Supplementary Material – Guiding Diffusion-Based Articulated Object Generation by Partial Point Cloud Alignment and Physical Plausibility Constraints

Jens U. Kreber and Joerg Stueckler
University of Augsburg
Augsburg, Germany
{jens.kreber,joerg.stueckler}@uni-a.de

1. Introduction

In this supplementary material we provide additional details and results for our approach. We first provide details on the modifications of the open-source implementation of NAP used in our experiments. Then, we detail results of our hyperparameter study. Finally, we provide a table of retrieved meshes as comparison to the meshes extracted from SDFs in our method.

2. Modifications of open-source implementation of NAP

In the open-source implementation of NAP provided by the authors [1], we changed the edge feature embedding for the graph network to be consistent with the described method according to our understanding. Additionally, we changed the order of sample and reference set in the calculation of the generative metrics which is more consistent with the original formulation of the metrics for our evaluation. For reference, we compare our computed metrics on the baseline (unguided) method against the values obtained with the open-source implementation of NAP (both with the modified edge embedding) and also provide the numbers as reported in the NAP paper. We evaluate with 5 different sample sets from different random seeds and compute the mean and standard deviation (in parentheses). See Table 1. It can be seen that with the modified feature embedding but old metric implementation (NAP metric), results differ only slightly from those reported in the original paper (from NAP). With the adapted metrics (ours), the deviation from the original results further increases. Additionally to the metrics MMD and 1-NNA, we report the COV (coverage) metric as in the NAP paper. As for the other metrics, details on the base metrics can be found in [2].

[1] https://github.com/JiahuiLei/NAP

3. Hyperparameter Study

Figure 1 to Figure 6 show the effect of the two hyperparameters in the respective loss setting. For our full method, we decide on the base hyperparameters $w_{\text{pc,base}} = 45, w_{\text{pen,base}} = 2, w_{\text{mob,base}} = 2$. Figure 7 shows the effect of the inverse temperature parameter τ. We considered values $\tau = 10^2$ and 10^3 as candidates due to simultaneously low mean E_{pc}, MMD and 1-NNA. We conducted additional experiments on point clouds from the validation split for the pc and pc+pen+mob variants. While the results are mixed, we found that $\tau = 10^3$ performs better in 4 and worse in 3 value/metric combinations and similarly in the remaining ones, so we choose this value for further experiments.

4. Mesh retrieval

In Figure 8, we show retrieved part meshes instead of meshes extracted from SDF latent codes and compare their mean distances to the observed point cloud.

5. Run Times

We provide the average run times of several ablations from Table 1 in the main paper in Table 2. The computations were all done on a cluster node with 16 CPU cores and an Nvidia A40 GPU.

References

[1] Jiahui Lei, Congyue Deng, William B. Shen, Leonidas J. Guibas, and Kostas Daniilidis. NAP: Neural 3D articulated object prior. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[2] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge J. Belongie, and Bharath Hariharan. PointFlow: 3D point cloud generation with continuous normalizing flows. In *Proc. of IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2019.

	MMD ↓ (ours)	MMD ↓ (NAP metric [1])	MMD ↓ (from NAP [1])
SDF	0.0284 (0.0004)	0.0272 (0.0012)	0.0268
retrieval	0.0265 (0.0008)	0.0226 (0.0013)	0.0215
	COV ↑ (ours)	COV ↑ (NAP metric [1])	COV ↑ (from NAP [1])
SDF	0.4833 (0.0109)	0.4976 (0.0097)	0.4944
retrieval	0.4806 (0.0131)	0.5301 (0.0143)	0.5234
	1-NNA ↓ (ours)	1-NNA ↓ (NAP metric [1])	1-NNA ↓ (from NAP [1])
SDF	0.6477 (0.0204)	0.5615 (0.0057)	0.5690
retrieval	0.5739 (0.0227)	0.5367 (0.0102)	0.5412

Table 1. Comparison of our modified implementation of NAP and generative metric computation with the open-source implementations of NAP and its metrics. Ours: modified edge feature embedding; NAP metric: metric from NAP open-source implementation with our modified edge feature embedding; from NAP: results reported in original paper. Values are given as mean (standard deviation).

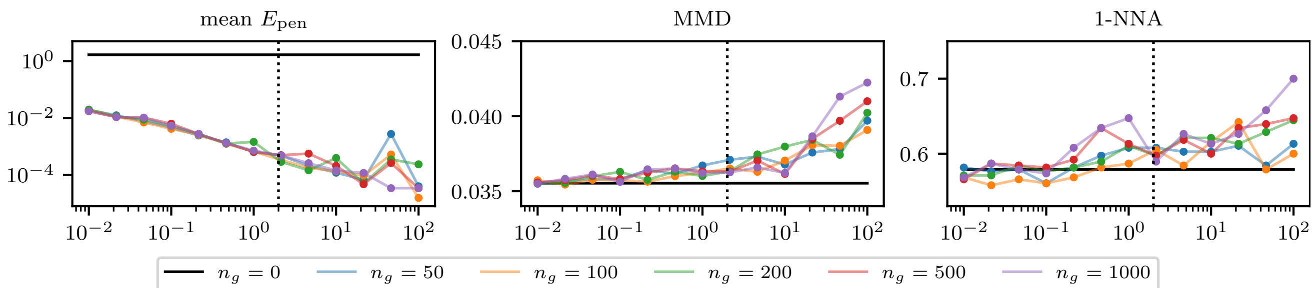

Figure 1. Hyperparameter sweep over guidance steps n_g and guidance weight w_{pen} (x-axis) for generation with the penetration guidance loss only, without category conditioning. Black line: NAP baseline. We choose $n_g = 500$ and $w_{\text{pen,base}} = 2$ (dashed line). The E_{pen} and MMD plots are identical to Figure 3 in the main paper.

cat	variant	avg run time (m:ss)
no	pc+pen+mob	1:57
no	pc	0:15
no	uncond	0:13
yes	pc+pen+mob	1:31
yes	pc	0:15
yes	uncond	0:12

Table 2. Average run time per generated sample (minutes:seconds) for several variants from Table 1 in the main paper. Average taken over 150 samples (5 each for the 30 point clouds from the test set). Adding the physical plausibility losses has a notable impact on run time. The category-aware model is slightly faster than the category-unaware model.

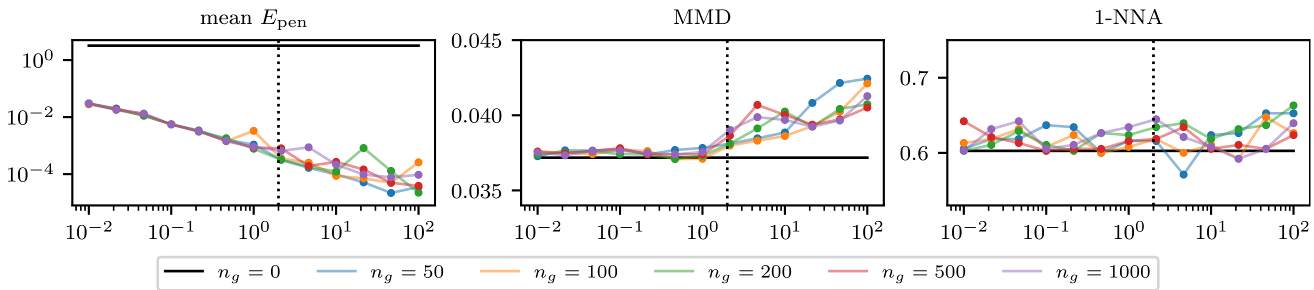

Figure 2. Hyperparameter sweep over guidance steps n_g and guidance weight w_{pen} (x-axis) for generation with the penetration guidance loss only, with category conditioning. We choose $n_g = 500$ and $w_{\text{pen,base}} = 2$ (dashed line).

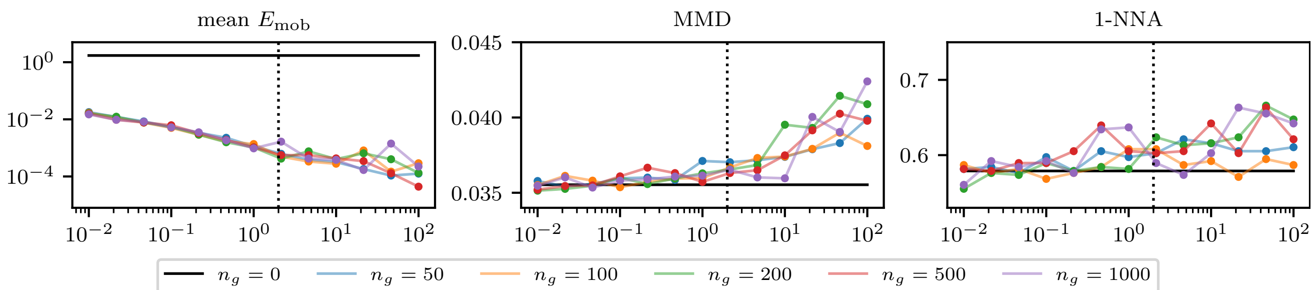

Figure 3. Hyperparameter sweep over guidance steps n_g and guidance weight w_{mob} (x-axis) for generation with the mobility guidance loss only, without category conditioning. Black line: NAP baseline. We choose $n_g = 500$ and $w_{\text{mob,base}} = 2$ (dashed line).

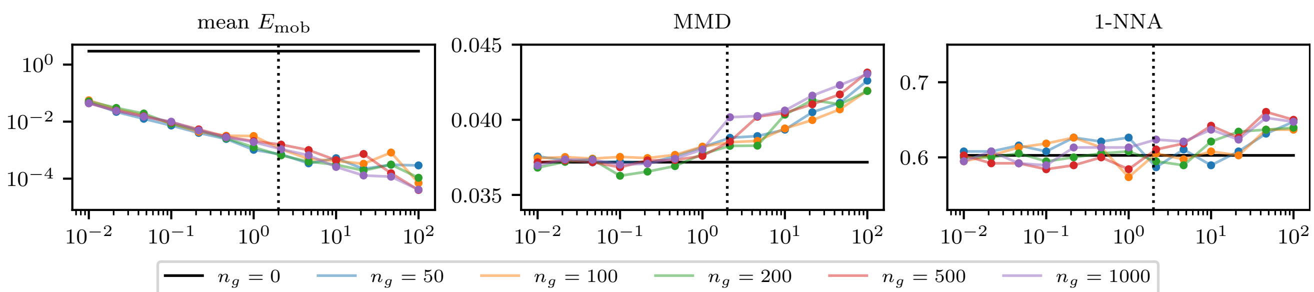

Figure 4. Hyperparameter sweep over guidance steps n_g and guidance weight w_{pen} (x-axis) for generation with the mobility guidance loss only, with category conditioning. We choose $n_g = 500$ and $w_{\text{mob,base}} = 2$ (dashed line).

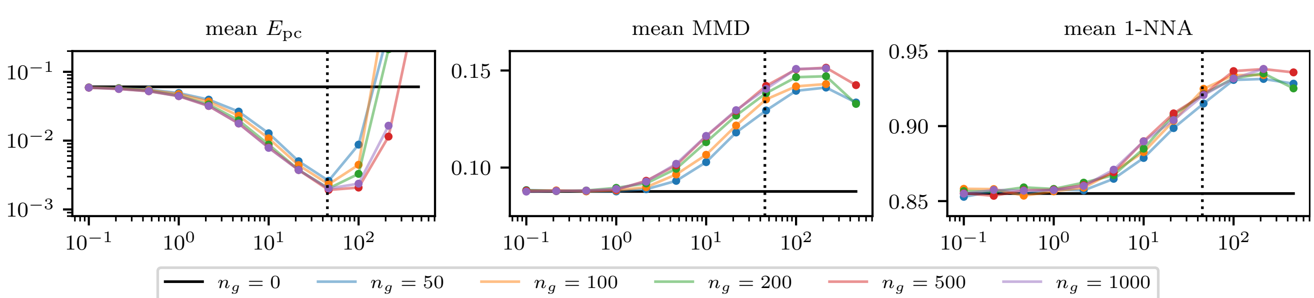

Figure 5. Hyperparameter sweep over guidance steps n_g and guidance weight w_{pen} (x-axis) for generation with point cloud guidance loss only, without category conditioning. Black line: NAP baseline. We choose $n_g = 500$ and $w_{\text{pc,base}} = 45$ (dashed line). For high guidance weight and number of guidance steps, the generation may diverge to implausible results, which is the cause for missing points in this plot.

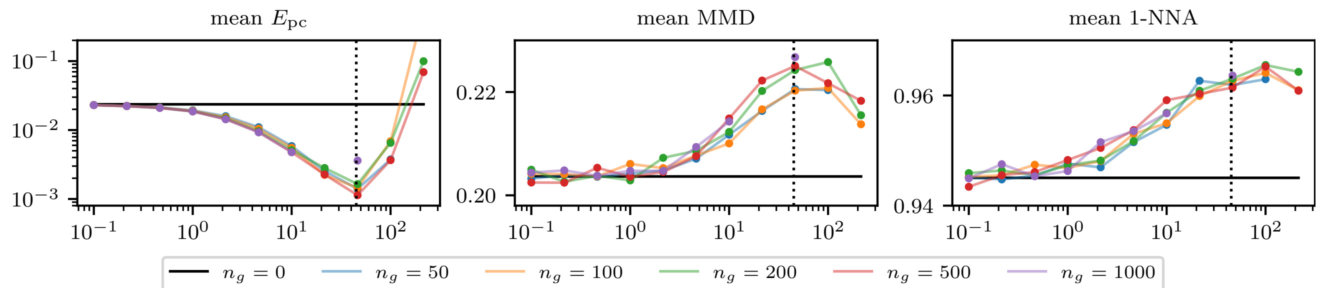

Figure 6. Hyperparameter sweep over guidance steps n_g and guidance weight w_{pen} (x-axis) for generation with point cloud guidance loss only, with category conditioning. We choose $n_g = 500$ and $w_{\text{pc,base}} = 45$. For high guidance weight and number of guidance steps, the generation may diverge to implausible results, which is the cause for missing points in this plot.

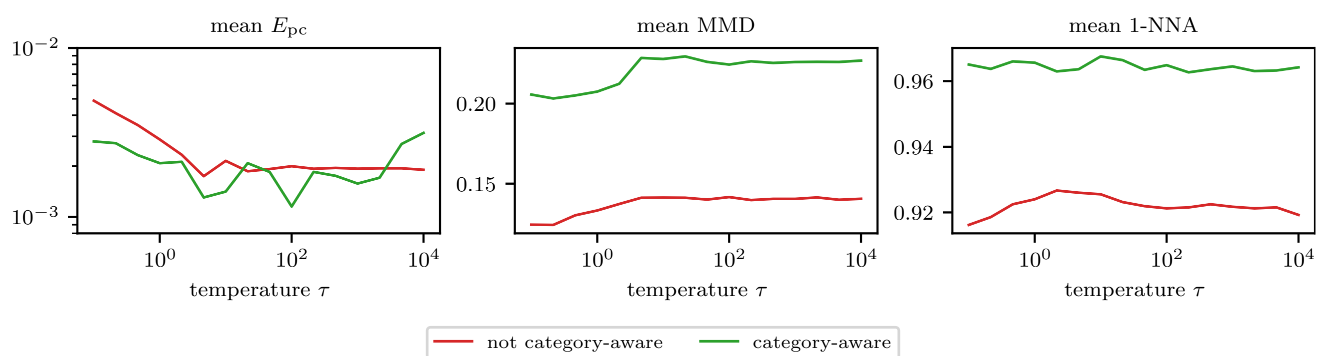

Figure 7. Hyperparameter sweep over inverse temperature parameter τ for generation with point cloud guidance loss only, with and without category conditioning. Other parameters are $n_g = 500$, $w_{pc} = 45$. We choose $\tau = 1000$.

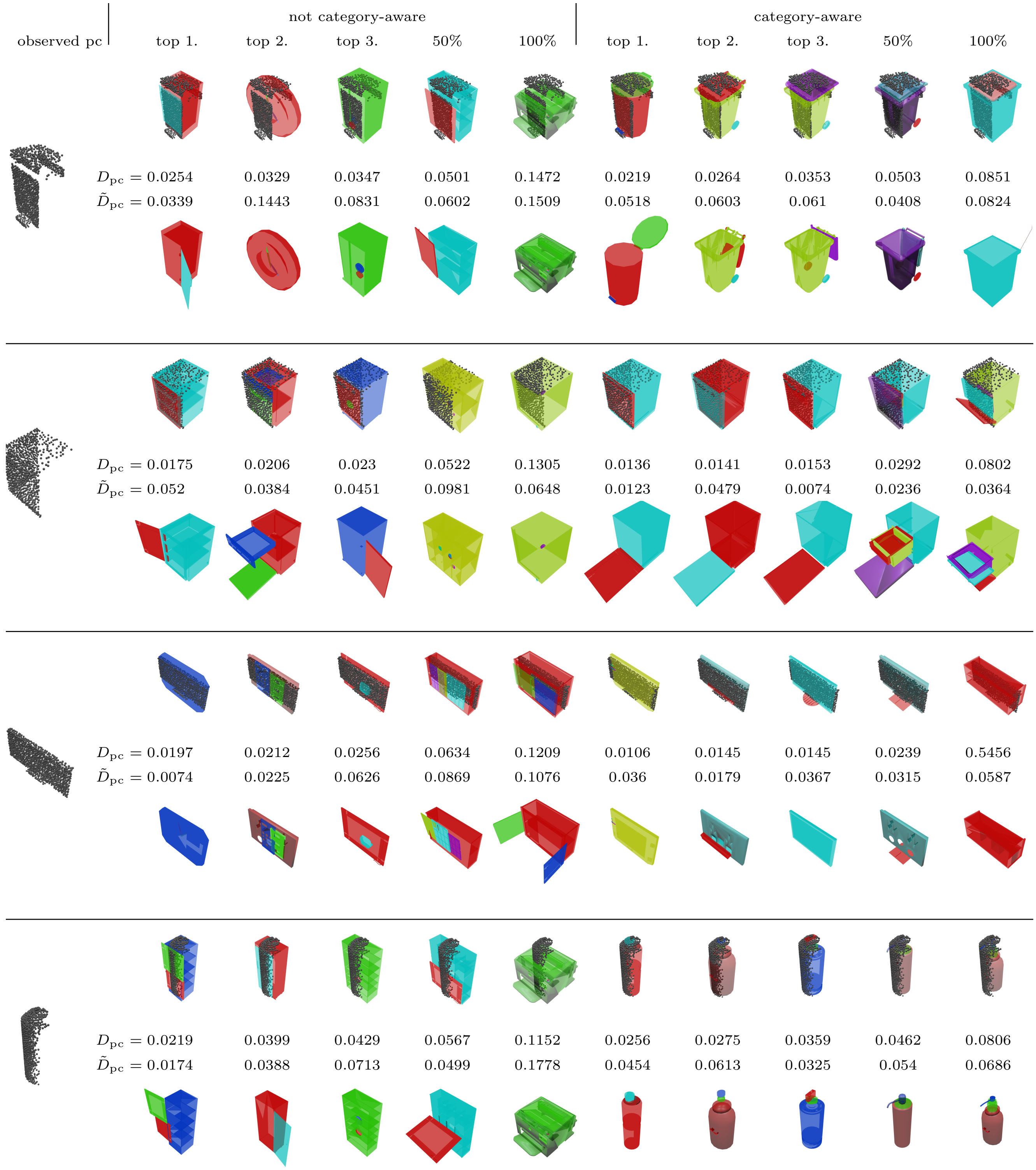

Figure 8. Shown are the same generated articulation graphs as in Figure 4 in the main paper, but with mesh retrieval instead of SDF-based mesh extraction. We keep the sorting by D_{pc} of the extracted objects and also provide those values for reference. In addition, we display the mean point distances to the retrieved meshes as $\tilde{D}_{\text{pc}}$.