent representations using a recurrent neural network. SPAIR (Crawford & Pineau, 2019) and SPACE (Lin et al., 2020) extend the attend-infer-repeat approach (Eslami et al., 2016) by laying a grid over the image and estimating the presence, relative position, and latent representation of objects in each cell. In GENESIS (Engelcke et al., 2020), the image is recurrently encoded into latent codes per object in a variational framework. (Locatello et al., 2020) propose Slot Attention for decomposing scenes into objects. In contrast to our method, the above methods do not represent the 3D geometry of the scene explicitly. (Liao et al., 2020; Nguyen-Phuoc et al., 2020) generate novel 3D scenes instead but do not explain input views like we do. Recently, (Henderson & Lampert, 2020; Li et al., 2020) exploit multiple images to describe 3D scenes.

Supervised learning for object instance segmentation, pose and shape estimation. Loosely related are supervised methods that segment object instances (Hou et al., 2019; Prabhudesai et al., 2020), estimate their poses (Xiang et al., 2017) or recover their 3D shape (Gkioxari et al., 2019; Kniaz et al., 2020). In Mesh R-CNN (Gkioxari et al., 2019), objects are detected in bounding boxes and a 3D mesh is predicted for each object. The method is trained supervised on images with annotated object shape ground truth.

Neural and differentiable rendering. (Eslami et al., 2018) encode images into latent representations which can be aggregated from multiple views. Scene rendering is deferred to a neural network which is trained to decode the latents into images from examples. Several differentiable rendering approaches have been proposed using voxel occupancy grids (Tulsiani et al., 2017; Gadelha et al., 2017; Jimenez Rezende et al., 2016; Yan et al., 2016; Gwak et al., 2017; Zhu et al., 2018; Wu et al., 2017; Nguyen-Phuoc et al., 2018), meshes (Kato et al., 2018; Loper & Black, 2014; Chen et al., 2019; Delaunoy & Prados, 2011; Ramamoorthi & Hanrahan, 2001; Meka et al., 2018; Athalye et al., 2018; Richardson et al., 2016; Liu et al., 2019; Henderson & Ferrari, 2019), signed distance functions (Sitzmann et al., 2019), or point clouds (Lin et al., 2018; Yifan et al., 2019). Recent literature overviews are (Tewari et al., 2020; Kato et al., 2020). In our approach, we find depth and mask

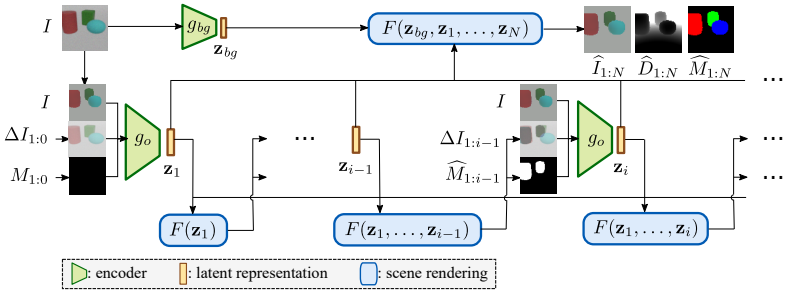

Figure 2. **Multi-object 3D scene representation network.** The image is sequentially encoded into object representations using an encoder network g_o. The object encoders additionally receive image and mask compositions $(\Delta I, M)$ generated from the previous object encodings. A differentiable renderer based decoder F composes images and masks from the encodings of previous steps. The background is encoded from the image in parallel and used in the final scene reconstruction.

values through equidistant sampling along the ray.

3. Method

We propose an autoencoder architecture which embeds images into object-wise scene representations (see Fig. 2 for an overview). Our recurrent network architecture does not require supervision on bounding boxes like object detection methods such as Yolo (Redmon et al., 2016) or Faster R-CNN (Ren et al., 2015). Each object is explicitly described by its 3D pose and latent embeddings for both its shape and textural appearance. Given the object-wise scene description, a decoder composes the images back from the latent representation through differentiable rendering. We train our autoencoder-like network in a self-supervised way from RGB-D images.

Scene Encoding. The network infers a latent $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_N, \mathbf{z}_{bg})$ which decomposes the scene into object latents $\mathbf{z}_i \in \mathbb{R}^d$, $i \in \{1, \ldots, N\}$ and a background component $\mathbf{z}_{bg} \in \mathbb{R}^{d_{bg}}$ where d, d_{bg} are the dimensionality of the object and background encodings and N is the object count.

Object are sequentially encoded by a deep neural network $\mathbf{z}_i = g_o(I, \Delta I_{1:i-1}, \widehat{M}_{1:i-1})$ (see Fig. 2). We share the same object encoder network and weights between all objects. To guide the encoder to regress the latent representation of one object after the other, we forward additional information about already reconstructed objects. Specifically, we decode the previous object latents into object composition images, depth images and occlusion masks $(\widehat{I}_{1:i-1}, \widehat{D}_{1:i-1}, \widehat{M}_{1:i-1}) := F(\mathbf{z}_{bg}, \mathbf{z}_1, \ldots, \mathbf{z}_{i-1})$. They are generated by F using differentiable rendering which we detail in the subsequent paragraph. We concatenate the input image I with the difference image $\Delta I_{1:i-1} := I - \widehat{I}_{1:i-1}$ and occlusion masks $\widehat{M}_{1:i-1}$, and input this to the encoder for inferring the representation of object i.

The object encoding $\mathbf{z}_i = (\mathbf{z}_{i,sh}^\top, \mathbf{z}_{i,tex}^\top, \mathbf{z}_{i,ext}^\top)^\top$ decomposes into encodings for shape $\mathbf{z}_{i,sh}$, textural appearance $\mathbf{z}_{i,tex}$, and 3D extrinsics $\mathbf{z}_{i,ext}$ (see Fig. 3). The shape encoding $\mathbf{z}_{i,sh} \in \mathbb{R}^{D_{sh}}$ parametrizes the 3D shape represented by a DeepSDF autodecoder (Park et al., 2019). Similarly, the texture is encoded in a latent vector $\mathbf{z}_{i,tex} \in \mathbb{R}^{D_{tex}}$ which

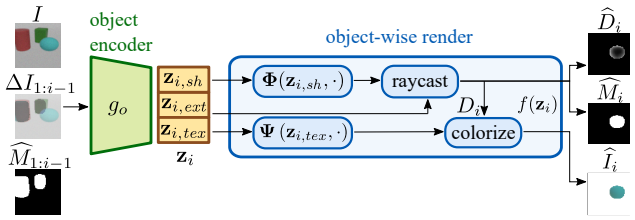

Figure 3. **Object-wise encoding and rendering.** We feed the input image, scene composition images and masks of the previously found objects to an object encoder network g_o which regresses the encoding of the next object $\mathbf{z}_i$. The object encoding decomposes into shape $\mathbf{z}_{i,sh}$, extrinsics $\mathbf{z}_{i,ext}$ and texture latents $\mathbf{z}_{i,tex}$. The shape latent parametrizes an SDF function network $\mathbf{\Phi}$ which we use in combination with the pose and scale of the object encoded in $\mathbf{z}_{i,ext}$ for raycasting the object depth and mask using our differentiable renderer f. Finally, the color of the pixels is found with a texture function network $\mathbf{\Psi}$ parametrized by the texture latent.

is used by the decoder to obtain color values for each pixel that observes the object. Object position $\mathbf{p}_i = (x_i, y_i, z_i)^\top$, orientation θ_i and scale s_i are regressed with the extrinsics encoding $\mathbf{z}_{i,ext} = (\mathbf{p}_i^\top, z_{\cos,i}, z_{\sin,i}, s_i)^\top$. The object pose $\mathbf{T}_w^o(\mathbf{z}_{i,ext}) = \begin{pmatrix} s_i \mathbf{R}_i^\top & -\mathbf{R}_i^\top \mathbf{p}_i \\ \mathbf{0} & 1 \end{pmatrix}$ is parametrized in a world coordinate frame with known transformation $\mathbf{T}_c^w$ from the camera frame.

We assume the objects are placed upright and model rotations around the vertical axis with angle $\theta_i = \arctan(z_{\sin,i}, z_{\cos,i})$ and corresponding rotation matrix $\mathbf{R}_i$. We use a two-parameter representation for the angle as suggested in (Zhou et al., 2019). We scale the object shape by the factor $s_i \in [s_{\min}, s_{\max}]$ which we limit in an appropriate range using a sigmoid squashing function.

The background encoder $g_{bg} := \mathbf{z}_{bg} \in \mathbb{R}^{d_{bg}}$ regresses the uniform color of the background plane, i.e. $d_{bg} = 3$. We assume the plane extrinsics and hence its depth image is known in our experiments.

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 $(\widehat{I}_i, \widehat{D}_i, \widehat{M}_i) := f(\mathbf{z}_i)$ determines color image $\widehat{I}_i$, depth image $\widehat{D}_i$ and occlusion mask $\widehat{M}_i$ from each object encoding independently (see Fig. 3). The renderer determines the depth at each pixel $\mathbf{u} \in \mathbb{R}^2$ (in normalized image coordinates) through raycasting in the SDF shape representation. Inspired by (Wang et al., 2020), we trace the SDF zero-crossing along the ray by sampling points $\mathbf{x}_j := (d_j\mathbf{u}, d_j)^\top$ in equal intervals $d_j := d_0 + j\Delta d, j \in \{0, \dots, N-1\}$ with start depth d_0. The points are transformed to the object coordinate system by $\mathbf{T}_c^o(\mathbf{z}_{i,ext}) := \mathbf{T}_w^o(\mathbf{z}_{i,ext})\mathbf{T}_c^w$. Subsequently, the signed distance ϕ_j to the shape at these transformed points is obtained by evaluating the SDF function network $\mathbf{\Phi}\left(\mathbf{z}_{i,sh}, \mathbf{T}_c^o(\mathbf{z}_{i,ext})\mathbf{x}_j\right)$. Note that the SDF network is also parametrized by the inferred shape latent of the object. The algorithm finds the zero-crossing at the first pair of samples with a sign change of the SDF $\mathbf{\Phi}$. The sub-discretization accurate location $\mathbf{x}(\mathbf{u})$ of the surface is found through linear interpolation of the depth regarding the corresponding SDF values of these points. The depth at a pixel $D_i(\mathbf{u})$ is given by the z coordinate of the raycasted point $\mathbf{x}(\mathbf{u})$ on the object surface in camera coordinates. If no zero crossing is found, the depth is set to a large constant. The binary occlusion mask $M_i(\mathbf{u})$ is set to 1 if a zero-crossing is found at the pixel and 0 otherwise. The pixel color $I_i(\mathbf{u})$ is determined using a decoder network $\mathbf{\Psi}$ which receives the texture latent $\mathbf{z}_{i,tex}$ of the object and the raycasted 3D point $\mathbf{x}(\mathbf{u})$ in object coordinates as inputs, i.e. $I_i(\mathbf{u}) = \mathbf{\Psi}\left(\mathbf{z}_{i,tex}, \mathbf{T}_c^o(\mathbf{z}_{i,ext})\mathbf{x}(\mathbf{u})\right)$. We speed up the raycasting process by only considering pixels that lie within the projected 3D bounding box of the object shape representation. This bounding box is known since the SDF function network is trained with meshes that are normalized to fit into a unit cube with a constant padding. Note that this rendering procedure is implemented using differentiable operations which makes it fully differentiable for the shape, color and extrinsics encodings of the object.

The scene images, depth images and occlusion masks $\left(\widehat{I}_{1:n}, \widehat{D}_{1:n}, \widehat{M}_{1:n}\right) = F(\mathbf{z}_{bg}, \mathbf{z}_1, \dots, \mathbf{z}_n)$ are composed from the individual objects $1, \dots, n$ with $n \leq N$ and the decoded background through z-buffering. We initialize them with the background color, depth image of the empty plane and empty mask. Recall that the background color is regressed by the encoder network. For each pixel $\mathbf{u}$, we search the occluding object i with the smallest depth at the pixel. If such an object exists, we set the pixel's values in $\widehat{I}_{1:N}, \widehat{D}_{1:N}, \widehat{M}_{1:N}$ to the corresponding values in the object images and masks.

Training. We train our network architecture in two stages. In a first stage, we learn the SDF function network from a collection of meshes. The second stage uses the pre-trained SDF models to learn the remaining components for the object-wise scene decomposition and rendering network. We train the SDF networks according to (Park et al., 2019) from a collection of meshes and sample points in a volume around the object and on the object surface. We normalize the size of the input meshes to fit into the unit cube with constant padding $\epsilon = 0.1$.

Our multi-object network is trained self-supervised from RGB-D images containing example scenes composed of multiple objects. To this end, we minimize the loss function

$$L_{total} = \lambda_I L_I + \lambda_D L_D + \lambda_{gr} L_{gr} + \lambda_{sh} L_{sh}, \quad (1)$$

which is a weighted sum of multiple sub-loss functions:

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

$$L_D = \frac{1}{|\Omega|} \sum_{\mathbf{u} \in \Omega} \left\| G\left(\widehat{D}_{1:N}\right)(\mathbf{u}) - G(D_{gt})(\mathbf{u}) \right\| \quad (3)$$

$$L_{gr} = \sum_i \max(0, -z_i) + \max(0, -\phi_i(z_i')) \quad (4)$$

$$L_{sh} = \sum_i \|\mathbf{z}_{i,sh}\|^2 \quad (5)$$

In particular, L_I is the mean squared error on the image reconstruction with Ω being the set of image pixels and I_{gt} the ground-truth color image. The depth reconstruction loss L_D penalizes deviations from the ground-truth depth D_{gt}. We apply Gaussian smoothing $G(\cdot)$ for which we decrease the standard deviation over time. 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, z_i' the corresponding projection onto the ground plane, and $\phi_i(\mathbf{x}_k) := \mathbf{\Phi}\left(\mathbf{z}_{i,sh}, \mathbf{T}_c^o(\mathbf{z}_{i,ext})\mathbf{x}_k\right)$. The shape regularization loss is scheduled with time-dependent weighting. This prevents the network from learning to generate unreasonable extrapolated shapes in the initial phases of the training, but lets the network refine them over time.

We use a CNN for both the object and the background encoder. Both consist of multiple convolutional layers with kernel size $(3, 3)$ and strides $(1, 1)$ each followed by ReLU activations and $(2, 2)$ max-pooling. The subsequent fully-connected layers yield the encodings for objects and background. Similar to (Park et al., 2019), we use multi-layer fully-connected neural networks for the shape decoder $\mathbf{\Phi}$ and texture decoder $\mathbf{\Psi}$. See suppl. material for more infos.

4. Experiments

We evaluate our approach on synthetic scenes based on the Clevr dataset (Johnson et al., 2017) and scenes generated with ShapeNet models (Chang et al., 2015). The Clevr-based scenes contain images with a varying number of colored shape primitives (spheres, cylinders, cubes) on a planar single-colored background. We modify the data

generation of Clevr in a number of aspects: **(1)** We remove shadows and additional light sources and only use the Lambertian rubber material for the objects' surfaces. **(2)** To further increase shape variety, we apply random scaling along the principal axes of the primitives. **(3)** An object might be completely hidden behind another one. Hence, the network needs to learn to hide single objects. We generate several multi-object datasets. Each dataset contains scenes with a specific number of objects which we choose from two to five. Each dataset consists of 12.5K images with a size of 64×64 pixels. Objects are randomly rotated and placed in a range of $[-1.5, 1.5]^2$ on the ground plane while ensuring that any two objects do not intersect. Additionally to the RGB images, we also generate depth maps for training as well as instance masks for evaluation. The images are split into 9K training, 1K validation, and 2.5K testing examples. For the pre-training of the DeepSDF (Park et al., 2019) network, we generate a small set of nine shapes per category with different scaling along the axes for which we generate ground truth SDF samples. Different to (Park et al., 2019), we sample a higher ratio of points randomly in the unit cube instead of close to the surface. We also evaluate on scenes depicting either cars or armchairs as well as a mixed set consisting of mugs, bottles and cans (tabletop) from the ShapeNet model set. 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. We increase the size of the dataset to (18K/2K/5K). The evaluation is performed on two different test sets: (1) with known shapes and (2) with new objects.

Network Parameters. For the Clevr / ShapeNet datasets, the object encoding dimension is set to $D_{sh} = 8/16$, and $D_{tex} = 7/15$. The shape decoder is pre-trained for 10K epochs. We decrease the loss weight λ_{sh} from $0.025/0.1$ to $0.0025/0.01$ during the first 500K iterations. The remaining weights are fixed to $\lambda_I = 1.0$, $\lambda_{depth} = 0.1/0.05$, $\lambda_{gr} = 0.01$. We add Gaussian noise to the input RGB images. Depth images are clipped at a distance of 12. The renderer evaluates at 12 steps along each ray. Gaussian smoothing is applied with kernel size 16 and decreasing sigma from $\frac{16}{3}$ to $\frac{1}{2}$ in 250K steps. We use the ADAM optimizer (Kingma & Ba, 2014) with learning rate 0.0001 and batch size 8 to train for a dataset-specific number of epochs (see supplementary material for more details).

Evaluations Metrics. We evaluate the learning of object-level 3D scene representations using measures for instance segmentation, image reconstruction, and pose estimation.

To evaluate our models' capability to recognize objects that best explain the input image, we consider established instance segmentation metrics. An object is considered to be correctly segmented if the intersection-over-union (IoU) score between ground truth and predicted mask is higher than a threshold τ. To account for occlusions, only objects that occupy at least 25 pixels are taken into account. We report average precision ($AP_{0.5}$), average recall ($AR_{0.5}$), $F1_{0.5}$-score for a fixed $\tau = 0.5$ as well as the mean AP over thresholds in range $[0.5, 0.95]$ with stepsize 0.05 (Everingham et al., 2010). Furthermore, we list the ratio of scenes were all visible objects were found w.r.t. $\tau = 0.5$ (allObj).

Next, we evaluate the quality of both the RGB and depth reconstruction of 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 (Eigen et al., 2014) 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. $\tau = 0.5$. More details on the metrics are provided in the supplementary material. We show results over five runs per configuration and report the mean.

4.1. Clevr Dataset

In Fig. 4, we show reconstructed images, depth and normal maps on the Clevr (Johnson et al., 2017) scenes. Our model provides a complete reconstruction of the individual objects although they might be partially hidden in the image. The network can infer the color of the objects correctly and gets a basic idea about shading (e.g. that spheres are darker on the lower half) and coarse texture. The shape characteristics such as extent, edges or curved surfaces are well recognized. Our model needs to fill all object slots. We sometimes observed that it fantasizes and hides additional objects behind others. Some reconstruction artifacts at object boundaries are due to rendering hard transitions between objects and background. More results and typical failure cases are shown in the supplementary material.

Our 3D scene model 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. Specifically, we evaluate an object intersection loss L_{int} on the newly sampled scenes to filter out those that turn out to be unrealistic due to an intersection between objects (see supplementary material for details).

Ablation Study. We evaluate various components of our model on the Clevr dataset with three objects. In Table 1, we evaluate on training settings where we left out each of the loss functions and also demonstrate the benefit of Gaussian smoothing (denoted by G) on the image reconstructions.

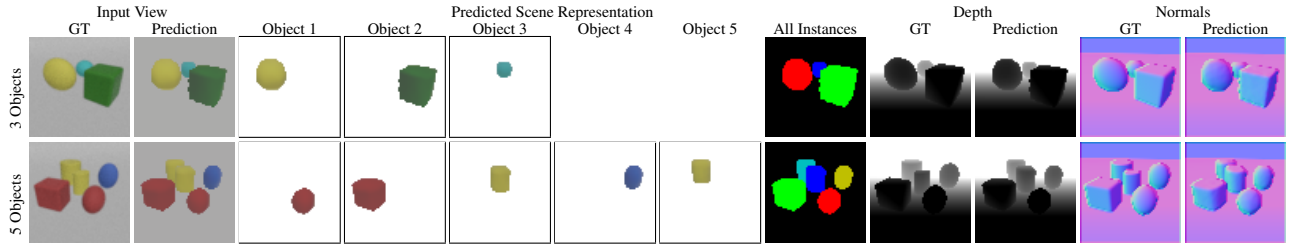

Figure 4. **Qualitative results on the Clevr dataset (Johnson et al., 2017) with three and five objects.** Our object-wise scene representation decouples all objects from the background and assign each object a separate instance label, geometry and appearance.

Table 1. **Results on the Clevr dataset (Johnson et al., 2017).** The combination of our proposed loss with Gaussian blur is essential to guide the learning of scene decomposition and object-wise representations. We highlight best (bold) and second best (underlined) results for each measure. Using different maximum numbers of objects in our network, we further train our model on scenes with 2, 4, or 5 objects. Despite the increased difficulty for larger number of objects, our model recognizes most objects in scenes with two to five objects. Models trained with fewer objects can successfully explain scenes with a larger number of objects (*# obj=o_{train}/o_{test}*).

	Instance Reconstruction					Image Reconstruction			Depth Reconstruction			Pose Est.
	mAP ↑	$AP_{0.5}$ ↑	$AR_{0.5}$ ↑	$F1_{0.5}$ ↑	allObj ↑	RMSE ↓	PSNR ↑	SSIM ↑	RMSE ↓	AbsRD ↓	SqRD ↓	Err_{pos}
# obj=3/3, w/o L_I	<u>0.686</u>	<u>0.941</u>	<u>0.879</u>	<u>0.899</u>	<u>0.709</u>	0.199	14.176	0.713	<u>0.595</u>	<u>0.023</u>	<u>0.073</u>	<u>0.159</u>
# obj=3/3, w/o L_D	0.023	0.086	0.076	0.078	0.008	<u>0.085</u>	<u>22.142</u>	<u>0.837</u>	2.745	0.231	1.061	1.341
# obj=3/3, w/o L_{sh}	0.01	0.032	0.027	0.028	0.001	0.13	17.907	0.763	1.455	0.147	0.556	0.676
# obj=3/3, w/o L_{gr}	0.09	0.195	0.205	0.198	0.008	0.09	21.163	0.799	1.159	0.087	0.32	0.81
# obj=3/3, w/o G	0.164	0.296	0.161	0.199	0.001	0.114	19.065	0.792	1.331	0.112	0.441	0.182
# obj=3/3, full	**0.712**	**0.949**	**0.942**	**0.943**	**0.85**	**0.049**	**26.466**	**0.914**	**0.554**	**0.019**	**0.061**	**0.155**
# obj=2/2	0.782	0.977	0.963	0.967	0.928	0.039	28.389	0.941	0.432	0.012	0.04	0.138
# obj=4/4	0.688	0.941	0.919	0.926	0.746	0.054	25.632	0.899	0.584	0.022	0.064	0.151
# obj=5/5	0.604	0.895	0.861	0.872	0.539	0.061	24.568	0.876	0.593	0.025	0.067	0.149
# obj=3/2	0.756	0.974	0.969	0.97	0.942	0.041	28.011	0.937	0.452	0.013	0.044	0.14
# obj=3/4	0.613	0.883	0.853	0.863	0.512	0.06	24.669	0.88	0.665	0.028	0.083	0.179
# obj=3/5	0.478	0.775	0.71	0.735	0.212	0.072	23.093	0.841	0.69	0.033	0.086	0.201

At the beginning of training, the shape regularization loss is crucial to keep the shape encoder close to the pretrained DeepSDF shape space and to prevent it from diverging due to the inaccurate pose estimates of the objects. Applying and decaying Gaussian blur distributes gradient information in the images beyond the object masks and allows the model to be trained in a coarse-to-fine manner. This helps the model to localize the various objects in the scene. Moreover, the depth loss is essential for learning the scene decomposition. Without this loss, the network can simply describe several objects using a single object with more complex texture. The usage of the ground loss prevents the model from fitting objects into the ground plane. The image reconstruction loss plays only a minor part for the scene decomposition task but is merely responsible for learning the texture of the objects. Using all our proposed loss functions yields best results over all metrics. Remarkably, our model is able to find objects at high recall rates (0.942 AR at 50% IoU).

Object Count. To demonstrate generalization to different numbers of objects we report results with varying maximum object numbers in Tab. 1. We train the models with the corresponding number of objects in the dataset. On average it is easier for our model to find and describe objects in less crowded scenes, while it still performs with high accuracy for five objects.

Due to the sequential architecture of our model, it can even be extended to parse scenes with more objects than it has been trained for. Since we use a shared encoder for all objects, we can simply reset the number of encoding rollouts to the number of objects in the test data. Note that we assume the maximum number of objects to be known. Although our model would be able to hide redundant objects behind already reconstructed ones without this explicit change, it cannot reconstruct additional objects. Our model yields reasonable results, but performs best for similar object numbers in training and testing. The achieved average recall and allObj measures indicate that the model is able to detect the objects at good rates. For instance, for # obj=3/5, we find all objects in about 21% cases but overall 71% of the objects according to $AR_{0.5}$. Extended quantitative and qualitative results can be viewed in the supplementary material.

4.2. ShapeNet Dataset

Our composed multi-object variant of ShapeNet (Chang et al., 2015) models is more difficult in shape and texture variation than Clevr (Johnson et al., 2017). For some object categories such as cups or armchairs, training can converge

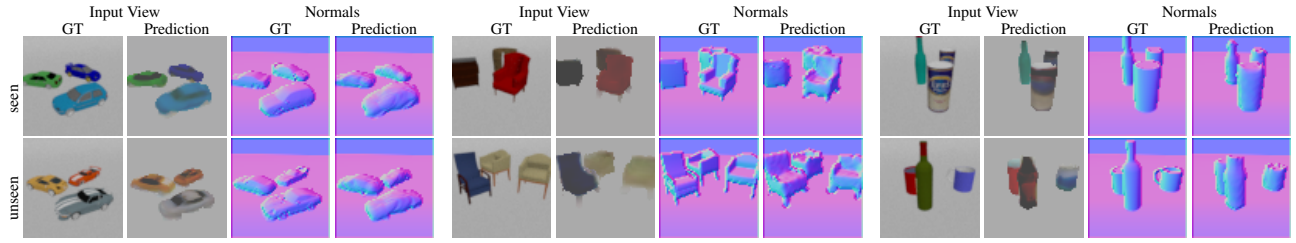

Figure 5. **Qualitative results on ShapeNet (Chang et al., 2015).** Our model obtains a good scene understanding if confronted with more difficult objects (cars, armchairs) and even handles objects from different categories (tabletop scenes with mugs, bottles and cans). It is able to estimate plausible pose and shape of individual objects and learns to decode more complex textures.

			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°]
tabletop	*seen*	best	0.628	0.936	0.870	0.895	0.659	0.057	25.242	0.908	0.786	0.026	0.132	0.182	89.14°
		mean	0.394	0.565	0.537	0.546	0.251	0.078	22.871	0.861	1.022	0.050	0.231	0.155	88.53°
	unseen	best	0.435	0.839	0.816	0.823	0.569	0.083	21.807	0.840	1.034	0.044	0.224	0.275	89.25°
		mean	0.285	0.530	0.521	0.523	0.237	0.102	20.160	0.800	1.172	0.061	0.291	0.238	89.99°

Table 2. **Evaluation on scenes with ShapeNet objects (Chang et al., 2015).** Results for scenes containing objects from different categories. We differentiate between scenes that consist of shapes that were seen during training and novel objects. We show mean and best outcome over five runs.

to local minima. We report mean and best results over five training runs in Tab. 2, 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 show several scene reconstructions in Fig. 5 and in the supplementary material.

For the cars, our model yields consistent performance in all runs with comparable decomposition results to our Clevr experiments. However, we found that cars exhibit a pseudo-180-degree shape symmetry which was difficult for our model to differentiate. Especially for small objects in the background, it favors to adapt the texture over rotating the object. For the armchair shapes, our model finds local minima in pseudo-90-degree symmetries. The median rotation error indicates better than chance prediction for the correct orientation. Rotation error histograms can be found in the supplementary material. For approximately correct rotation predictions, we found that our model was able to differentiate between basic shape types but often neglected finer details like thin armrests which are difficult to differentiate in the images. Our tabletop dataset provides another type of challenge: the network needs to distinguish different object categories with larger shape and scale variation. For this setting, we added further auxiliary losses to penalize object positions outside of the image view as well as object intersections (see supplementary material for details). Our model is able to predict the different shape types with coarse textures. On scenes with instances that were not seen during training, our model often approximates the shapes with similar training instances.

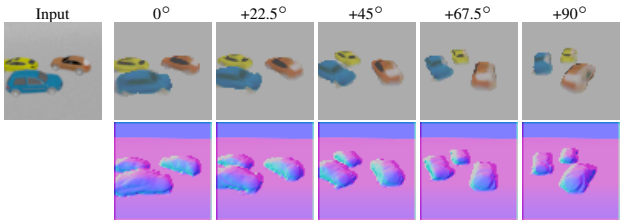

Figure 6. **Novel view renderings.** Our model is able to generate new scene renderings for largely rotated camera views from just a single input RGB image. While we noticed a reduced texture accuracy for unseen object parts compared to visible parts, the normal maps are generally good and demonstrate that our model obtains a good 3D structural understanding of the scene.

Due to the learned 3D structure, our model is able to render novel views from a scene given a single image (see Fig. 6). Although our model never saw multiple views of the same scene during training and is not tuned for this task, we obtain reasonable results for both scene geometry and appearance. We observe a lower reconstruction accuracy for invisible scene parts, especially for the texture.

4.3. Real Data

We further evaluated our model on real images of toy cars and wooden building blocks (see Fig. 7) as well as on the real block tower dataset from (Lerer et al., 2016) (see Fig. 8).

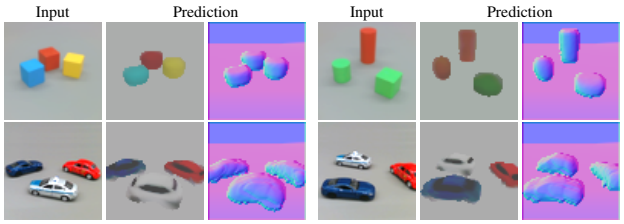

Figure 7. **Evaluation on real images.** We show results on real images by our model that was trained on synthetic data. We notice that our model is able to capture the coarse scene layout and shape properties of the objects. However, challenges arise due to domain, lighting, camera intrinsics and view point changes indicating interesting directions for future research.

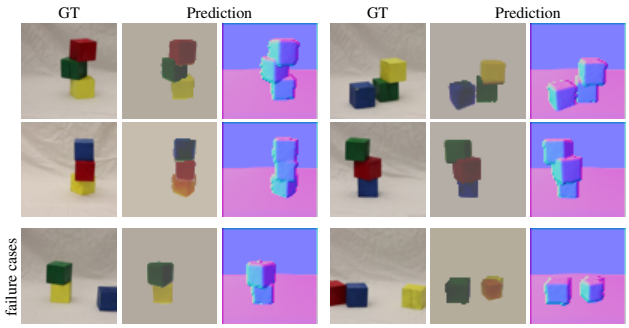

Figure 8. **Parsing real images of block towers (Lerer et al., 2016).** We trained our model on synthetic data containing stacked cubes and show reconstruction results on real images. Overall, our model recognizes the scene configuration well. Occasionally objects are missed, especially if they are close to the image boundary. Further details and results are in the supplementary material.

For the former dataset, we adjusted brightness and contrast of the photos to visually match the background color of the synthetic data. For the block tower dataset, images were cropped and scaled, and in contrast to all other experiments the 3D shape autodecoder was trained only on cubes. Despite different camera and image properties, our model decomposes the scenes into individual objects and obtains a coarse understanding about their shape and appearance without any domain adaptation or fine-tuning on real data.

Limitations. We show typical failure cases of our approach in Fig. 9. Self-supervised learning without regularizing assumptions leads typically to ill-conditioned problems. We use a pre-trained 3D shape space to confine the possible shapes, impose a multi-object decomposition of the scene, and use a differentiable renderer of the latent representation. In our self-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. In such cases, the network can favor to adapt 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. Currently, the network is trained for a maximum number of objects. If all objects in the scene are explained, it hides further objects which could be alleviated by learning a stop criterion.

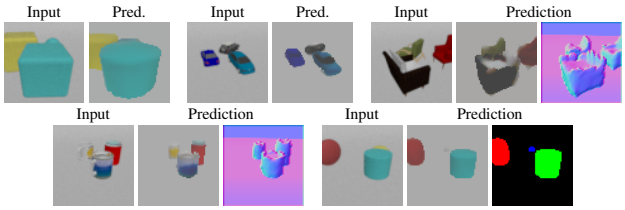

Figure 9. **Limitations**. Input and output pairs for typical failure cases and limitations of our method due to ambiguities for self-supervised learning. See text for details.

5. Conclusion

We propose a novel deep learning approach for multi-object scene representation learning and parsing. Our approach infers the 3D structure of a scene from a single RGB image by recursively parsing the image for shape, texture and poses of the objects. A differentiable renderer allows images to be generated from the latent scene representation and the network to be trained semi-supervised from RGB-D images. Object shapes are represented by signed distance functions. We employ pre-trained shape spaces that are represented by deep neural networks using a continuous function representation. Our experiments demonstrate the ability of our model to parse scenes with various object counts and shapes. We provide an ablation study to motivate design choices and discuss assumptions and limitations of our approach. We demonstrate the advantages of our model to reason about the underlying 3D space of a seen scene by performing explicit manipulation on the individual objects or rendering novel views. To the best of our knowledge, our approach is the first to jointly learn 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. We believe our approach provides an important step towards self-supervised learning of object-level 3D scene parsing and generative modeling of complex scenes from real images. The usage of synthetic data allows us to evaluate the individual design choices of our model in a controlled setup. We also show successful reconstructions of real images. Our work is currently limited to scenes with few objects and simple backgrounds. Future work will address the challenges of more complex scenes.

ACKNOWLEDGMENTS

This work has been supported by Cyber Valley, the Max Planck Society and Innosuisse funding (Grant No. 34475.1 IP-ICT). We are grateful to the Max Planck ETH Center for Learning Systems for supporting Cathrin Elich. We further thank Michael Strecke for his support with generating our ShapeNet dataset.

References

Achlioptas, P., Diamanti, O., Mitliagkas, I., and Guibas, L. Learning representations and generative models for 3D point clouds. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 40–49, 2018.

Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. Synthesizing robust adversarial examples. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 284–293, 2018.

Burgess, C. P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., and Lerchner, A. MONet: Unsupervised scene decomposition and representation, 2019.

Chang, A. X., Funkhouser, T. A., Guibas, L. J., Hanrahan, P., Huang, Q.-X., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. Shapenet: An information-rich 3d model repository. *ArXiv*, abs/1512.03012, 2015.

Chen, W., Gao, J., Ling, H., Smith, E., Lehtinen, J., Jacobson, A., and Fidler, S. Learning to predict 3D objects with an interpolation-based differentiable renderer. In *Advances In Neural Information Processing Systems (NeurIPS)*, 2019.

Chen, Z. and Zhang, H. Learning implicit fields for generative shape modeling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5939–5948, 2019. doi: 10.1109/CVPR.2019.00609.

Choy, C. B., Xu, D., Gwak, J., Chen, K., and Savarese, S. 3d-r2n2: A unified approach for single and multi-view 3D object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

Crawford, E. and Pineau, J. Spatially invariant unsupervised object detection with convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

Delaunoy, A. and Prados, E. Gradient flows for optimizing triangular mesh-based surfaces: Applications to 3D reconstruction problems dealing with visibility. *International Journal of Computer Vision (IJCV)*, 95:100–123, 11 2011. doi: 10.1007/s11263-010-0408-9.

Eigen, D., Puhrsch, C., and Fergus, R. 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, pp. 2366–2374, Cambridge, MA, USA, 2014. MIT Press.

Engelcke, M., Kosiorek, A. R., Jones, O. P., and Posner, I. GENESIS: Generative scene inference and sampling with object-centric latent representations. In *Accepted for International Conference on Learning Representations (ICLR)*, 2020.

Eslami, S. M. A., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., Kavukcuoglu, K., and Hinton, G. E. Attend, infer, repeat: Fast scene understanding with generative models. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 3233–3241, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Eslami, S. M. A., Jimenez Rezende, D., Besse, F., Viola, F., Morcos, A. S., Garnelo, M., Ruderman, A., Rusu, A. A., Danihelka, I., Gregor, K., Reichert, D. P., Buesing, L., Weber, T., Vinyals, O., Rosenbaum, D., Rabinowitz, N., King, H., Hillier, C., Botvinick, M., Wierstra, D., Kavukcuoglu, K., and Hassabis, D. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. doi: 10.1126/science.aar6170.

Everingham, M., Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88 (2):303–338, 2010. ISSN 0920-5691. doi: 10.1007/s11263-009-0275-4. URL https://doi.org/10.1007/s11263-009-0275-4.

Gadelha, M., Maji, S., and Wang, R. 3d shape induction from 2d views of multiple objects. In *2017 International Conference on 3D Vision (3DV)*, pp. 402–411, Oct 2017. doi: 10.1109/3DV.2017.00053.

Gkioxari, G., Malik, J., and Johnson, J. Mesh R-CNN. In *Proc. of International Conference on Computer Vision (ICCV)*, 2019.

Greff, K., Kaufman, R. L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., and Lerchner, A. Multi-object representation learning with iterative variational inference. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

Groueix, T., Fisher, M., Kim, V. G., Russell, B. C., and Aubry, M. AtlasNet: A papier-mâché approach to learning 3d surface generation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 216–224. IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00030.

Gwak, J., Choy, C. B., Chandraker, M., Garg, A., and Savarese, S. Weakly supervised 3d reconstruction with adversarial constraint. In *Int. Conf. on 3D Vision (3DV)*, 2017.

Henderson, P. and Ferrari, V. Learning single-image 3d reconstruction by generative modelling of shape, pose and shading. *International Journal of Computer Vision*, 2019. doi: 10.1007/s11263-019-01219-8. URL https://doi.org/10.1007/s11263-019-01219-8.

Henderson, P. and Lampert, C. H. Unsupervised object-centric video generation and decomposition in 3D. In *Advances in Neural Information Processing Systems (NeurIPS) 33*, 2020.

Hou, J., Dai, A., and Niessner, M. 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2019.

Jimenez Rezende, D., Eslami, S. M. A., Mohamed, S., Battaglia, P., Jaderberg, M., and Heess, N. Unsupervised learning of 3D structure from images. In *Advances in Neural Information Processing Systems 29*, pp. 4996–5004, 2016.

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., and Girshick, R. B. 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.

Kar, A., Häne, C., and Malik, J. Learning a multi-view stereo machine. In *Proc. of Advances in Neural Information Processing (NeurIPS)*, 2017.

Kato, H., Ushiku, Y., and Harada, T. Neural 3d mesh renderer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Kato, H., Beker, D., Morariu, M., Ando, T., Matsuoka, T., Kehl, W., and Gaidon, A. Differentiable rendering: A survey, 2020.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

Kniaz, V. A., Knyaz, V. V., Remondino, F., Bordodymov, A., and Moshkantsev, P. Image-to-voxel model translation for 3d scene reconstruction and segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer International Publishing, 2020.

Lerer, A., Gross, S., and Fergus, R. Learning physical intuition of block towers by example. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 430–438, New York, New York, USA, 20–22 Jun 2016. PMLR. URL http://proceedings.mlr.press/v48/lerer16.html.

Li, N., Eastwood, C., and Fisher, R. B. Learning object-centric representations of multi-object scenes from multiple views. In *Advances in Neural Information Processing Systems (NeurIPS) 33*, 2020.

Liao, Y., Schwarz, K., Mescheder, L., and Geiger, A. Towards unsupervised learning of generative models for 3d controllable image synthesis. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Lin, C., Kong, C., and Lucey, S. Learning efficient point cloud generation for dense 3d object reconstruction. In McIlraith, S. A. and Weinberger, K. Q. (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 7114–7121. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16530.

Lin, Z., Wu, Y.-F., Peri, S. V., Sun, W., Singh, G., Deng, F., Jiang, J., and Ahn, S. SPACE: Unsupervised object-oriented scene representation via spatial attention and decomposition. In *Accepted for International Conference on Learning Representations (ICLR)*, 2020.

Liu, S., Li, T., Chen, W., and Li, H. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems (NeurIPS) 33*, 2020.

Loper, M. M. and Black, M. J. OpenDR: An approximate differentiable renderer. In *European Conference on Computer Vision (ECCV)*, 2014.

Meka, A., Maximov, M., Zollhoefer, M., Chatterjee, A., Seidel, H.-P., Richardt, C., and Theobalt, C. LIME: Live intrinsic material estimation. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2018.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. Occupancy networks: Learning 3d reconstruction in function space. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Nguyen-Phuoc, T., Li, C., Balaban, S., and Yang, Y.-L. RenderNet: A deep convolutional network for differentiable rendering from 3D shapes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Nguyen-Phuoc, T., Richardt, C., Mai, L., Yang, Y.-L., and Mitra, N. Blockgan: Learning 3d object-aware scene representations from unlabelled images. In *Advances in Neural Information Processing Systems (NeurIPS) 33*, 2020.

Oechsle, M., Mescheder, L., Niemeyer, M., Strauss, T., and Geiger, A. Texture fields: Learning texture representations in function space. In *Proceedings IEEE International Conf. on Computer Vision (ICCV)*, 2019.

Park, J. J., Florence, P., Straub, J., Newcombe, R., and Lovegrove, S. Deepsdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Prabhudesai, M., Lal, S., Tung, H. F., Harley, A. W., Potdar, S., and Fragkiadaki, K. 3dq-nets: Visual concepts emerge in pose equivariant 3d quantized neural scene representations. In *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1567–1570. IEEE, 2020. doi: 10.1109/CVPRW50498.2020.00202. URL https://doi.org/10.1109/CVPRW50498.2020.00202.

Qi, C. R., Su, H., Nießner, M., Dai, A., Yan, M., and Guibas, L. Volumetric and multi-view cnns for object classification on 3d data. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Ramamoorthi, R. and Hanrahan, P. A signal-processing framework for inverse rendering. In *SIGGRAPH*, pp. 117–128, 2001.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016. doi: 10.1109/CVPR.2016.91.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, volume 28, pp. 91–99. Curran Associates, Inc., 2015.

Richardson, E., Sela, M., Or-El, R., and Kimmel, R. Learning detailed face reconstruction from a single image. In *Proceedings of the Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Shin, D., Fowlkes, C., and Hoiem, D. Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Sitzmann, V., Zollhöfer, M., and Wetzstein, G. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Tatarchenko, M., Dosovitskiy, A., and Brox, T. Multi-view 3d models from single images with a convolutional network. In *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 322–337, Cham, 2016. Springer International Publishing.

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

Tulsiani, S., Zhou, T., Efros, A., and Malik, J. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 209–217, 07 2017. doi: 10.1109/CVPR.2017.30.

Wang, R., Yang, N., Stueckler, J., and Cremers, D. 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.

Wang, Z. and Bovik, A. C. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: From error visibility to structural similarity. *Trans. Img. Proc.*, 13(4):600–612, April 2004. ISSN 1057-7149. doi: 10.1109/TIP.2003.819861. URL https://doi.org/10.1109/TIP.2003.819861.

Wu, J., Zhang, C., Xue, T., Freeman, W. T., and Tenenbaum, J. B. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 82–90, 2016.

Wu, J., Wang, Y., Xue, T., Sun, X., Freeman, W. T., and Tenenbaum, J. B. MarrNet: 3D shape reconstruction via 2.5D sketches. In *Advances In Neural Information Processing Systems (NeurIPS)*, 2017.

Xiang, Y., Schmidt, T., Narayanan, V., and Fox, D. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In *Robotics: Science and Systems (RSS)*, 2017.

Xie, H., Yao, H., Sun, X., Zhou, S., and Zhang, S. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *International Conference on Computer Vision (ICCV)*, pp. 2690–2698. IEEE, 2019. doi: 10.1109/ICCV.2019.00278. URL https://doi.org/10.1109/ICCV.2019.00278.

Xu, Q., Wang, W., Ceylan, D., Mech, R., and Neumann, U. DISN: deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 490–500, 2019.

Yan, X., Yang, J., Yumer, E., Guo, Y., and Lee, H. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2016.

Yang, Y., Chen, Y., and Soatto, S. 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*, pp. 6557–6566. IEEE, 2020. doi: 10.1109/CVPR42600.2020.00659. URL https://doi.org/10.1109/CVPR42600.2020.00659.

Yifan, W., Serena, F., Wu, S., Öztireli, C., and Sorkine-Hornung, O. Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH ASIA)*, 38(6), 2019.

Zhou, Y., Barnes, C., Lu, J., Yang, J., and Li, H. On the continuity of rotation representations in neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Zhu, J.-Y., Zhang, Z., Zhang, C., Wu, J., Torralba, A., Tenenbaum, J., and Freeman, B. Visual object networks: Image generation with disentangled 3d representations. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2018.

Supplementary Material

A. Overview

In this supplementary material, we present further evaluation results on various datasets. In particular, we present more qualitative results on the Clevr dataset (Johnson et al., 2017) (Fig. 10) as well as ShapeNet scenes (Chang et al., 2015) in Fig. 16, Fig. 17, and Fig. 18. Typical failure cases for ablations of our method are shown in Fig. 11. We further present more detailed results on our experiments with different object counts in Tab. 4, and Tab. 5, Tab. 6 and show qualitative results for these experiments in Fig. 14. Further information about our experiments on real-world images are provided in Sec. D with additional qualitative results shown in Fig. 12 and Fig. 13. In Fig. 15 and Fig. 19, we present results for latent traversals for both Clevr and ShapeNet scenes. Renderings of novel views on ShapeNet scenes are shown in Fig. 20. We provide rotation error histograms in Fig. 21. Detailed explanations are provided in the captions of the corresponding Figures and Tables.

Moreover, we also provide more information about the network architecture, additional auxiliary loss functions and the applied parameter settings in Sec. B. A more detailed listing of the reported metrics can be found in Sec. C.

B. Network Architecture & Parameters

Additional Loss Functions. For our experiments on the ShapeNet tabletop dataset, we use two additional loss functions:

- We favor poses which render the object visible in the image

$$L_p = \sum_i \max(-\min(x_i^p, w - x_i^p), 0) \ , \tag{6}$$

  where x_i^p is the pixel position of the object center and w is the image width.

- We penalize intersections between objects through

$$L_{int} = \sum_{i,j<i} \frac{1}{K} \sum_{k=1}^{K} \max(-(\phi_i(\mathbf{x}_k) + \phi_j(\mathbf{x}_k)), 0) \ , \tag{7}$$

  where i, j are object indices, $\mathbf{x}_k$ are sample points distributed evenly between the object centers and $\phi_i(\mathbf{x}_k) := \mathbf{\Phi}\left(\mathbf{z}_{i,sh}, \mathbf{T}_c^o(\mathbf{z}_{i,ext})\mathbf{x}_k\right)$.

Network Parameters. We provide detailed information about our network architecture and parameter settings in Tab. 3.

C. Evaluation Metrics

Instance Reconstruction. We evaluate the decomposition capability of our model by comparing the predicted object masks $\widehat{M}_{1:N}$ with the ground truth masks M_{gt}. For each object combination $(M_i, M_{gt,j})$, the IoU w.r.t. the occupied pixels is determined. We call object o_i to be a *true positive* if there is an object $o_{gt,j}$ for which $\text{IoU}(M_i, M_{gt,j}) \geq \tau$ for some threshold τ. All other predicted objects are considered as *false positives*. Ground truth objects that were not associated with a prediction are stated to be *false negatives*. As objects might not be viewable in the image due to occlusion, we only consider masks with a minimum number of 25 occupied pixels. For an image with object mask predictions and ground truth $(\widehat{M}_{1:N}, M_{gt})$, we denote the total number of true positives as $TP_\tau(\widehat{M}_{1:N}, M_{gt})$, the number of false positives as $FP_\tau(\widehat{M}_{1:N}, M_{gt})$, and the number of false negatives as $FN_\tau(\widehat{M}_{1:N}, M_{gt})$.

From this, we compute our reported metrics as follows. For a set of images $\mathcal{I}$ and $\mathcal{T} := \{0.5, 0.55, ..., 0.95\}$, we have

$$AP = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} AP_\tau, \tag{8}$$

$$AP_\tau = \frac{1}{|\mathcal{I}|} \sum_{(\widehat{M}_{1:N}, M_{gt})} Prec_\tau \tag{9}$$

$$= \frac{1}{|\mathcal{I}|} \sum_{(\widehat{M}_{1:N}, M_{gt})} \frac{TP_\tau}{TP_\tau + FP_\tau} \tag{10}$$

$$AR_\tau = \frac{1}{|\mathcal{I}|} \sum_{(\widehat{M}_{1:N}, M_{gt})} Rec_\tau \tag{11}$$

$$= \frac{1}{|\mathcal{I}|} \sum_{(\widehat{M}_{1:N}, M_{gt})} \frac{TP_\tau}{TP_\tau + FN_\tau} \tag{12}$$

$$F1_\tau = \frac{1}{|\mathcal{I}|} \sum_{(\widehat{M}_{1:N}, M_{gt})} 2\frac{Prec_\tau \cdot Rec_\tau}{Prec_\tau + Rec_\tau} \tag{13}$$

where

$$TP_\tau = TP_\tau(\widehat{M}_{1:N}, M_{gt}), \tag{14}$$

$$FP_\tau = FP_\tau(\widehat{M}_{1:N}, M_{gt}), \tag{15}$$

$$FN_\tau = FN_\tau(\widehat{M}_{1:N}, M_{gt}), \tag{16}$$

$$Prec_\tau = Prec_\tau(\widehat{M}_{1:N}, M_{gt}), \tag{17}$$

$$Rec_\tau = Rec_\tau(\widehat{M}_{1:N}, M_{gt}). \tag{18}$$

Image Reconstruction. We use the following metrics for

Table 3. **Network Parameters.** We report the parameter setting that was used for our experiments.
Notation: *: same as Clevr, _: latent vector is concatenated to input of this layer (see (Park et al., 2019))

			Clevr	ShapeNet
network architecture	object encoder	Conv	$[32, 32, 64, 64]$	*
		FC	$[256, 64]$	*
	shape decoder	FC	$[64, 64, \underline{64}, 64]$	*
	color decoder	FC	$[64, 64, 64, 64]$	*
	obj. repr.	D_{sh}	8	16
		D_{tex}	7	15
training setup	# epochs		500	400
	batch size		8	*
	learning rate		0.0001	*
	loss functions, weights	λ_I	1.0	*
		λ_D	0.1	0.05
		λ_{gr}	0.01	*
		λ_{sh}	lin(0.025-0.0025; 500K)	lin(0.1-0.01; 500K)
		(λ_{inter})	-	(0.001)
		(λ_{view})	-	(0.005)
data	image size		(64, 64)	*
	dataset size		(9K/ 1K/ 2.5K)	(18K/ 2K/ 5K)
	# objects		2, 3, 4, 5	3
	position range		$[1.5, 1.5]^2$	*
	size range		$[0.625, 1.25]$	*cars:* $[1.0, 1.5]$, *chairs:* $[0.75, 1.25]$, *tabletop:* $[0.8, 1.5]$

evaluating the reconstructed images:

$$MSE(\widehat{I}_{1:N}, I_{gt}) = \frac{1}{|\Omega|} \sum_{\mathbf{u} \in \Omega} \|\Delta I(\mathbf{u})\|^2 \tag{19}$$

$$RMSE(\widehat{I}_{1:N}, I_{gt}) = \sqrt{MSE(\widehat{I}_{1:N}, I_{gt})}, \tag{20}$$

$$PSNR(\widehat{I}_{1:N}, I_{gt}) = 10 \log_{10} \frac{L^2}{MSE(\widehat{I}_{1:N}, I_{gt})} \ , \tag{21}$$

with $\Delta I(\mathbf{u}) = \widehat{I}_{1:N}(\mathbf{u}) - I_{gt}(\mathbf{u})$ and L is the dynamic range of allowable image pixel intensities (Wang & Bovik, 2009). We refer the reader to (Wang et al., 2004) for a detailed explanation of the SSIM metric.

We use the scikit-image implementation[1] to compute PSNR and SSIM scores.

Depth Reconstruction. Our depth reconstruction evaluation is based on (Eigen et al., 2014) and is evaluated with the following measures:

$$RMSE(\widehat{D}_{1:N}, D_{gt}) = \sqrt{\frac{1}{|\Omega|} \sum_{\mathbf{u} \in \Omega} \|\Delta D(\mathbf{u})\|^2}, \tag{22}$$

$$AbsRD(\widehat{D}_{1:N}, D_{gt}) = \frac{1}{|\Omega|} \sum_{\mathbf{u} \in \Omega} \frac{|\Delta D(\mathbf{u})|}{D_{gt}(\mathbf{u})}, \tag{23}$$

$$SqRD(\widehat{D}_{1:N}, D_{gt}) = \frac{1}{|\Omega|} \sum_{\mathbf{u} \in \Omega} \frac{\|\Delta D(\mathbf{u})\|^2}{D_{gt}(\mathbf{u})} \tag{24}$$

with $\Delta D(\mathbf{u}) = \widehat{D}_{1:N}(\mathbf{u}) - D_{gt}(\mathbf{u})$.

Pose Estimation. We evaluate the error on the predicted pose only for objects that were denoted as *true positive*, i.e. for which we found a valid ground truth object match. Since we are missing the association between object masks and object poses in our data, we compare each predicted object's position $\mathbf{p}_i$ to the closest ground truth object ($\mathbf{p}_{gt,j}$) according to its 3D position. Each ground truth object is

[1] https://scikit-image.org/docs/dev/api/skimage.metrics.html

assigned at most once in a greedy proceeding.

$$Err_{pos} = \frac{1}{|P|} \sum_{(\mathbf{p}_i, \mathbf{p}_{gt,j}) \in P} \sqrt{\|\mathbf{p}_i - \mathbf{p}_{gt,j}\|^2}, \tag{25}$$

$$\text{with} \quad P = \{\text{found matches } (\mathbf{p}_i, \mathbf{p}_{gt,j})\},$$

$$Err_{rot} = \text{median}_{(\mathbf{p}_i, \mathbf{p}_{gt,j}) \in P} \left[\frac{360^\circ}{2\pi} \Delta r_{i,j} \right] \tag{26}$$

$$\text{with} \quad \Delta r_{i,j} = \min \Big(|r_i - r_{gt,j}|, 2\pi - |r_i - r_{gt,j}| \Big),$$

$$r_i = \text{arctan2}(z_{cos,i}, z_{sin,i}),$$

D. Evaluation on Real Data

We show evaluation results on real data from our model that was trained on synthetic data.

First, we directly evaluated our basic model that was trained on 3-object scenes on own photos of toy building blocks on a gray table (Fig. 12).

Second, we evaluated our method on images from (Lerer et al., 2016) (Fig. 13). As stacked objects do not occur in our standard datasets, we trained a new model on an appropriate variant of the Clevr dataset. Specifically, our new synthetic dataset contains only cubes which can be placed on top of each other. Moreover, the colors for the individual objects where chosen from {red, yellow, green, blue}and modified with gaussian noise ($\sigma = 0.05$). The size range was decreased to [0.5, 0.75]. We further adapted the camera pose to better align to the real images. Other properties are the same as for our standard Clevr dataset. During training, we applied further data randomization by adding uniform color noise sampled from a gaussian distribution ($\sigma = 0.025$). The used shape decoder was pre-trained on cubes only. For evaluation, we considered the last frame of each sequence containing up to three objects of the dataset from (Lerer et al., 2016). These images were cropped and down-scaled to obtain images with required size.

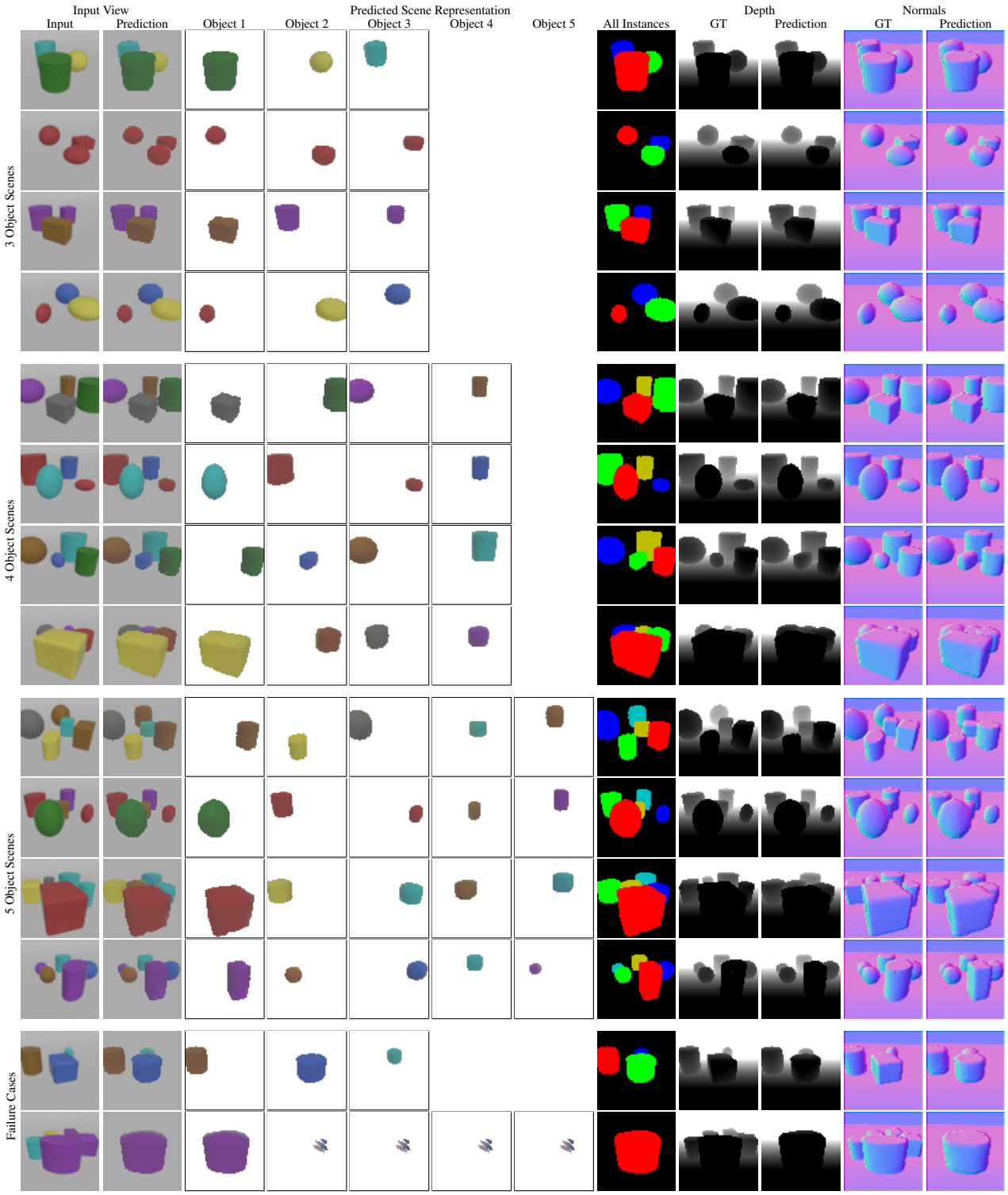

Figure 10. **Qualitative results on the Clevr dataset (Johnson et al., 2017) with three, four, and five objects.** Our model is able to decompose the scene into the individual objects. It recognizes basic color appearance and geometric properties like basic shape type and deformations (best seen in normal map). It is able to infer complete objects although some of them might be partly occluded by others in the input image. In the last two rows we also show failure cases: We found that our model sometimes misinterprets cubes as cylinders which is presumably due to the similarity of their shape and appearance at the image resolution. In few cases, it only detects a low number of objects, predominantly the most significant ones.

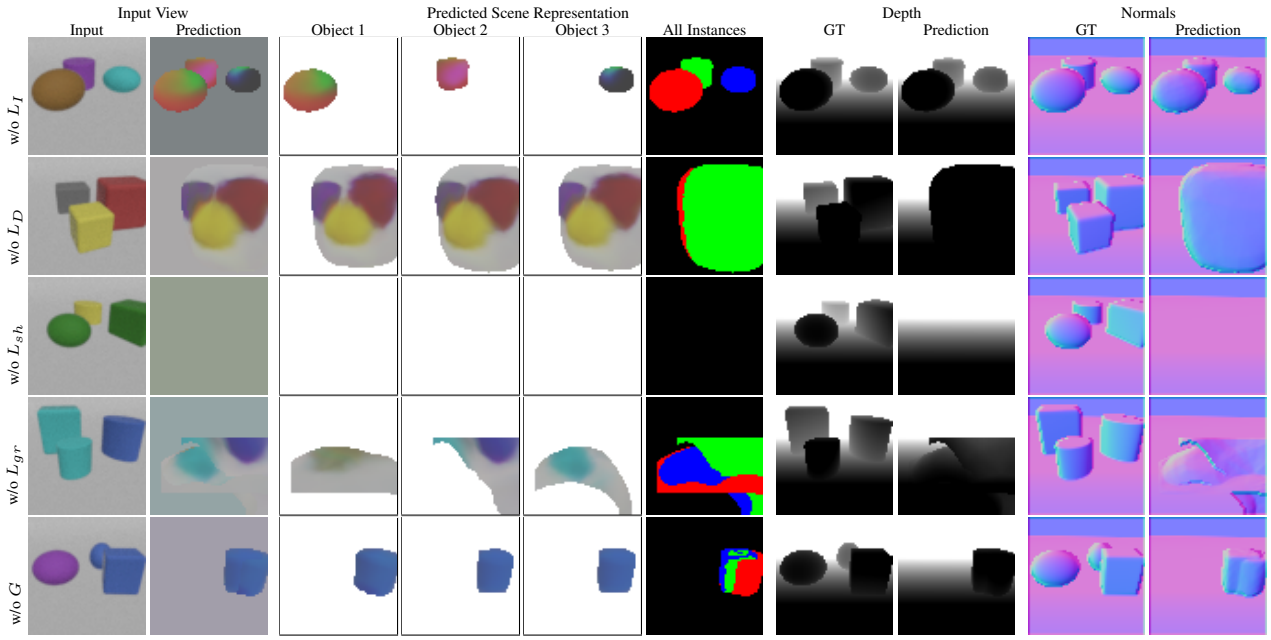

Figure 11. **Qualitative results for ablation study.** When training variants of our model without specific components, we observe typical failure cases. While a model trained without L_I is able to decompose the scene in the input image, it is obviously not able to recover the correct objects' appearance. L_D is crucial for learning the decomposition as otherwise the model can adapt the texture to obtain similar RGB reconstructions. If the shape is not regularized (L_{sh}) to match the pre-trained shape latent space, the model was not able to predict any reasonable object at all. L_{gr} helps to prevent the objects from being merged into the ground as well as to make sure that objects have a closed surface towards the ground. Without the Gaussian blur at the beginning of the training, the model often fails to detect the different objects but focuses on a single one instead.

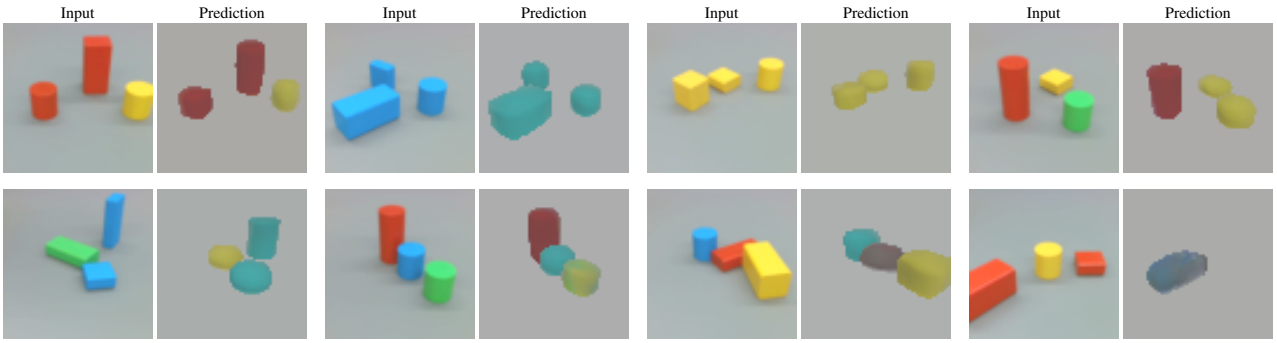

Figure 12. **Evaluation on real Clevr-like images.** We show results on real images by our model that was trained on the synthetic Clevr dataset. In several of the shown images our model can capture the coarse scene layout and shape properties of the objects. However, challenges arise due to domain, lighting, camera intrinsics and view point changes indicating interesting directions for future research.

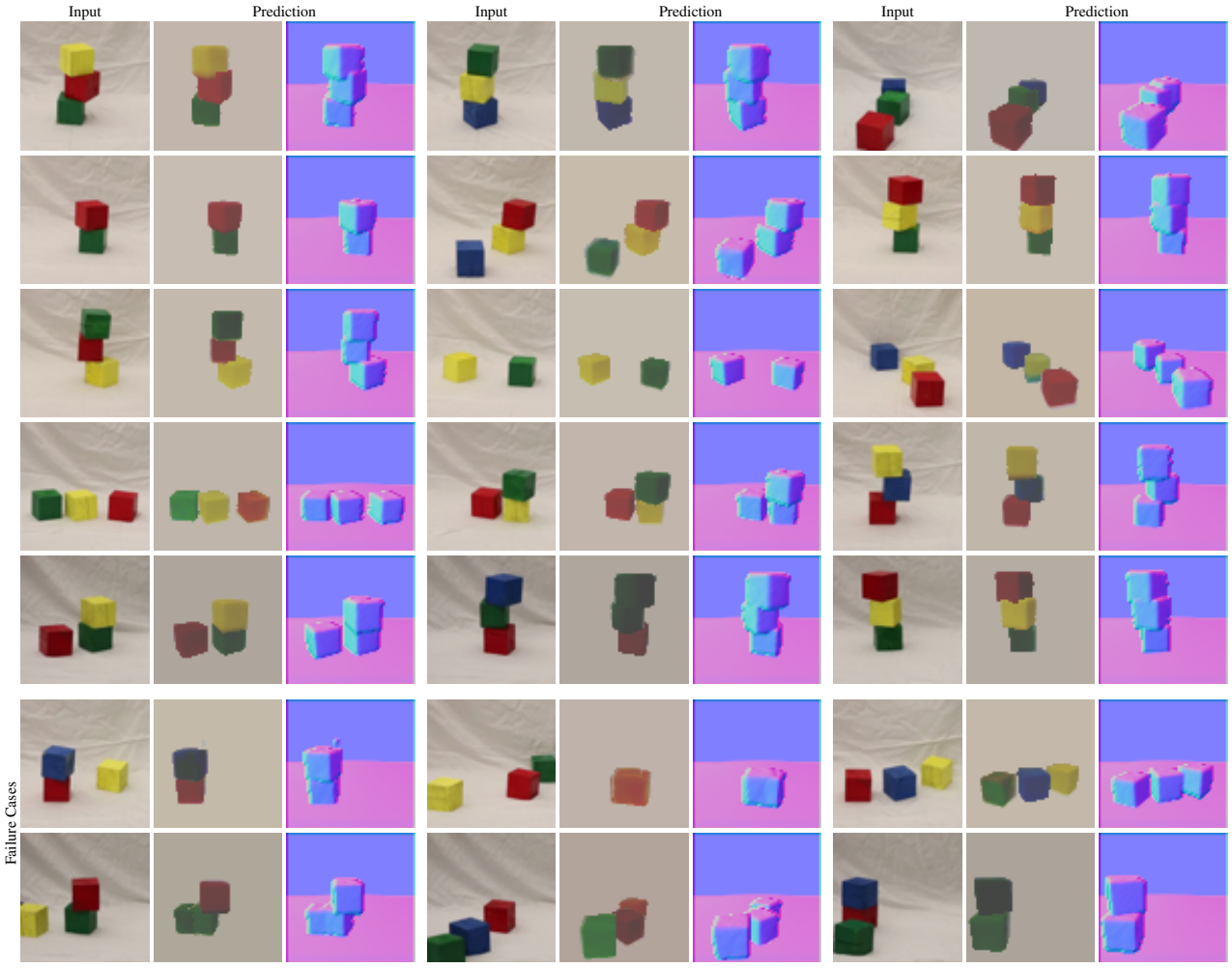

Figure 13. **Parsing real images of block towers (Lerer et al., 2016).** Our model infers reasonable scene decomposition for the most part. The configuration of the objects including their color and pose is well described. However, we observe that it sometimes misses single objects or uses an inaccurate coloring. Albeit our simple background model is not able to reconstruct the shades of the cloth in the background, it is robust against this unfamiliar structure.

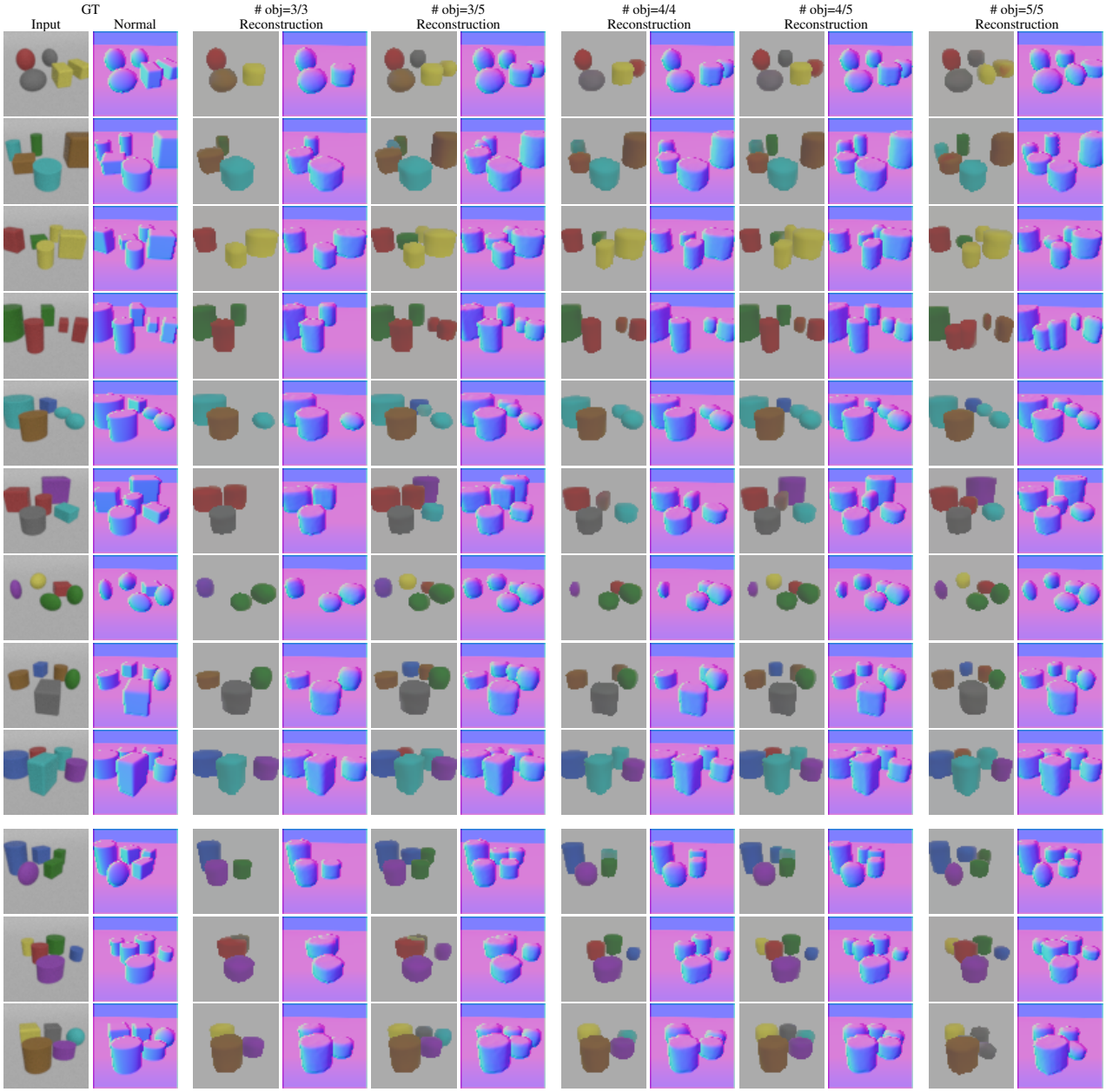

Figure 14. **Qualitative results on the Clevr dataset (Johnson et al., 2017) with varied number of objects.** As we use a shared encoder for detecting the objects in a recurrent architecture, it is possible to evaluate our model on a different number o_{test} of objects than it was trained on (o_{train}). For this, we reset the number of recurrent encoding steps to the number of objects in the test data. We show reconstruction results for varying numbers $\#obj = o_{train}/o_{test}$. Remarkably, our models that were trained only on either three or four objects are able to recognize larger number of objects.

Table 4. **Absolute scores on the Clevr dataset (Johnson et al., 2017) for scenes with varied number of objects ($\#obj = o_{train}/o_{test}$).** Experiments are ordered w.r.t. o_{train}. We use the encoder that was trained on o_{train} objects and adapt the number of slots o_{test} to the number of objects in the test set. Models achieve slightly better results when evaluated on scenes with a lower number of objects. If tested on scenes with larger number of objects, our model is able to detect more object than it has seen during training as can be seen from the $AR_{0.5}$ and allObj score.

	Instance Reconstruction					Image Reconstruction			Depth Reconstruction			Pose Est.
	mAP ↑	$AP_{0.5}$ ↑	$AR_{0.5}$ ↑	$F1_{0.5}$ ↑	allObj ↑	RMSE ↓	PSNR ↑	SSIM ↑	RMSE ↓	AbsRD ↓	SqRD ↓	Err_{pos}
# obj=2/2	0.782	0.977	0.963	0.967	0.928	0.039	28.389	0.941	0.432	0.012	0.040	0.138
# obj=2/3	0.606	0.877	0.842	0.854	0.622	0.060	24.827	0.884	0.671	0.027	0.085	0.214
# obj=2/4	0.406	0.698	0.629	0.655	0.186	0.083	21.972	0.81	0.906	0.049	0.149	0.293
# obj=2/5	0.294	0.583	0.474	0.516	0.031	0.095	20.714	0.769	0.921	0.056	0.151	0.312
# obj=3/2	0.756	0.974	0.969	0.97	0.942	0.041	28.011	0.937	0.452	0.013	0.044	0.14
# obj=3/3	0.712	0.949	0.942	0.943	0.85	0.049	26.466	0.914	0.554	0.019	0.061	0.155
# obj=3/4	0.613	0.883	0.853	0.863	0.512	0.06	24.669	0.88	0.665	0.028	0.083	0.179
# obj=3/5	0.478	0.775	0.71	0.735	0.212	0.072	23.093	0.841	0.69	0.033	0.086	0.201
# obj=4/2	0.720	0.969	0.959	0.961	0.923	0.044	27.39	0.929	0.484	0.015	0.051	0.146
# obj=4/3	0.708	0.953	0.943	0.945	0.852	0.05	26.252	0.911	0.564	0.020	0.064	0.153
# obj=4/4	0.688	0.941	0.919	0.926	0.746	0.054	25.632	0.899	0.584	0.022	0.064	0.151
# obj=4/5	0.575	0.869	0.81	0.832	0.397	0.063	24.258	0.869	0.600	0.026	0.067	0.165
# obj=5/2	0.606	0.919	0.913	0.914	0.845	0.053	25.959	0.908	0.582	0.021	0.075	0.174
# obj=5/3	0.628	0.914	0.908	0.908	0.778	0.057	25.181	0.892	0.657	0.026	0.091	0.168
# obj=5/4	0.640	0.916	0.899	0.903	0.691	0.058	24.950	0.885	0.649	0.027	0.082	0.161
# obj=5/5	0.604	0.895	0.861	0.872	0.539	0.061	24.568	0.876	0.593	0.025	0.067	0.149

Table 5. **Relative scores on the Clevr dataset (Johnson et al., 2017) for scenes with varied number of objects.** Experiments are ordered w.r.t. o_{train} and relative scores $(obj = n/m)$ / $(obj = n/n)$ are presented. Results are based on the same experiments as in Tab. 4.

	Instance Reconstruction					Image Reconstruction			Depth Reconstruction			Pose Est.
	mAP ↑	$AP_{0.5}$ ↑	$AR_{0.5}$ ↑	$F1_{0.5}$ ↑	allObj ↑	RMSE ↓	PSNR ↑	SSIM ↑	RMSE ↓	AbsRD ↓	SqRD ↓	Err_{pos}
# obj=2/3	0.775	0.898	0.874	0.883	0.670	1.538	0.875	0.939	1.553	2.250	2.125	1.551
# obj=2/4	0.519	0.714	0.653	0.677	0.200	2.128	0.774	0.861	2.097	4.083	3.725	2.123
# obj=2/5	0.376	0.597	0.492	0.534	0.033	2.436	0.730	0.817	2.132	4.667	3.775	2.261
# obj=3/2	1.062	1.026	1.029	1.029	1.108	0.837	1.058	1.025	0.816	0.684	0.721	0.903
# obj=3/4	0.861	0.930	0.906	0.915	0.602	1.224	0.932	0.963	1.200	1.474	1.361	1.155
# obj=3/5	0.671	0.817	0.754	0.779	0.249	1.469	0.873	0.920	1.245	1.737	1.410	1.297
# obj=4/2	1.047	1.030	1.044	1.038	1.237	0.815	1.069	1.033	0.829	0.682	0.797	0.967
# obj=4/3	1.029	1.013	1.026	1.021	1.142	0.926	1.024	1.013	0.966	0.909	1.000	1.013
# obj=4/5	0.836	0.923	0.881	0.898	0.532	1.167	0.946	0.967	1.027	1.182	1.047	1.093
# obj=5/2	1.003	1.027	1.060	1.048	1.568	0.869	1.057	1.037	0.981	0.840	1.119	1.168
# obj=5/3	1.040	1.021	1.055	1.041	1.443	0.934	1.025	1.018	1.108	1.040	1.358	1.128
# obj=5/4	1.059	1.023	1.044	1.036	1.282	0.951	1.016	1.010	1.094	1.080	1.224	1.081

Table 6. **Relative scores on the Clevr dataset (Johnson et al., 2017) for scenes with varied number of objects.** Experiments are ordered w.r.t. o_{test} and relative scores $(obj = n/m)$ / $(obj = m/m)$ are presented. Results are based on the same experiments as in Tab. 4.

	Instance Reconstruction					Image Reconstruction			Depth Reconstruction			Pose Est.
	mAP ↑	$AP_{0.5}$ ↑	$AR_{0.5}$ ↑	$F1_{0.5}$ ↑	allObj ↑	RMSE ↓	PSNR ↑	SSIM ↑	RMSE ↓	AbsRD ↓	SqRD ↓	Err_{pos}
# obj=3/2	0.967	0.997	1.006	1.003	1.015	1.051	0.987	0.996	1.046	1.083	1.100	1.014
# obj=4/2	0.921	0.992	0.996	0.994	0.995	1.128	0.965	0.987	1.120	1.250	1.275	1.058
# obj=5/2	0.775	0.941	0.948	0.945	0.911	1.359	0.914	0.965	1.347	1.750	1.875	1.261
# obj=2/3	0.851	0.924	0.894	0.906	0.732	1.224	0.938	0.967	1.211	1.421	1.393	1.380
# obj=4/3	0.994	1.004	1.001	1.002	1.002	1.020	0.992	0.997	1.018	1.052	1.049	0.987
# obj=5/3	0.882	0.963	0.964	0.963	0.915	1.163	0.951	0.976	1.186	1.368	1.492	1.084
# obj=2/4	0.590	0.742	0.684	0.707	0.249	1.537	0.857	0.901	1.551	2.227	2.328	1.940
# obj=3/4	0.891	0.938	0.928	0.932	0.686	1.111	0.962	0.979	1.139	1.272	1.297	1.185
# obj=5/4	0.930	0.973	0.978	0.975	0.926	1.074	0.973	0.984	1.111	1.227	1.281	1.066
# obj=2/5	0.487	0.651	0.550	0.592	0.058	1.557	0.843	0.878	1.553	2.240	2.254	2.094
# obj=3/5	0.791	0.866	0.825	0.843	0.393	1.180	0.940	0.960	1.164	1.320	1.284	1.349
# obj=4/5	0.952	0.971	0.941	0.954	0.737	1.033	0.987	0.992	1.012	1.040	1.000	1.107

Figure 15. **Latent traversal on the Clevr dataset (Johnson et al., 2017).** We linearly adapt the first object's shape (top) or texture (middle) latent to match each of the other objects' respective representation. Moreover, we move the first object within the scene (bottom). As we reason about objects in 3D, we are able to recognize intersections between objects and exclude invalid scenes (missing images in last row). By doing so, we are able to generate new plausible scenes. Object shapes are best seen in normal maps.

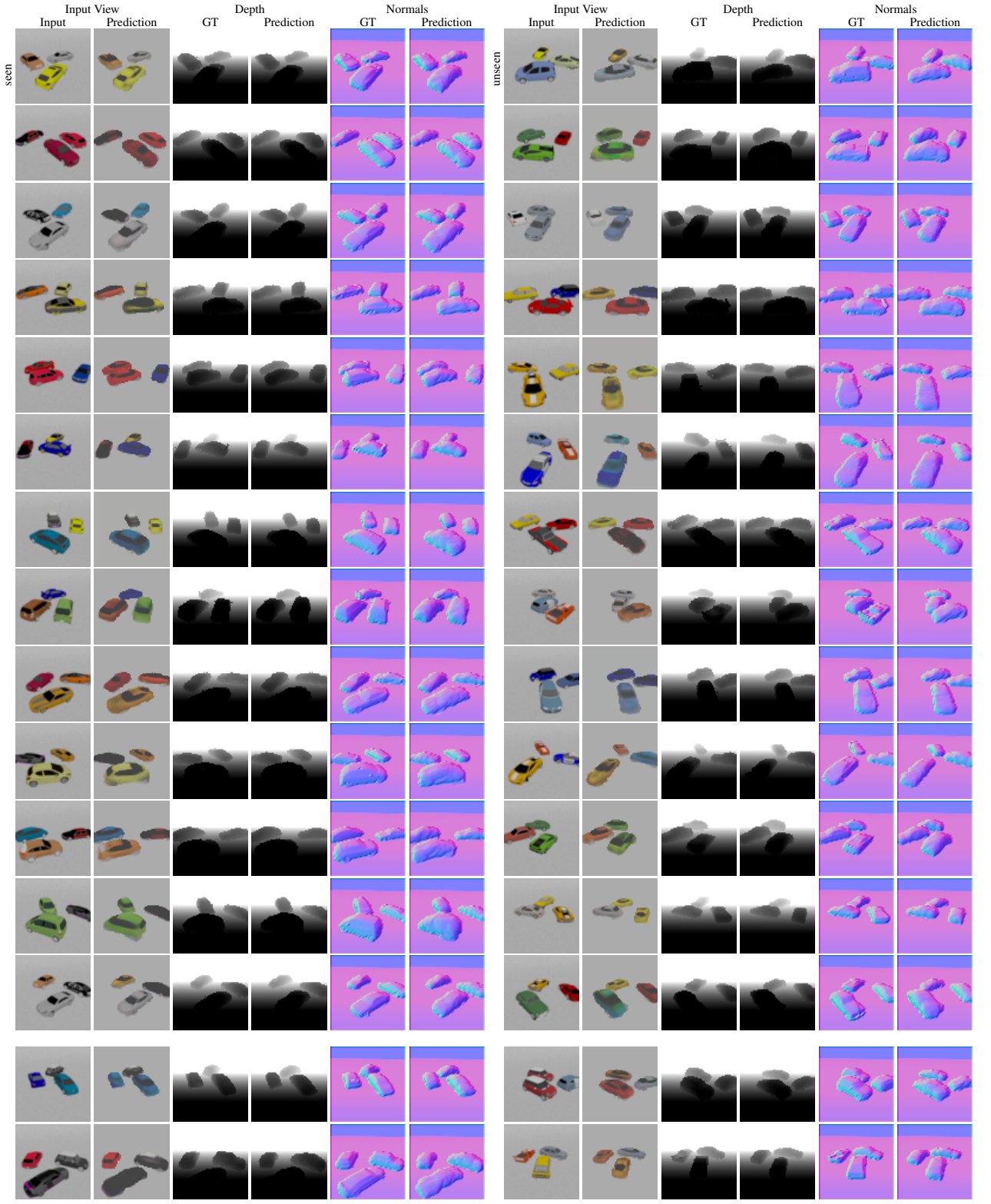

Figure 16. **Qualitative results on ShapeNet datasets (Chang et al., 2015) with car models.** Our model generates reasonable reconstruction for scenes with both seen and unseen object instances. For the latter case, it describes objects with similar shapes and textures is has seen in training. Typical failure cases are related to a pseudo-180-degree symmetry of the cars that is not distinguished by the model but handled by adapting the texture. In the lower two rows, all cars face in the wrong direction. This is in most cases not obvious from the reconstructed images only.

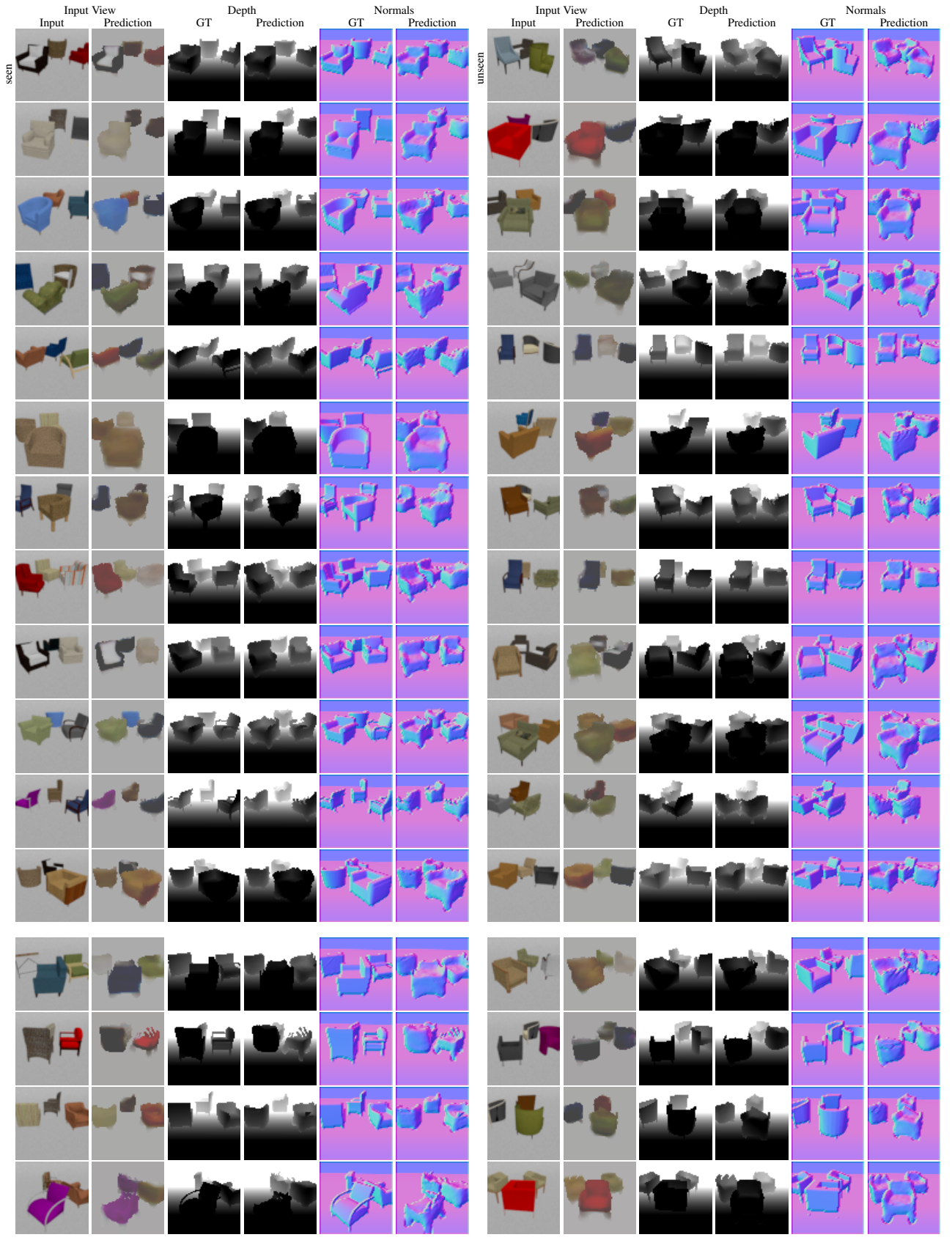

Figure 17. **Qualitative results on ShapeNet datasets (Chang et al., 2015) with chair models.** For the chair models it is more important to predict the correct rotation to infer a well matching shape than for other models in our datasets. The model still got easily trapped in local minima of 90-degree rotation 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.

Figure 18. **Qualitative results on tabletop scenes with ShapeNet (Chang et al., 2015) models.** For our mixed dataset, our model needs to predict object shapes from three different categories (mugs, bottles, cans) as well as respective typical size ranges. We found that our model is able to distinguish between the objects based on their typical characteristics. Unseen objects in the second test set are typically replaced by known objects from the training set which are similar in appearance. Handles of cups as well as thin, long bottlenecks are often neglected by the model. Especially for small objects, the model sometimes misses to reconstruct an object in the scene. The last row shows reconstructions from a failed training run in which only one object can be found.

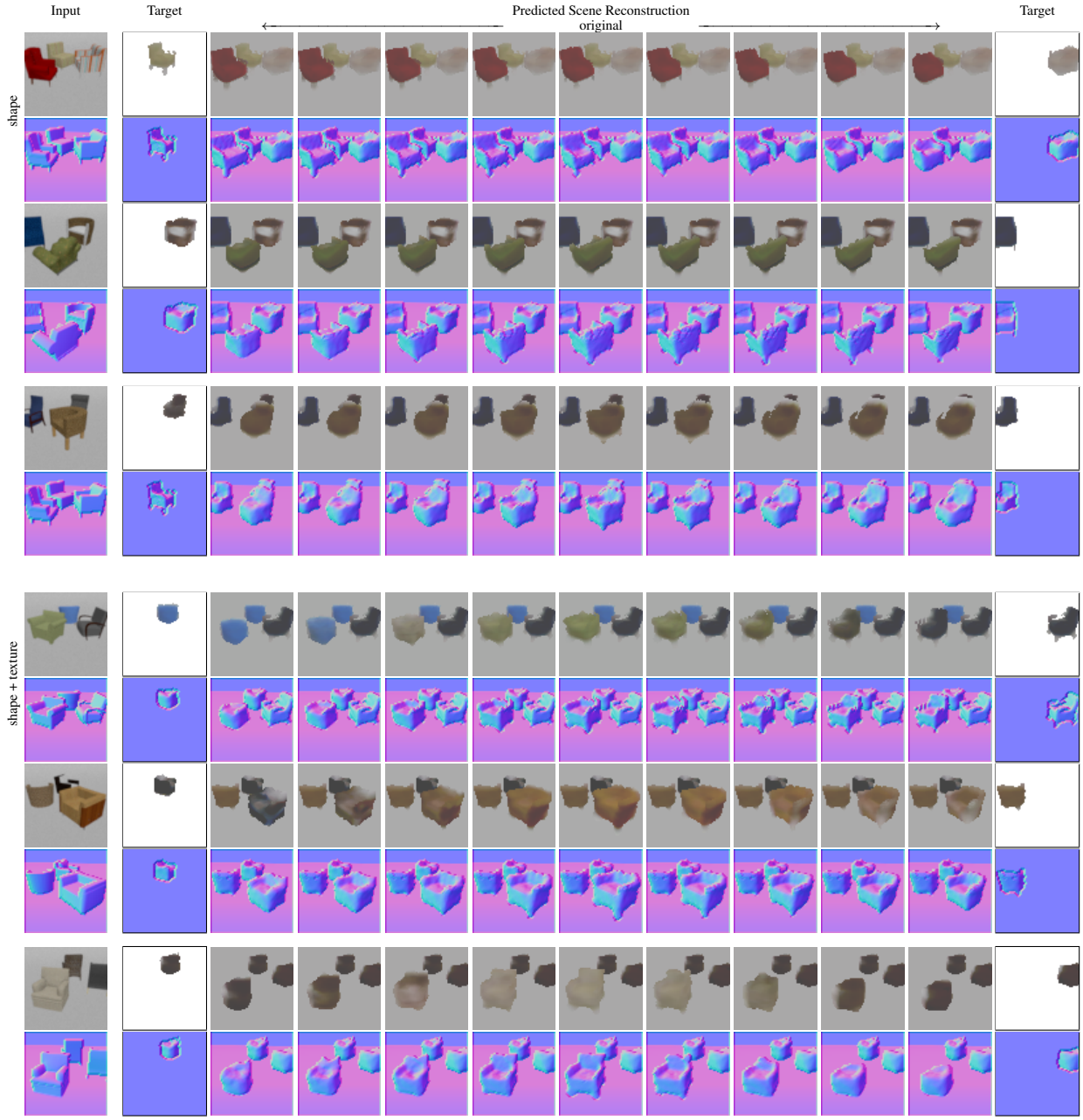

Figure 19. **Latent traversal on ShapeNet datasets (Chang et al., 2015) with chair models.** We linearly adapt the first object's latent to match each of the other objects' respective representation in either shape alone (top rows) or shape and texture (bottom rows). By this, we are able to generate new plausible scenes. Object shapes are best seen in the normal maps.

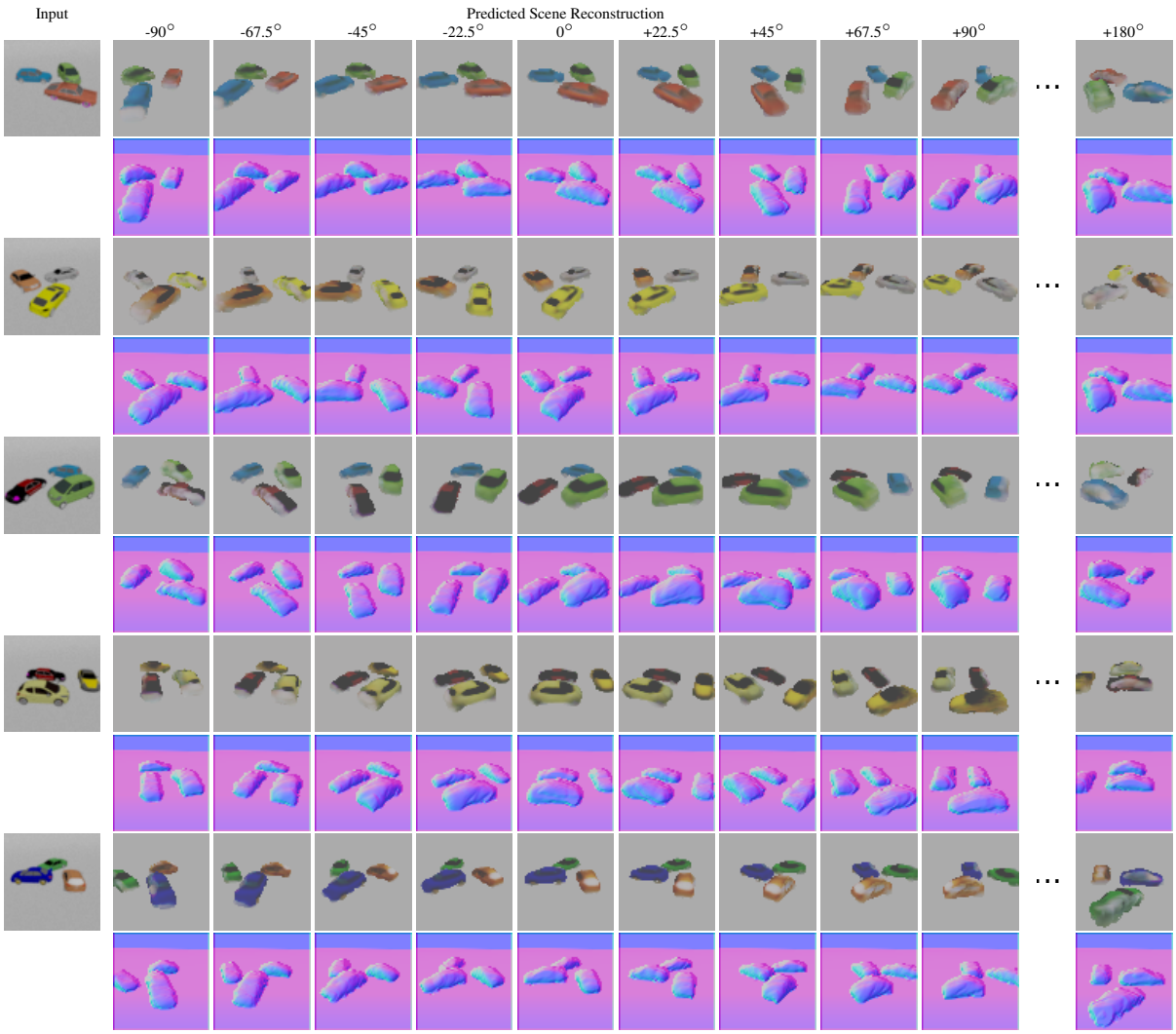

Figure 20. **Rendering of novel views.** As our model reasons about the underlying 3D structure of a given image, it is able to render novel views of a scene. This is possible although our model was trained exclusivity from single images. The reconstructed normal maps show that the model learned to reason about the depicted objects in 3D space. It can be observed that our model renders the reverse side of the car objects less accurate than the visible parts. This might be due to limited range in rotation that the model infers due to pseudo-symmetry.

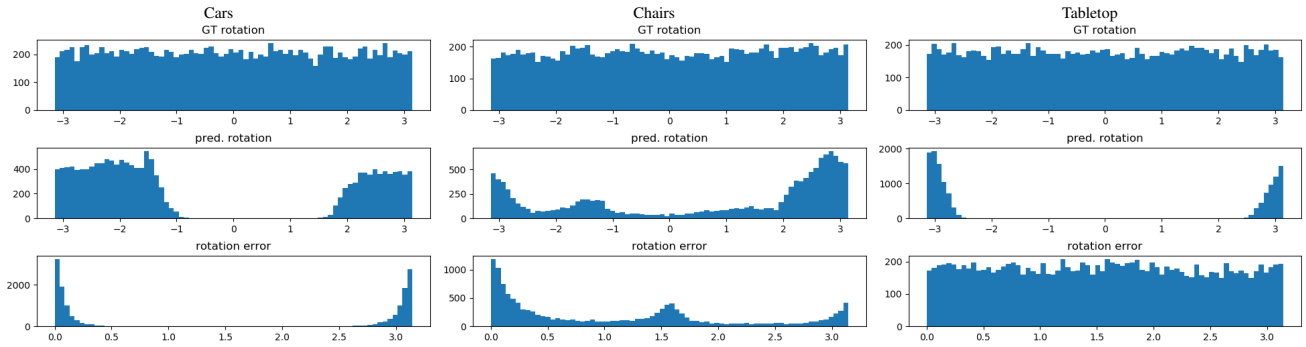

Figure 21. **Rotation Prediction on ShapeNet (Chang et al., 2015).** From top to bottom: GT and predicted rotation angles for each dataset and resulting rotation angles. While values for GT rotation are naturally uniformly distributed over the entire range of $[-\pi, \pi]$ for all scenes, we found that predicted rotation estimates can be 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 minima where it predicts a rotation up to a pseudo-symmetry. In contrast, it predicts rotation almost uniformly for the tabletop scenes due to the rotational symmetry of the shapes and the capability of adapting the texture.