Supplementary Materials for Dynamic Plane Convolutional Occupancy Networks

Stefan Lionar[1*] Daniil Emtsev[1*] Dusan Svilarkovic[1*] Songyou Peng[1,2]
[1]ETH Zurich [2] Max Planck ETH Center for Learning Systems
{slionar, demtsev, dsvilarko}@ethz.ch songyou.peng@inf.ethz.ch

In this supplementary document, we first give the detailed information of our architectures in Section 1. We then discuss the convergence of our models under different configurations in Section 2. In Section 3, we provide the details of our ablation study evaluating the performance of Convolutional Occupancy Networks (ConvONet) [5] with pre-defined 5 and 7 static planes. In Section 4, we present per-category quantitative results and more qualitative results on ShapeNet [1] dataset.

1. Network Architectures

In this section, we provide additional details of our network architectures.

Encoder: The encoder of our networks is composed of a point cloud encoder and a plane predictor network. The architecture of our encoder is illustrated in Fig. 1

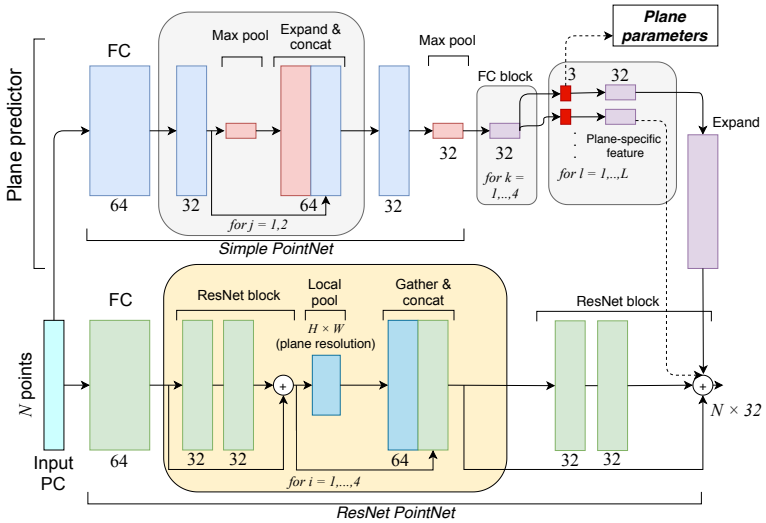

Figure 1. **Encoder architecture.** Our encoder is composed of a point cloud encoder (ResNet PointNet) and a plane predictor network.

- **Point Cloud Encoder:** We use the ResNet PointNet variant of [5], in which the pooling operation is performed locally. The max-pooling operation is applied only over the features falling onto the same grid. We use 5 ResNet blocks to obtain the per-point feature, as described in the supplementary of [5].

- **Plane Predictor Network:** The plane predictor network predicts the plane parameters of L dynamic planes by firstly learning the global context of the input point clouds using the simple variant of PointNet network [6]. The information

*Equal contribution. This is a 3D Vision course project at ETH Zurich.

of N point clouds is encoded into one global feature and then passed into a sequence of fully-connected layer (FC) blocks. We use 4 FC blocks and set a hidden dimension of 32 for the plane predictor network.

To predict the plane parameters, we pass the output from the FC blocks to L shallow networks consisting of 3 hidden dimensions, where the three numbers are the predicted plane parameters. The plane parameter predictions are subsequently passed through L fully-connected networks with one layer and a hidden dimension D, which is the same as the point cloud encoder's hidden dimension to obtain the plane-specific features. Each plane-specific feature is then expanded from $1 \times D$ to $N \times D$, and added to the output of point cloud encoder individually before processing into U-Net.

U-Net: We use a U-Net [7] to process the plane features and adapt a modified implementation from [5]. We set the input and output feature dimensions to 32 and choose the depth of the U-Net such that the receptive field is equal to the size of the feature plane. In doing so, we set a depth of 4 for our experiments with ShapeNet dataset (64^2 grids) and a depth of 5 for our scene experiments (128^2 grids).

Decoder: We use the decoder networks of [4] with 5 ResNet blocks and a hidden dimension of 32.

2. Convergence

In this section, we present the effect of positional encoding and the higher number of dynamic planes on the training convergence.

Positional encoding: We use the convolutional occupancy networks implementation of [5] and compare their validation IoU progression with and without positional encoding. As seen in Fig. 2, we observe consistency that the models trained with positional encoding converge faster. It applies in both the object-level experiment using ShapeNet dataset and the scene-level experiment using the synthetic indoor scene dataset.

Higher number of dynamic planes: Fixing the use of positional encoding, we also compare the convergence of our models with 3, 5, and 7 dynamic planes, as illustrated in Fig. 2. The faster convergence is pronounced in the scene experiment when increasing the number of dynamic planes from 3 to 5.

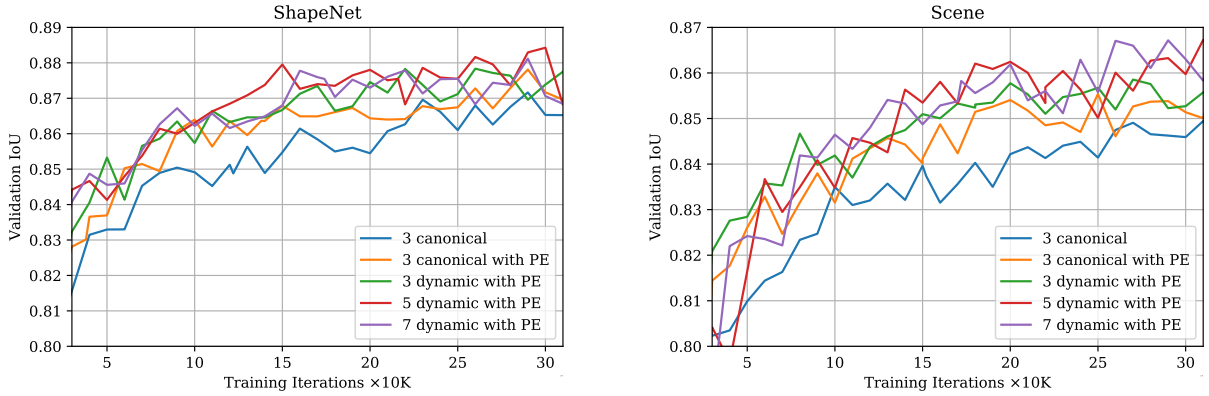

Figure 2. **Training progression**. The progression of validation IoU showing the effect of positional encoding and the higher number of dynamic planes on the training convergence. *Left:* ShapeNet. *Right:* Scene dataset.

3. Ablation study

In this section, we test the performance of ConvONet [5] with pre-defined 5 and 7 static planes to verify the effectiveness of our method. On top of the three canonical planes, we experiment with additional arbitrarily chosen sets of planes and flipping sets of planes following the prediction of our networks. We use the planar projection detailed in Section 3 of the main paper to project the per-point features to the 2D static planes. For the experiments with seven planes, we select the seventh plane normal rather arbitrarily as $(1, 1, 1)$. All models are trained until at least 900,000 iterations.

The results are shown in Table 1. When arbitrary planes are chosen, the performance is worse than following the prediction from our networks to include flipping sets of normals. We also see that ConvONet has limited capacity in capturing richer information from increasing the number of static planes. As we can see in Table 1 and the object-level results in Section 4 of the main paper, increasing the number of static planes mostly does not improve and even degrade the performance compared to using three canonical planes. Moreover, the results from our method are consistently superior, given the same number of planes. This ablation study further verifies the benefit of our method in jointly learning dynamic planes to predict the best planes for reconstruction and plane-specific features to increase the model capacity.

Without positional encoding										
Pre-defined plane normal							*IoU*	*Chamfer-L_1*	*Normal C.*	*F-score*
Canonical planes			Additional arbitrary set							
$(1,0,0)$	$(0,1,0)$	$(0,0,1)$	$(1,0,1)$	$(0,1,1)$	-	-	0.878	0.047	0.936	0.936
			$(1,0,1)$	$(0,1,1)$	$(1,1,1)$	$(-1,1,1)$	0.880	0.045	0.937	0.941
			Additional flipping set							
			$(-1,0,0)$	$(0,-1,0)$	-	-	0.880	0.046	0.937	0.939
			$(-1,0,0)$	$(0,-1,0)$	$(0,0,-1)$	$(1,1,1)$	0.883	0.045	0.937	0.941
With positional encoding										
Pre-defined plane normal							*IoU*	*Chamfer-L_1*	*Normal C.*	*F-score*
Canonical planes			Additional arbitrary set							
$(1,0,0)$	$(0,1,0)$	$(0,0,1)$	$(1,0,1)$	$(0,1,1)$	-	-	0.885	0.045	0.939	0.942
			$(1,0,1)$	$(0,1,1)$	$(1,1,1)$	$(-1,1,1)$	0.885	0.045	0.938	0.941
			Additional flipping set							
			$(-1,0,0)$	$(0,-1,0)$	-	-	0.890	0.044	0.940	0.945
			$(-1,0,0)$	$(0,-1,0)$	$(0,0,-1)$	$(1,1,1)$	0.889	0.043	0.939	0.946

Table 1. **Ablation study results.** We compare the performance of ConvONet [5] with pre-defined 5 and 7 static planes. When arbitrary planes are chosen, the performance is worse than following the prediction from our networks to include flipping sets of normals.

4. 3D Reconstruction on ShapeNet

In this section, we provide the per-category quantitative and additional qualitative results on the ShapeNet subset of Choy *et al*. [2].

Observations: Table 2 shows a quantitative comparison between ConvONet [5] and ours. Our models consistently outperform ConvONet [5] in all metrics. The most considerable improvement is observed from the challenging class *lamp*. Moreover, we see progressive improvement using a higher number of dynamic planes for our models.

The qualitative comparison is presented in Fig. 3, where we show the reconstruction of objects with intricate structures. Qualitatively, our models are able to preserve geometric details better compared to ConvONet [5]. The use of higher dynamic planes is also shown to improve reconstruction accuracy, especially in capturing intricate geometrical details, such as thin components and holes. For example, in the first row of Fig. 3, only our models with 5 and 7 dynamic planes are seen to reconstruct the phone's antenna accurately.

	IoU					*Chamfer-L_1*				
	Without PE	With PE				Without PE	With PE			
Category	ConvONet (3C)	ConvOnet (3C)	Ours (3D)	Ours (5D)	Ours (7D)	ConvONet (3C)	ConvOnet (3C)	Ours (3D)	Ours (5D)	Ours (7D)
airplane	0.847	0.859	0.865	0.862	**0.866**	0.034	0.032	**0.031**	**0.031**	**0.031**
bench	0.834	0.837	0.845	0.845	**0.847**	0.035	0.035	0.034	0.034	**0.033**
cabinet	0.938	0.942	0.941	0.942	**0.943**	0.048	0.046	0.047	0.047	**0.045**
car	0.887	0.889	0.891	**0.894**	**0.894**	0.073	0.073	0.072	0.071	**0.069**
chair	0.872	0.873	0.878	0.882	**0.883**	0.047	0.045	0.045	**0.044**	**0.044**
display	0.928	0.931	0.931	0.933	**0.934**	0.037	0.036	0.036	0.036	**0.035**
lamp	0.779	0.789	0.800	0.806	**0.807**	0.060	0.059	0.057	**0.054**	0.055
loudspeaker	0.914	0.919	**0.921**	**0.921**	**0.921**	0.065	0.062	0.063	0.063	**0.061**
rifle	0.844	0.853	0.858	**0.859**	**0.859**	0.029	0.027	**0.026**	**0.026**	**0.026**
sofa	0.937	0.939	0.941	0.941	**0.942**	0.042	0.041	0.041	0.041	**0.040**
table	0.888	0.893	**0.898**	**0.898**	**0.898**	0.040	0.039	**0.038**	**0.038**	**0.038**
telephone	0.954	0.956	0.955	**0.956**	**0.956**	0.028	0.027	0.028	**0.027**	**0.027**
vessel	0.867	0.873	0.877	0.879	**0.883**	0.043	0.041	0.041	0.040	**0.038**
mean	0.884	0.889	0.892	0.894	**0.895**	0.045	0.043	0.043	**0.042**	**0.042**
	Normal Consistency					*F-Score*				
airplane	0.930	0.932	**0.934**	**0.934**	**0.934**	0.966	0.970	0.972	**0.973**	**0.973**
bench	0.920	0.922	**0.924**	**0.924**	**0.924**	0.966	0.968	0.970	0.971	**0.972**
cabinet	0.955	0.956	0.957	**0.958**	0.957	0.954	0.957	0.957	0.957	**0.959**
car	0.892	0.892	0.894	**0.896**	**0.896**	0.856	0.857	0.857	0.864	**0.867**
chair	0.942	0.942	0.944	0.945	**0.946**	0.939	0.939	0.943	0.948	**0.950**
display	0.968	0.968	0.969	**0.970**	**0.970**	0.972	0.974	0.975	0.976	**0.978**
lamp	0.899	0.901	0.905	**0.908**	0.906	0.890	0.895	0.904	0.910	**0.911**
loudspeaker	0.935	0.939	0.939	**0.940**	0.938	0.888	0.896	0.894	**0.897**	**0.897**
rifle	0.931	0.929	0.934	**0.933**	0.931	0.980	0.981	0.983	0.983	**0.986**
sofa	0.957	0.958	**0.960**	0.959	**0.960**	0.953	0.955	0.957	0.958	**0.959**
table	0.958	0.959	**0.961**	**0.961**	**0.961**	0.966	0.969	0.972	**0.973**	**0.973**
telephone	0.982	0.982	**0.983**	**0.983**	**0.983**	0.988	0.988	0.989	**0.990**	**0.990**
vessel	0.919	0.919	0.923	**0.924**	**0.924**	0.935	0.937	0.938	0.943	**0.948**
mean	0.938	0.938	0.940	**0.941**	**0.941**	0.943	0.945	0.947	0.950	**0.951**

PE = positional encoding. C = canonical planes. D = dynamic planes.

Table 2. **Object-level 3D reconstruction from point clouds.** This table shows the detailed per-category results on ShapeNet dataset comparing ours and ConvONet [5].

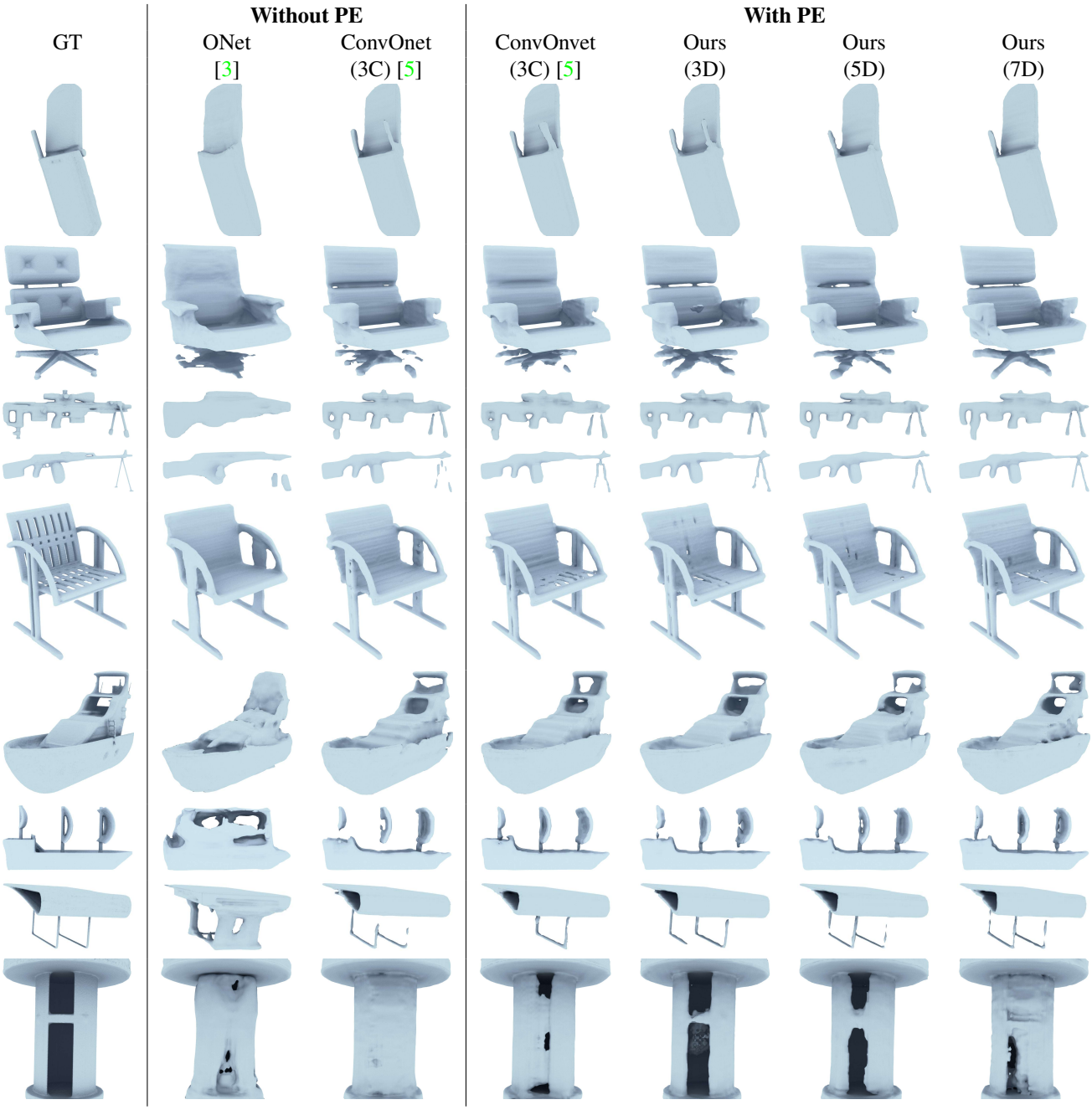

Figure 3. **Qualitative comparison of object-level reconstruction from point clouds.** We selectively choose the objects with intricate geometric details, such as thin components and holes.

References

[1] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[2] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016.

[3] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[4] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[5] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[6] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.