Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5745–5753, 2019.

A Experimental Details

A.1 Dataset Details

Shapenet Chang et al. (2015): For training on Shapenet, we follow Deng et al. (2021) and extract 500,000 SDF values on the object's surface and 500,000 randomly sampled in a cube of side length 2. We also render a single RGB-D image with a camera sampled on the viewing hemisphere for testing according to Duggal & Pathak (2022). We use the same train/test split for each category as DIF-Net (Deng et al., 2021), but only keep the intersection of objects in both ours and the baselines test sets.

Pascal3D+ Xiang et al. (2014): Pascal3D+ is a real-world dataset containing camera images. The dataset provides object silhouettes, camera poses, and the CAD models used to annotate the camera poses. Since the same set of CAD models is used to annotate both the training and testing set, the dataset is considered to possess a bias (Tulsiani et al., 2017). However, unlike the other baselines, our method is only trained on Shapenet and therefore has not seen any of the CAD models from Pascal3D+ during training. We evaluate our method on the car, chair, and airplane category of Pascal3D+. We follow the testing split that is used by SDF-SRN (Lin et al., 2020). To generate partial point clouds in the camera frame as the input for our method, we first transform the CAD models into the camera frame using the ground truth camera poses. We remove points that are invisible from the camera origin. We do not have access to either ground truth poses or the complete CAD model during training and testing.

Pix3D Lim et al. (2013): Like Pascal3D+, Pix3D contains real-world 2D images annotated with 3D CAD models. Unlike Pascal3D+, Pix3D uses a variety of CAD models that align better with the images. We randomly select 200 images from the Pix3D chair dataset as the test set. We follow the same process as Pascal3D+ to generate the partial point clouds in the camera frame. Again, our method does not access the complete CAD models, and ground truth camera poses.

DDAD Guizilini et al. (2020): DDAD is an autonomous driving benchmark containing diverse urban scenes captured using car sensors. The dataset includes RGB videos captured using six cameras covering the 360-degree surrounding of the vehicle. DDAD also provides depth data across an entire 360-degree field of view scanned using long-range LiDAR sensors. We test our method on one scene with 100 frames. We extract only the frames that contain other cars in the camera's field of view. To generate the partial point clouds of the observed cars, we crop the LiDAR data with the ground truth poses and the masked images. The cropped LiDAR scans in the camera frame serve as our method's input. The DDAD dataset does not provide ground truth models of the observed cars, and our method does not have access to the ground truth camera poses. We therefore show only qualitative results.

A.2 Implementation Details

DIF-Net Architecture DIF-Net uses a SIREN (Sitzmann et al., 2020) network as the MLP backbone for both, the deformation and template networks. The Hyper-network is a ReLU network, where each MLP predicts the weights of one of the layers in the deformation network D.

Training Details We initialize the latent codes to small values from $\mathcal{N}(0, 0.01)$. Each model is optimized for 60 epochs and during each epoch, the model has access to 200,000 free and 200,000 surface points. We set the weights for the SDF loss $\mathcal{L}_{sdf}$ to 3e3, 1e2, 5e1 and 5e2 according to Sitzmann et al. (2020). We follow Deng et al. (2021) in choosing the weighting parameters as follows. λ_1 and λ_4 are 1e2 and 1e6 for all categories. $lambda_2$ is 5, 2, 5 for *car, plane and chair*. λ_3 is 1e2, 1e2 and 5e1 for each of the above categories.

Inference Details We jointly optimize the object pose and shape using the Adam optmizer with a learning rate of 0.001 for the shape and 0.01 for the pose. We optimize each shape for a total of 30 iterations, taking roughly 4 seconds.

Equi-pose Li et al. (2021): In this paper, we directly apply Equi-pose as an off-the-shelf pose estimation module. Since our method does not require accurate camera poses but rough initialization, we use the model weights trained on ModelNet40 (Wu et al., 2015) provided by the authors for all the experiments in this paper.

A.3 BASELINES

SDF-SRN Lin et al. (2020): We directly use the source code and pre-trained weights for Pascal3D+ and Shapenet provided by the original authors. In this paper, we follow SDF-SRN's test split for Pascal3D+ dataset. As for Shapenet, we take the union between our test set and SDF-SRN's test set as the Shapenet test set.

TARS-3D Duggal & Pathak (2022): We use the original implementation of TARS-3D from the authors. Since TARS-3D follows the same dataset setup as SDF-SRN, we use the network weights trained on Shapenet provided by the authors.

PoinTr Yu et al. (2021): We directly adopt the original code and network weights trained on Shapenet provided by the authors. For fair comparison, we transform the input partial point clouds into their coordinate frame with both ground truth and estimated camera poses. Furthermore, we follow their original training setup where the input point clouds are downsampled to 2048 points using farthest point sampling, and predict 8192 points as the complete point cloud.

B ADDITIONAL RESULTS

B.1 EXTRACTED TEMPLATE SHAPES

Here we show the extracted template shapes from our pretrained 3D prior network. We extract the zero-level set of the neural field using marching cubes. We can observe that the template represents shapes close to the "mean" object for cars and chairs. For planes however, the neural field does not represent an object, but fuses common aspects of different shapes together.

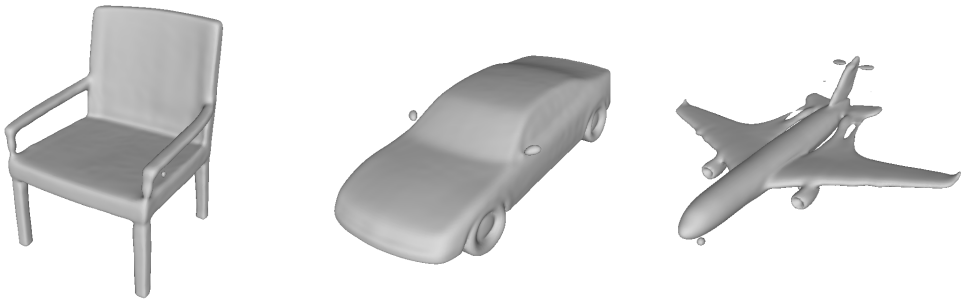

Figure 6: Extracted learned template shapes

B.2 PASCAL3D+ RESULTS

Figure 7: Additional results on the Pascal3D+ dataset

B.3 PIX3D RESULTS

Figure 8: Additional results on the Pix3D dataset