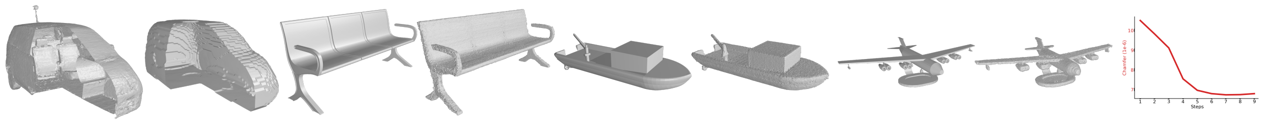

NDF (cars) SAL[4] (cars) GT Ours GT Ours GT Ours Steps vs Error

Figure 1: Left - Comparison of NDFs with SAL. Middle three - NDF reconstructions for different ShapeNet categories. Right - Graph of: number of projection steps against Chamfer distance between reconst. and GT.

Cars	10000 Points	3000 Points
SAL[4]	6.39	7.39
Ours	**0.074**	**0.275**

Full ShapeNet	OccNet	PSGN	DMC	IF-Net	Ours
3000 Points	4.0	4.0	1.0	0.2	**0.05**

Table 1: Chamfer-L_2 $\downarrow$ results $\times 10^{-4}$. **Left Table:** Results of point cloud completion for unprocessed cars from 10.000 points and 3.000 points. **Right Table:** Results on full watertight ShapeNet from 3000 points.

We thank all reviewers for their useful feedback. Reviewers acknowledge that NDFs are "well founded", "simple and general, which promises wider applicability"[**R2**], they "certainly hold technical value"[**R1**] and all agree on their wide applicability. NDFs are interesting for researchers [**R4**] to build on it – we will make all code and models available to the research community.

Comparison with SAL[4] and full ShapeNet baselines [**R2**,**R4**] We compare to SAL on the car class of ShapeNet and we observe that it can reconstruct the watertight outer surface whereas NDF can reconstruct also the inner structures. We tried training SAL on the full ShapeNet but it diverges from the signed distance, so no meshes can be produced. We contacted the authors, they recommended us to train SAL on individual classes. There is no evidence that it should work competitively on all classes together. In conclusion, SAL is useful to watertight noisy meshes, and NDF is more flexible on the types of surfaces it can output. We also added a comparison to other prior works on full ShapeNet. We conduct the experiment on watertight surfaces as required by these works. We find that NDF outperforms all baselines.

Unclear benefits of representing open surfaces [**R1**,**R3**] Our NDFs 1.) are directly trainable on raw scans and 2.) directly represent open surfaces in their output. This allows us to train on detailed ground truth data without lossy watertighting *and* to implicitly represent a much wider class of shapes. All reviewers agree on NDFs wider applicability. Note, our paper is not limited to toy examples [**R1**,**R3**], we show examples for complex interior structures of cars, reconstruction of garments, completion of huge multi-object scenes.(Scenes in Supplementary) Reviewers [**R1**,**R2**,**R4**] acknowledge our faithful 3D reconstruction results.
As an add-on, and to inspire future research, we also show toy examples for 2D representation of curves, functions and manifolds, demonstrating NDFs are not limited to closed 3D surfaces. This is interesting [**R1**], when a curve with multiple y values for one x should be represented, as in contrast to NDFs, a NN trained to predict y from x, would create an average.

Experiment details[**R3**,**R4**] *"Is there a test/train split, and are quantitative statistics provided on the test set?" "what is the network architecture, learning rate?"*[**R3**] Please see the Supplementary for the answer. To make the paper self-contained, we added more details from the reference architecture of [13] that we build on. We follow the suggestion and added training time. We also transferred key information into the main paper.

NDF Properties [**R2**] *More information about run-time and num_steps* We added this ablation see figure above, each num_ step takes 3.7 sec on a Tesla V100 for 1 Mio. points. *Why are normals computed in the surrounding a good approximation?* A UDF is non-differentiable at 0 but its surrounding gives normal information: if we move a small step away from the surface in normal direction, the closest point is unchanged and we estimate the correct normal. If we move away in not normal direction, we rely on local smoothness of the normal field.

NDF Limitations [**R4**,**R2**] 1.) Unlike SDFs, no standard contouring algorithm like Marching Cubes exists for NDFs. Using BPA[8] for meshing on a predicted dense point cloud can lead to large amounts of triangles and is slow to compute. 2.) As for all implicit models, a lot of points have to be predicted at inference. To save time for NDF (and SDF) inference, our point to surface projection can be explored to do coarse predictions that give distance and direction on where to do finer sampling.

Works on unsigned distance fields in traditional CG/CV? [**R4**] We investigated traditional literature (lines 52-56 for point to surface mapping and in lines 197-203 for rendering). Distance fields are predominantly used as *signed* distance fields. We added more related classical papers on shape matching and sculpting.

Other suggestions and comments We focused on common and most pressing concerns in the rebuttal. We value all other suggestions by reviewers and incorporated them into the paper.