Small Steps and Level Sets: Fitting Neural Surface Models with Point Guidance

Supplementary Material

A. Further Algorithm Details

We provide more details on the high level workings of our algorithm here. For implementation details, see Appendix B. We show visualzations of our algorithm in 2D in Fig. 9 (one step with explanations) and in Figs. 10 and 11 (all steps for one shape).

A.1. Initialization

As explained in Sec. 4, we initialize our network to the exterior level set of radius $\delta_{\text{init}} = 16\delta_{\mathcal{X}}$. We mentioned that we can do this robustly in Sec. 4.3 because (1) $\delta_{\text{init}} \gg \delta_{\mathcal{X}}$ so we can estimate good normals for any $y \in \Omega_{\delta_{\text{init}}}$, and (2) higher radius exterior level sets are a lot less complex.

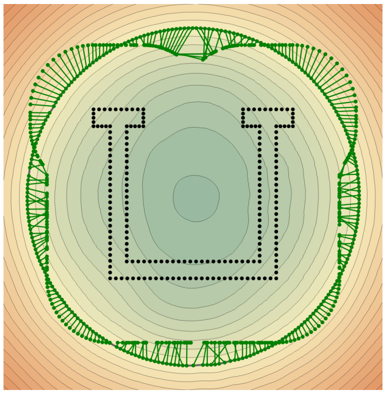

Figure 8. Visualization of our fast initialization in 2D.

To see (1), note that for any $y \in \Omega_{\delta_{\text{init}}}$, the ground truth normal n_y is given by

$$n_y^{\text{GT}} = \widehat{y - x^\star_{\mathcal{S}}(y)} \tag{21}$$

where $x^\star_{\mathcal{S}}(y)$ is the closest point on $\mathcal{S}$ to y. We cannot calculate this as we do not know $\mathcal{S}$, but we have $\mathcal{X}$ which we assume is sampled densely from $\mathcal{S}$. In particular, we assume there is some sampling radius $\delta_{\mathcal{X}}$ (which we approximate for our algorithm). Thus $\|x^\star(y) - x^\star_{\mathcal{S}}(y)\|_2 \approx \delta_{\mathcal{X}}$ where $x^\star(y) = x^\star_{\mathcal{X}}(y)$ is the closest point in $\mathcal{X}$ to y. Since at initialization $\|y - x^\star_{\mathcal{X}}(y)\|_2 \approx \delta_{\text{init}} \gg \delta_{\mathcal{X}}$, it follows that the direction of $y - x^\star_{\mathcal{S}}(y) \approx y - x^\star_{\mathcal{X}}(y)$, so our estimated normal directions

$$n_y = \widehat{y - x^\star_{\mathcal{X}}(y)} \tag{22}$$

are fairly accurate.

To see (2), we refer the reader to Appendix F.

Thus to actually initialize the network at $\Omega_{\delta_{\text{init}}}$, we first get our guiding points $\mathcal{Y}$ to densely sample the exterior level set, estimate good normals for each guiding point using Eq. (22), and then optimize using our loss with a high weighting for the approximate SDF loss as the approximate SDF is fairly accurate with good normals.

We can quickly get our guiding points $\mathcal{Y}$ to densely sample $\Omega_{\delta_{\text{init}}}$ by starting them at a sphere bounding $\mathcal{X}$ and move them inwards at random directions until $d(y, \mathcal{X}) \leq \delta_{\text{init}}$. We then find the closest point in $\mathcal{X}$, $x^\star_{\mathcal{X}}(y)$, and move it back along that line containing those two points until the distance to the closest point $\mathcal{X}$ is exactly δ_{init}. We show this in Fig. 8 for 2D. However for complex shapes, $\Omega_{\delta_{\text{init}}}$ may still be somewhat complex, so to get a dense sampling of the whole exterior level set we also simulate 'bouncing' off the level sets to reach other surfaces.

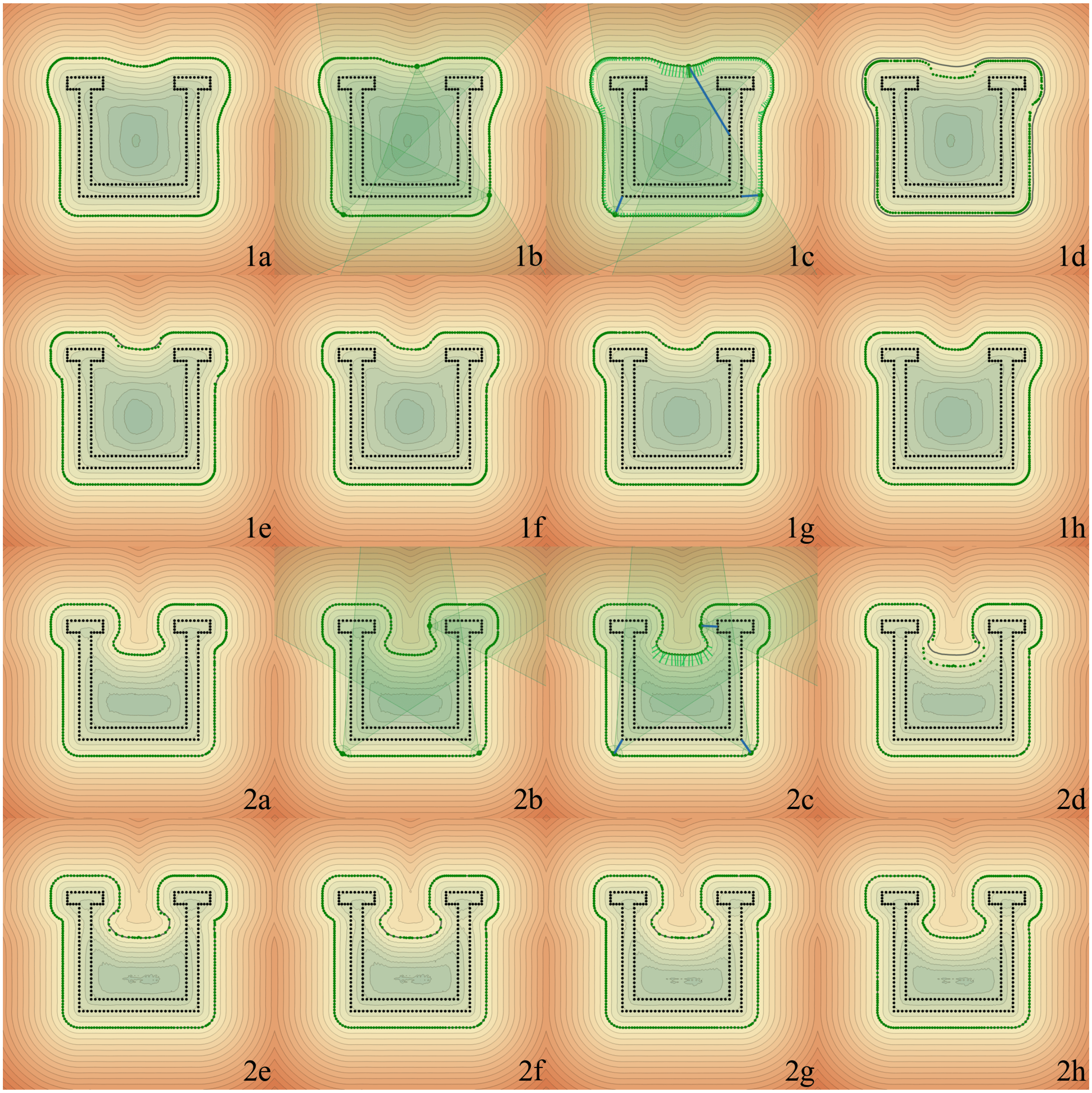

Figure 9. Extended visualisation of one iteration of our algorithm (a-h), at two different points in time (1 and 2). This relates to the algorithmic details explained in Appendix A.2.

A.2. Optimization and Resampling

One issue with optimizing the neural SDF Φ with respect to the guiding points $\mathcal{Y}$ is that the movement of the guiding points $\mathcal{Y}$ from Sec. 4.1 are unlikely to be consistent with each other. The surface implied by $\mathcal{Y}$ is likely to be quite rough and have outliers, so it is unlikely that SDF optimization, which requires is regularized to make Φ fairly smooth, will have its zero level set surface $\mathcal{R}$ interpolating $\mathcal{Y}$. This leads to a high optimization loss and defects in Φ. This is a problem, as we use SDF values and gradients to resample the zero level set and to get normals for the new points (which is important for figuring out where to move $\mathcal{Y}$ next), so the neural SDF must be sufficiently faithful to a true SDF.

Thus we do two rounds of SDF optimization. In the first round, we optimize the SDF parameters with low regularization, leading to $\mathcal{R}$ and $\mathcal{Y}$ being close but inconsistent (1e and 2e in Fig. 9). We then project $\mathcal{Y}$ back onto the newly optimized $\mathcal{R}$ (1f and 2f in Fig. 9). We then optimize the SDF parameters with the new $\mathcal{Y}$ with high regularization, which leads to Φ being much more faithful to SDF properties, and $\mathcal{R}$ and $\mathcal{Y}$ being quite consistent (1g and 2g in Fig. 9).

Finally we resample the guiding points $\mathcal{Y}$ on this new surface $\mathcal{R}$ (1h and 2h in Fig. 9). To do this, we first sample noisy points around $y \in \mathcal{Y}$ and then project them to the surface (using the fact that Φ is now a good approximation of an SDF) to get a dense sampling of $\mathcal{R}$. We then use farthest point sampling on the projected points to also get a uniform sampling of fixed size.

Note that this method is similar to predictor-corrector methods for following the zero path [33], where we first produce a 'predictor' point θ not on the zero path (this is given by the first optimization with high regularisation, it is not on the zero path as it is not a good SDF) but in the right direction for optimizing the loss. We then apply 'corrector' iterations to bring it back on the path (the second part of the optimization).

A.3. Algorithm with ground truth normals

We note that in our initialization, if we have ground truth normals for each $x \in \mathcal{X}$, a better way to get good approximate normals for y on some exterior level set of $\mathcal{X}$ is to find the closest point in $\mathcal{X}$, $x^{\star}_{\mathcal{X}}(y)$, and use its ground truth normal.

Thus for our algorithm with ground truth normals, we first optimize for the level set of radius δ_{init} using our current initialization procedure except with our better normal estimates. We then apply the initialization method again except for the level set of radius δ_0, again using our better normal estimates, noting that we start our optimization from the state of the network after the first run. We keep applying this for smaller level set radii until we reach δ_f, and then finally optimize on our input points. However as we have the ground truth normals for our input points, we still use the approximate sdf loss at this last stage of optimization unlike our version of the algorithm without ground truth normals.

B. Implementation Details

We primarily use *PyTorch* [36] for our implementation. Distance to set queries are efficiently done using the `KeOps` library [14] within the `PyTorch` framework. For visualization and debugging during training we used the `rerun.io` library.

B.1. Domain and Preprocessing

We use the domain $[-1.1, 1.1]^3 \subset \mathbb{R}^3$. Given input points, we first center them around the origin, and then isotropically scale it down by 1.2 times the norm of the furthest point from the origin. Thus after preprocessing the input points fit within the closed ball centered at the origin of radius $\frac{5}{6}$.

To calculate $\delta_{\mathcal{X}}$, our conservative estimate of the radius of sampling of $\mathcal{X}$, we find the closest 4 neighbours for each $x \in \mathcal{X}$, i.e., knn$(x, 1)$,...,knn$(x, 4)$ where knn(x, k) is the distance to the k^{th} closest neighbour to x. We then separately sort the distances for each k, and take the largest 5% of each of these. $\delta_{\mathcal{X}}$ is then the average of all of these distances.

B.2. Moving the Guiding Points

We do our implementation in using the `KeOps` library [14] within the `PyTorch` framework due to its ability to handle computations that usually use a large amount of memory to do efficiently. In particular we define a cost function for the cone C_y and for the half ball H_y in order to implement the optimization in Eq. (12) as a single argmin using `KeOps`. We set $a_y = 0.9$ and $s_m = 0.1$.

Furthermore we note we remove guiding points y if there is no $x \in \mathcal{X}$ inside the set $H_y \cup \mathcal{C}_y$. We also remove guiding points y if get get moved to a point y' that is in the outside region per the current Φ, i.e., $\Phi(y', \theta) > \varepsilon$ for some $\varepsilon > 0$. The intuition behind this is that sometimes our guiding points, when moving inward, close up a hole in the shape and thus move to the outside of the volume enclosed by the current network's zero level set.

After moving the points, we also check whether they actually went into the interior of the current $\mathcal{R}$ by querying Φ for whether $\Phi(y) < 0$, $y \in \mathcal{Y}$. If the adjustment distance was very small, then y is probably close to Ω_δ and violating this is fine. Otherwise, such points are removed.

B.3. Network implementation and Optimization

Architecture. We use a parameter encoding based coordinate network as inspired by NGLOD [42], where learnable features are stored in space. We use a 2 layer network with 256 hidden units. The network takes in the position, as well

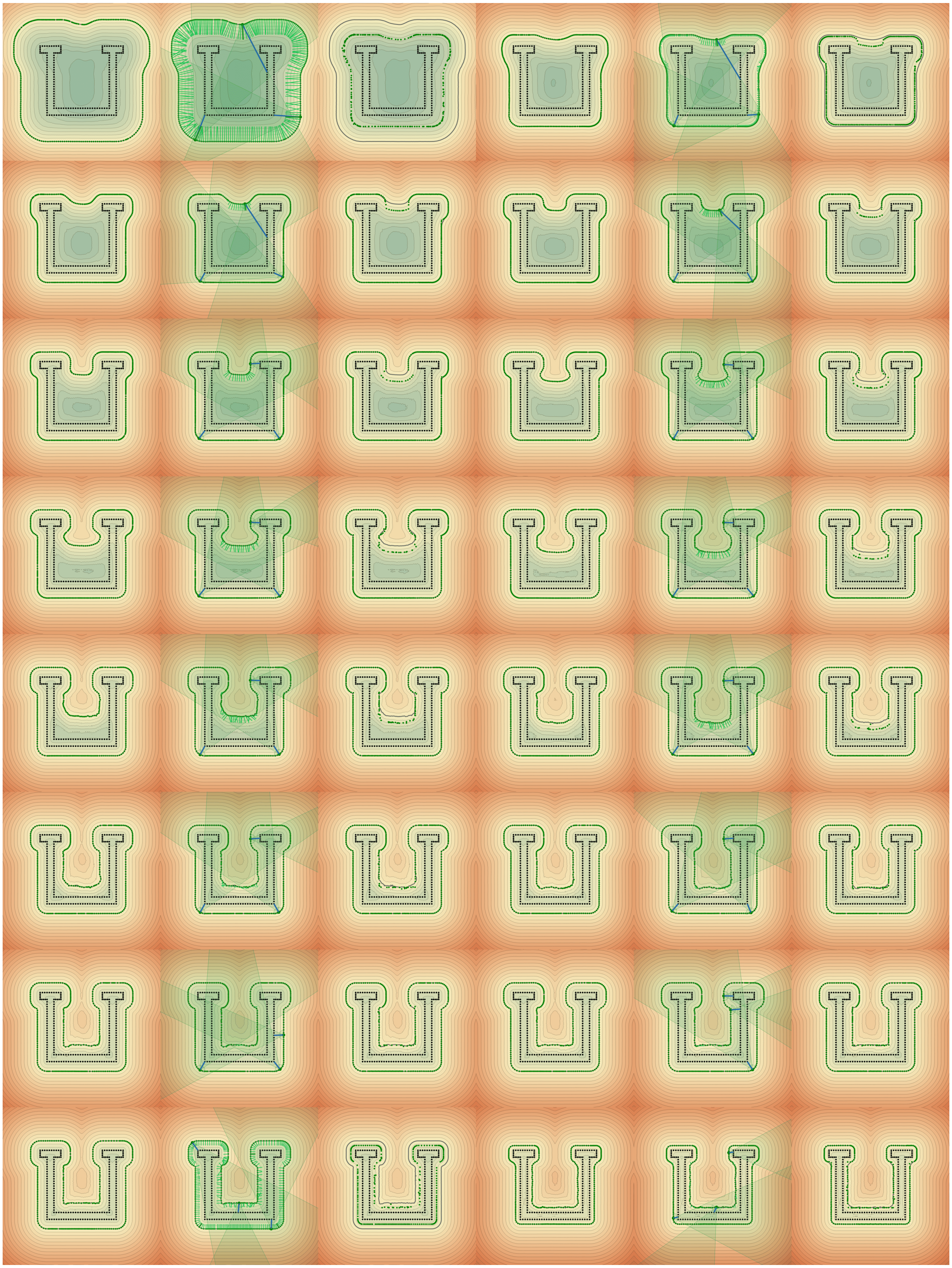

Figure 10. Visualization of our method in 2D (first half shown).

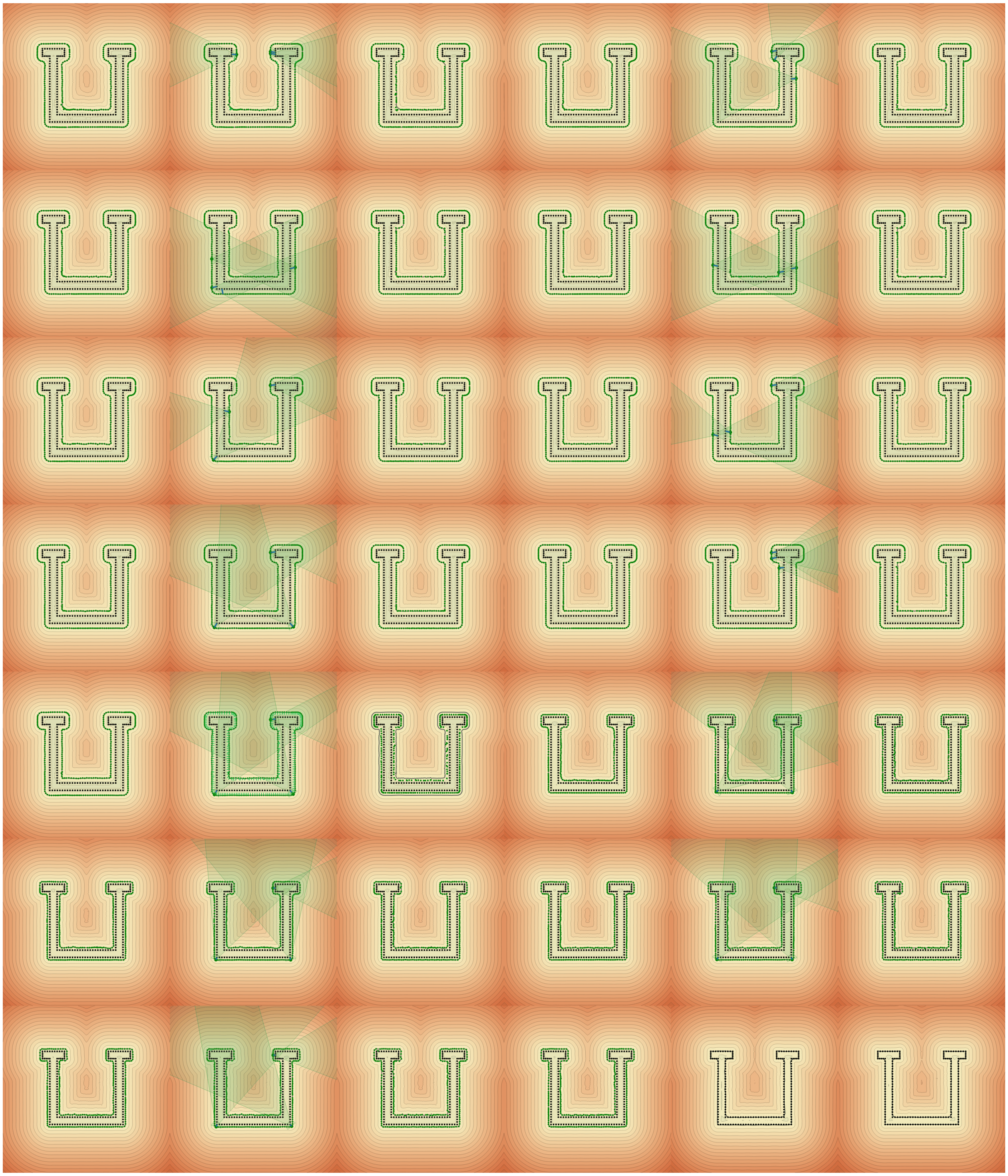

Figure 11. Visualization of our method in 2D (second half shown)

as Fourier feature [43] and feature grid [31, 42] encodings of the position. We use a dense grid of depth 7 and 2 values per level. The grid features are more than 99% of our parameters.

We implement this using the implementation of such networks given by SDFStudio [48], which builds on top of NerFStudio [44]. The benefit of this is that they have implemented many methods and thus their framework has many tricks for both time and learning efficiency (for example using fast fully fused MLPs from `tiny-cuda-nn` [30]).

Note that we cannot use a sparse structure, where the majority of features are concentrated around $\mathcal{Y}$, since our guiding points $\mathcal{Y}$ change during optimization. We also do not use the hash map implementation of Istant-NGP[31] since we found it to give undesirable roughness.

Sampling. When sampling domain points (i.e., from $\mathcal{D}$), we use a combination of grid points (we sample the grid with 256 values per dimension) and the octree sampling from OG-INR [25], where we build the octree around the current $\mathcal{Y}$ at each small step. We also weight domain samples more if they are closer to the zero level set.

Losses. When we implement our loss terms, we have $\mathcal{L}_{\text{zls}}(\theta, \mathcal{Y}$ (computed on $\mathcal{Y}$),$\mathcal{L}_{\text{input}}(\theta, \mathcal{X}, \delta)$ (computed on $\mathcal{X}$), $\mathcal{L}^{\mathcal{Y}}_{\text{eik}}(\theta)$ (computed on $\mathcal{Y}$), $\mathcal{L}^{\mathcal{D}}_{\text{eik}}(\theta)$ (computed on $\mathcal{D}$), $\mathcal{L}_{\widetilde{\text{sdf}}}(\theta, \mathcal{Y})$ (computed on $\mathcal{D}$), $\mathcal{L}_{\text{normal}}(\theta, \mathcal{Y}, \mathcal{N}(\mathcal{Y}))$ (computed on $\mathcal{Y}$) and $\mathcal{L}_{\text{pull}}(\theta, \mathcal{Y})$ (computed on $\mathcal{D}$). Our weightings for these losses are 100, 100, 2, 1, 0, 0, 0 when finetuning. Note that we do not use the approximate SDF loss when finetuning on the input points as they might not be complete or might have outliers (the preprocessed subset still has interior points for some shapes). When confident in our normals (e.g., at initialization or for our method with input GT normals) we use 100, 100, 10, 50, 50, 10, 20 at the start (if it is initialization and we need to drastically change our shape) and 100, 100, 5, 2, 2, 2, 1 at finetuning (as close to the zero level set is most important). During our small steps, our first optimization has the loss weightings 500, 500, 10, 2, 100, 0, 1 (we want to overfit to the new guiding point locations, with some compromises to whether Φ is a good SDF), and 100, 100, 50, 50, 10, 0, 50 in our second optimization (now that the new zero level set is determined, we want to make Φ be an SDF).

B.4. Projection and Resampling

When projecting points to the zero level set, we apply the standard SDF projection formula with 50 iterations, and afterwards remove points where we this has failed. We check this by (1) querying Φ after projection to see if the SDF is far away from 0, and (2), check the gradient of Φ when querying, as if it is not close to 1 then the point probably became stuck during projection iterations.

To resample the zero level set, we first add Gaussian noise to each current sample, and project these points back to the zero level set. We then apply farthest point sampling to get it to a fixed sample size.

Guidance	Squared Chamfer ↓			IoU ↑		
δ_f	Mean	Median	Std	Mean	Median	Std
0.005	1.02e-4	9.92e-5	5.94e-5	0.3340	0.1687	0.3129
0.010	9.01e-5	**4.68e-5**	7.77e-5	0.7987	0.9420	0.2674
0.015	8.94e-5	7.17e-5	6.72e-5	0.8958	0.9408	0.1003
0.020	9.95e-5	7.13e-5	7.28e-5	0.8712	0.9090	0.1109
$0.5\delta_{\mathcal{X}}$	1.06e-4	9.50e-5	5.67e-5	0.2531	0.1730	0.2303
$\delta_{\mathcal{X}}$	**5.51e-5**	4.87e-5	**3.12e-5**	**0.9153**	**0.9598**	**0.0906**
$2\delta_{\mathcal{X}}$	1.01e-4	7.45e-5	7.64e-5	0.8717	0.9350	0.1220

Table 4. Sensitivity analysis on the 20 highest SION ShapeNet shapes. We only change the values of our level set radii δ by varying the final radii δ_f, and setting the initial radii to $\delta_0 = 16\delta_f$. Note that δ_X is an adaptive value (bottom three rows); on these 20 shapes it has a mean of 0.00988 and a range of [0.00582, 0.01370].

We also oversample on regions that need to move more. We determine this by the distance of the current samples to the input point cloud. We find this improves stability of the steps (the optimization has more points to guide in in areas it needs to change) and convergence speed (the optimization is more likely to interpolate those points with the re-weighting for a fixed number of iterations).

We determine when the small steps have converged to the level set by computing the distance of our guiding points to the input points, and determining whether every guiding point is within $\delta + \varepsilon$ of an input point, where $\varepsilon = 1e-2$.

C. Convergence and hyperparameter sensitivity

We provide a sensitivity analysis on the final level set radii δ_f in Tab. 4, which sheds light on the two cases where the point moving strategy does not converge to the GT points. First note that our choice of $\delta_f = \delta_{\mathcal{X}}$ performs the best ($\delta_{\mathcal{X}}$ computes an approximate sampling radius per shape). ***Case 1***: if δ_f is smaller than the gap between input points, then many guiding points permeate the surface of the shape. This leads to a sharp reduction in performance ($\delta_f = 0.005$, $0.5\delta_{\mathcal{X}}$). ***Case 2***: if the entry to the shape's concavities are smaller than δ_f, then such concavities will not be explored. We found that increasing δ_f led to a gradual decrease in performance ($\delta_f = 0.015$, 0.020, $2\delta_{\mathcal{X}}$), while still outperforming competing methods.

D. Runtime comparison and ablation

We compare our average runtime to other methods in Tab. 5. Our method's runtime is primarily dependent on the complexity of the shape being reconstructed, which is demonstrated in Fig. 7. The runtime is dominated by optimizing Φ (Line 7 of Alg. 1), which is repeated as many times as is needed to explore concave regions of the shape. Thus for the simplest shapes, our method's runtime is similar to OG-

Method	Parameters	Time per iter. (s)	Num iters	Time (s)
N Est. +SPSR	-	-	-	42
SIREN (wo n)	264K	0.052	10000	520
SAL	2.1M	0.175	10000	1750
DiGS	264K	0.120	10000	1200
SAP	120K	-	-	330
OG-SIREN	264K	0.158	600	135
OG-NGLOD	68.7M	0.173	300	92
SDF + n (Our Impl.)	17.5M	0.052	2100	180
SDF (Our Impl.)	17.5M	0.052	2100	180
PG-SDF + n (Ours)	17.5M	0.052	7500	430
PG-SDF (Ours)	17.5M	0.052	13500	600

Table 5. Time and number of parameters for each model on ShapeNet. We ran our experiments on a single Nvidia RTX 2080 GPU and an i7-8700K CPU. Since our method adapts the number of iterations based on the complexity of the shape, we report average times here (same has been done for OG-SIREN and OG-NGLOD, whose initialisation time varies based on shape complexity).

Guidance				Squared Chamfer ↓			IoU ↑			Time(min)
$\mathcal{Y}_0$	$\mathcal{Y}$	δ_0	δ_f	Mean	Median	Std	Mean	Median	Std	
×	×	–	–	2.41e-3	1.18e-3	2.65e-3	0.2795	0.2255	0.2078	**1.41**
✓	×	–	–	2.63e-4	1.91e-4	2.31e-4	0.5694	0.6091	0.2278	2.23
✓	✓	4	2	9.89e-5	7.70e-5	7.95e-5	0.8657	0.9163	0.1219	10.8
✓	✓	1	1	8.08e-5	7.52e-5	5.66e-5	0.8538	0.9386	0.1602	19.8
✓	✓	4	1	**5.51e-5**	**4.87e-5**	**3.12e-5**	**0.9153**	**0.9598**	**0.0906**	22.7

Table 6. Ablation results on the 20 highest SION ShapeNet shapes, with average runtime. Note that since these are the most difficult shapes, the runtimes are much higher than the average over the whole dataset. $\mathcal{Y}_0$: uses our SDF initialization strategy instead of spherical initialization [15]. $\mathcal{Y}$: uses our point guidance strategy. δ_0/δ_f: starting and final level set radii, given in multiples of $\delta_{\mathcal{X}}$.

SIREN, and for most shapes, our method's runtime is similar to SIREN (wo n). Note that only OG-SIREN and OG-NGLOD also change their runtime based on the complexity of the shape, all other methods have a fixed runtime. Even with more iterations, they can fail to converge on complex shapes. We also extend our ablations to include average time in Tab. 6, which shows that we can outperform other methods even with a variant of our method that is twice as fast as our default.

E. Experimental Details

E.1. Other Methods

For SAL, IGR, SIREN, FFN [43] and NSP we report results from the papers of NSP [46], and note that all these methods have code available.

For DiGS [5] and OG-SIREN/OG-NGLOD [25] we report the results from their paper and have also ran the implementation released by the authors. For SAP [39] we use the results from the OG-INR paper [25], and have double checked the results with the implementation released by the authors.

Following OG-INR, for SPSR [22] and for normal estimation, we use the implementation in Open3D [50] with default settings.

E.2. Metrics

We follow the evaluation procedure of DiGS [5] and OG-INR [25], which uses Squared Chamfer and IoU on ShapeNet.

Squared Chamfer distance is defined as

$$d_C^{sq}(\chi_1, \chi_2) = d_{\vec{C}}^{sq}(\chi_1, \chi_2) + d_{\vec{C}}^{sq}(\chi_2, \chi_1) \tag{23}$$

$$d_{\vec{C}}^{sq}(\chi_1, \chi_2) = \frac{1}{|\chi_1|} \sum_{x_1 \in \chi_1} \min_{x_2 \in \chi_2} \|x_1 - x_2\|_2^2 . \tag{24}$$

(25)

Note that the distances are in the same scaling as the input/ground truth point cloud, they have not applied any scaling before calculating distances.

For the volumetric IoU, following Occupancy Networks, given the 100k points randomly sampled in the space provided by the dataset, the ground truth occupancy labels $O_{GT}(x) \in \{0, 1\}$ provided by the dataset, and the predicted occupancy $O_\Phi(x) = [\![\Phi(x) < 0]\!]$, the IoU is given by

$$IoU_\chi(O_{GT}, O_\Phi) = \frac{\sum_{x \in \chi} O_{GT}(x) \text{ and } O_\Phi(x)}{\sum_{x \in \chi} O_{GT}(x) \text{ or } O_\Phi(x)} . \tag{26}$$

F. Level Set Complexity

Our aim in this section is to expand upon the observation in the main paper that the exterior a-level sets Ω_a $(a \geq 0)$ generally become less *complex* as a increases. In particular, Ω_a for $a > 0$ is often less complex than the zero level set $\Omega_0 = \partial\mathcal{V}$. We make (and formalize) four statements in F.1.

Before the statements, let us clarify relevant definitions. Identical to the main paper, we consider our shape to be an open volume $\mathcal{V} \subset \mathcal{D}$ where the domain of the volume $\mathcal{D} \subset \mathbb{R}^3$ is compact. We are interested in properties of the exterior a-level set Ω_a of the surface $\partial\mathcal{V}$ $(a \geq 0)$ using the definition of an exterior a-level set from the main paper. We also use the definition of a SDF from the main paper, and use $F_{\mathcal{V}}$ to denote the SDF to $\mathcal{V}$.

We will further assume that the surface $\partial\mathcal{V}$ is regular, which we now define.

Definition F.1. A surface $\partial\mathcal{V}$ is regular if at every point on $\partial\mathcal{V}$ the SDF $F_{\mathcal{V}}$ is smooth and at least one partial derivative of the SDF is nonzero. Equivalently, at each point $x \in \partial\mathcal{V}$ there exists a local parameterization $x(u,v)$ from an open subset of $\mathcal{R}^2$ to a neighbourhood of x such that $x(u,v)$ is a smooth homeomorphism and $x(u,v)$ has linearly independent partial derivatives.

Note that this is a relatively mild assumption as we can get regular approximations to our surface within any tolerance.Furthermore let us denote the convex hull of a set S by $\mathbf{conv}(S)$, and define the diameter of a surface by the following.

Definition F.2. The diameter of a surface $\partial\mathcal{V}$, denoted $d_{\partial\mathcal{V}}$, is the maximum distance between any two points on $\partial\mathcal{V}$

$$d_{\partial\mathcal{V}} = \sup_{x,y\in\partial\mathcal{V}} d(x,y) \tag{27}$$

F.1. Four Propositions

1. **As a increases Ω_a decreases in SION to zero quickly.**

   Proposition F.3. For any sphere $\mathcal{R}$ surrounding Ω_a, SION(Ω_a; $\mathcal{R}$) is zero for all $a > \alpha$, where α depends only on Ω_0. Furthermore, $\alpha \leq d_{\partial\mathcal{V}}$.

2. **As a increases Ω_a depends less on the concave regions of the initial shape.**

   Proposition F.4. As a increases, Ω_a limits towards the exterior a-level set of $\mathcal{A}_0 = \Omega_0 \cap \partial\mathbf{conv}(\Omega_0)$. In particular for any a, exterior a-level set Ω_a does not depends on points in Ω_0 that are further than $\frac{d_{\partial\mathcal{V}}}{2a}$ to $\partial\mathbf{conv}(\Omega_0)$.

3. **As a increases Ω_a limits to a convex shape.**

   Proposition F.5. As $a \to \infty$, the Hausdorff distance between Ω_a and the boundary of its convex hull $\partial\mathbf{conv}(\Omega_a)$ limits to 0. In particular, for any $a > d_{\partial\mathcal{V}}$ the Hausdorff distance is bounded by $a - \sqrt{a^2 - d^2_{\partial\mathcal{V}}}$.

4. **As a increases Ω_a limits to a sphere, though extremely slowly.**

   Proposition F.6. As a increases the principal curvatures of Ω_a either stay zero or limit to $-\frac{1}{a}$, which is the curvature of a sphere of radius a (equivalently the radii of the principal curvatures either stay ∞ or limits to a). Furthermore as $a \to \infty$, exterior a-level set Ω_a limits to a sphere.

We start by restating properties of SDFs and parellel/offset surfaces in Appendix F.2 and Appendix F.3, respectively. We then give our definitions and prove useful results in Appendix F.4. Finally, we prove each of the four above propositions in Appendix F.5.

F.2. SDF and Level Set Properties

We first note some basic properties of SDFs [28]. Let z be any point in $\mathcal{D}$. If there is a unique closest point $x^\star(z)$ in $\partial\mathcal{V}$ to z, then

- $F_{\mathcal{V}}$ is differentiable at z
- $z - x^\star(z) = F_{\mathcal{V}}(z)n(x^\star(z))$
- $\nabla_z F_{\mathcal{V}}(z) = n(x^\star(z))$
- $\|\nabla_z F_{\mathcal{V}}(z)\|_2 = \|n(x^\star(z))\|_2 = 1$

where $n(x^\star(z))$ is the outward normal of the surface $\partial\mathcal{V}$ at $x^\star(z)$ (which exists since any point on $\partial\mathcal{V}$ is its own unique closest point to $\partial\mathcal{V}$ so $F_{\mathcal{V}}$ is differentiable there).

Note that there is a unique closest point almost everywhere in $\mathcal{D}$ (i.e., the set of points have have multiple closest points is a set of measure zero), so the normal field is also defined almost everywhere. Furthermore $n(x)$ is defined everywhere on the shape's surface $\partial\mathcal{V}$ as the surface is smooth.

F.3. Parallel/Offset Surface Properties

While exterior level sets of SDFs are not deeply studied (to our knowledge), we note that surfaces formed by offsetting a constant value a in the normal direction are well studied. These are called parallel surfaces in differential geometry [13] and offset surfaces in computer aided design [18, 37]. Let Ω_0 be a regular surface with parameterization $x_0(u,v) : U \subset \mathbb{R}^2 \to \mathbb{R}^3$ with unit normal $n_0(u,v)$, then the parallel or offset surface of distance a, which we will denote Ω_{0+a}, is defined by

$$x_a(u,v) = x_0(u,v) + an_0(u,v). \tag{28}$$

Note that the exterior a-level set is a subset of the parallel/offset surface of distance a: $\Omega_a \subseteq \Omega_{0+a}$, which we prove in Lemma F.7 **a**.

Let the principal curvatures at $x_0(u,v)$ be $\kappa_{0,1}(u,v)$ and $\kappa_{0,2}(u,v)$, with respective (unit) principal directions. This means that any curve on the surface Ω_0 that passes

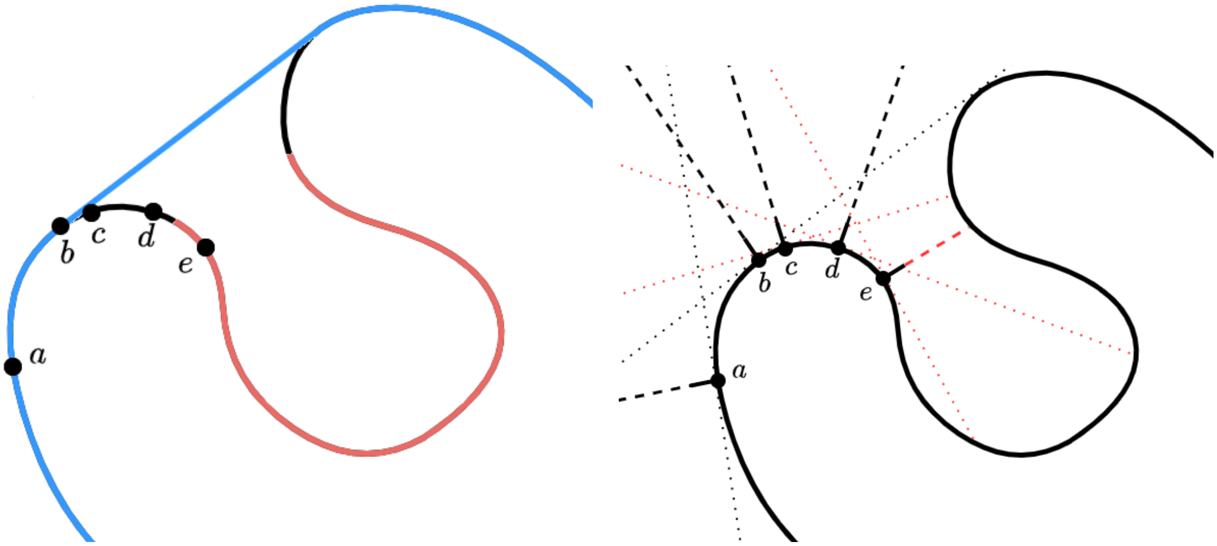

Figure 12. A shape (shown by the black boundary) with 5 points $a - e$ identified. Left: The convex hull is shown in blue, and the SION surface point (relative to a surrounding sphere) are shown in red. Right: the normal direction and tangent lines are shown for each point. The normal direction is red if it is a SION point, and the tangent line is red if it is not supporting. See Appendix F.6 for the case each point belongs to.

through the point parameterised by u, v in the i^{th} principal direction is well approximated by a circle of radius $r_{0,i}(u,v) = \left|\frac{1}{\kappa_{0,i}(u,v)}\right|$.

It turns out that $x_a(u,v)$ not only has the same normal direction $n_a(u,v) = n_0(u,v)$, but it also has the same principal directions and center for the curves in each principal direction [18], with principal curvatures

$$\kappa_{a,i}(u,v) = \frac{\kappa_{0,i}(u,v)}{1 - a\kappa_{0,i}(u,v)} \tag{29}$$

and principal radii

$$r_{a,i}(u,v) = \left|\frac{1}{\kappa_{a,i}(u,v)}\right| \tag{30}$$

$$= \begin{cases} r_{0,i}(u,v) + a & \kappa_{a,i}(u,v) < 0 \\ r_{0,i}(u,v) - a & \kappa_{a,i}(u,v) > 0 \end{cases}. \tag{31}$$

F.4. Important Definitions and Results

We first show that the exterior a-level set Ω_a is a subset of the parallel surface Ω_{0+a}. This allows us to use the local parameterization $x_a(u,v)$ of Ω_{0+a} for Ω_a. Note that the local parameterization $x_a(u,v)$ inherits the smoothness from $x_0(u,v)$, however Ω_{0+a} might not be regular as the offset can cause the parameterization to not be one-to-one and the partial derivatives to not be linearly independent. Furthermore, $F_\mathcal{V}$ may not be smooth at all points on Ω_{0+a} and thus Ω_a.

Lemma F.7. Assume $F_\mathcal{V}$ is an SDF with a smooth zero level set Ω_0. Then for all $a \geq 0$,

a. for all $p \in \Omega_a$, we have $p = x + an(x)$ for any $x \in \Omega_0$ that is a closest point to p. Thus $\Omega_a \subseteq \Omega_{0+a}$.

b. for any $p \in \Omega_a$ that has a unique closest point $x \in \Omega_0$, $F_\mathcal{V}$ is differentiable at p, with $p = x + an(x)$ and $n(p) = n(x)$.

c. Ω_a is piecewise smooth, i.e., the set of points $p \in \Omega_a$ for which $F_\mathcal{V}$ is not differentiable is a set of measure zero relative to the surface Ω_a.

Proof.

a. Consider $p \in \Omega_a$ and let $x \in \Omega_0$ be a closest point in Ω_0 to p (so $\|p - x\| = a$). Let us assume that $p - x = ad$ where $\|d\| = 1$ and $d \neq n(x)$. Then by the smoothness of Ω_0, $\exists \hat{x} \in \Omega_0$ such that $\hat{x} = x + \delta d'$, where $d' \perp n(x)$ and $\langle d', d \rangle > 0$. Thus $\hat{x}$ lies on the halfspace $\{z \mid \langle z, d \rangle > \langle x, d \rangle\}$, so $\|\hat{x} - p\| < a$. This is a contradiction, as $d(p, \Omega_0) = a$.

Note this further implies that $\Omega_a \subseteq \Omega_{0+a}$.

b. Follows from the SDF properties listed in Appendix F.2.

c. We want to show that the set of points $p \in \Omega_a$ for which $F_\mathcal{V}$ is not differentiable is a set of measure zero. Let us assume there is some measurable subset $A \subset \Omega_a$ where $F_{\mathcal{V}}$ is not differentiable. Then for any point $p \in A$, there exists at least two different closest points in Ω_0 to p, $x_1, x_2, ..., x_k$. Then consider $p' = p + \varepsilon \frac{n(x_1)+n(x_2)+...+n(x_k)}{k}$.

Note that $x_1, x_2, ..., x_k$ must be closest points to p' for some small epsilon (as otherwise p must have another closest point) and $\frac{n(x_1)+n(x_2)+...+n(x_k)}{k} \neq 0$ as then p cannot be on the exterior. However doing this for all $p \in A$ would create a set for which $F_{\mathcal{V}}$ is not differentiable but have non-zero measure within $\mathcal{D}$, which cannot be the case by the SDF properties we listed earlier. Thus no such A can exists.

□

Definition F.8. A point $p \in S$ has a *supporting hyperplane* for the set S if there exists a hyperplane P such that $p \in P$ and S is entirely contained within one of the two closed half-spaces bounded by the hyperplane (note that p does not need to be the only point in the intersection of the supporting hyperplane and S).

By the supporting hyperplane theorem, if a set is nonempty and convex then every point on the boundary of the set has a supporting hyperplane, and if if a set is closed, has nonempty interior and has a supporting hyperplane at each point on its boundary then it is convex [8].

Definition F.9. Let S be a surface. If a point $p \in S$ has a well defined normal, then its *tangent hyperplane* is the unique hyperplane P such that $x \in P$ and for any $y \in P$, $y - x \perp n(x)$ (or $s(y - x, n(x)) = 0$). Furthermore we define the *closed interior tangent half space* as the closed half space bounded by the hyperplane that $n(x)$ faces away from: $\{z \in \mathcal{D} \mid s(z - x, n(x)) \leq 0\}$, and the *open exterior half space* as the open half space bounded by the hyperplane that $n(x)$ faces towards: $\{z \in \mathcal{D} \mid s(z - x, n(x)) > 0\}$.

Note that if a point on a surface has a well defined normal and has a supporting hyperplane, then its supporting hyperplane is unique and is its tangent hyperplane [40].

Next we wish to compare exterior a-level sets to the boundary of their convex hull $\partial\mathbf{conv}(\Omega_a)$ (note $\partial\mathbf{conv}(\Omega_a) \subset \mathbf{conv}(\Omega_a)$ as Ω_a is compact). In particular we define the following three sets:

- $\mathcal{A}_a = \partial\mathbf{conv}(\Omega_a) \cap \Omega_a$ is the set of points in Ω_a that coincide with the boundary of the convex hull. These are the points on the level set that have a supporting hyperplane.
- $\mathcal{B}_a = \partial\mathbf{conv}(\Omega_a) \setminus \Omega_a$ is the set of points on the boundary of the convex hull that are not part of the level set.
- $\mathcal{C}_a = \Omega_a \setminus \partial\mathbf{conv}(\Omega_a)$ is the set of points on the level set that are not part of the boundary of the convex hull. These are the points on the level set that do not have a supporting hyperplane.

Finally we prove some useful results about convex hulls and convex hulls of exterior a-level sets.

Lemma F.10. $\mathcal{A}_a$ is a parallel/offset surface of $\mathcal{A}_0$, i.e., $\mathcal{A}_a = \mathcal{A}_{0+a}$. Furthermore every point $p \in \mathcal{A}_a$ has a well defined normal and thus its supporting hyperplane is unique and is its tangent hyperplane.

Proof. We first show that $\mathcal{A}_{0+a} \subseteq \mathcal{A}_a$. Let $x \in \mathcal{A}_0$. Note that its tangent hyperplane is its unique supporting hyperplane. Now $x + an(x) \in \Omega_a$ as if $d(x + an(x), \Omega_0) < a$ then x cannot have a supporting hyperplane. Furthermore the hyperplane tangent to $n(x)$ at $x + an(x)$ must be a supporting hyperplane for Ω_a at $x + an(x)$ as translating points from Ω_0 by distance a cannot take them to the other side of that hyperplane. Thus $x + an(x) \in \mathcal{A}_a$.

We now show that $\mathcal{A}_a \subseteq A_{0+a}$. Let $p \in \mathcal{A}_a$. Clearly $p = x + an(x)$ for some $x \in \Omega_0$. However if $x \in \mathcal{B}_0$, then there is some $z \in \mathcal{A}_0$ such that $n(z) = n(x)$ and $\langle n(x), z - x\rangle > 0$. Furthermore $z + an(z) \in \Omega_a$ and clearly $n(z + an(z)) = n(x + an(x))$ and $\langle n(x + an(x)), z + an(z) - x + an(x)\rangle > 0$, so $p = x + an(x)$ cannot be in $\mathcal{A}_a$. Thus $x \in \mathcal{A}_0$, so we have shown that $\mathcal{A}_a$ is a parallel/offset surface of $\mathcal{A}_0$.

Finally, to show that every point $p \in \mathcal{A}_a$ has a well defined normal we need to show that p has a unique closest point to Ω_0. Assume that p has two closest points $x, y \in \Omega_0$. Thus $p = x + an(x) = y + an(y)$ and $x \neq y$ and $n(x) \neq n(y)$. However as proven above, $x, y \in \mathcal{A}_0$. Thus since p has a supporting hyperplane, that same hyperplane translated must be supporting at x and y. However x and y have well defined normals, so their supporting hyperplanes are unique and must be their tangent hyperplanes, which are different. This is a contradiction. Thus every point $p \in \mathcal{A}_a$ has a well defined normal and thus its supporting hyperplane is unique and is its tangent hyperplane. □

Lemma F.11. Let p be a convex combination of points $p_1, \ldots, p_n \in \mathcal{A}_a$ that all lie on the same hyperplane and have the same normal direction that is orthogonal to that hyperplane. Then either $p \in \mathcal{A}_a$ or $p \in \mathcal{B}_a$. Moreover, any $p \in \mathcal{B}_a$ can be written as such a convex combination.

Proof. Let p be a convex combination of points $p_1, \ldots, p_n \in \mathcal{A}_a$ that all lie on the same hyperplane and have the same normal direction that is orthogonal to that hyperplane. This means that the hyperplane is their shared tangent hyperplane, which must be a supporting hyperplane for $\mathbf{conv}(\Omega)$ at each of those points. Now p lies on the hyperplane as the convex combination cannot take it off the hyperplane, so the hyperplane is a supporting hyperplane for p as well. Thus p is on the boundary of the convex hull, so $p \in \partial\mathbf{conv}(\Omega_a)$, which means it is in $\mathcal{A}_a$ or $\mathcal{B}_a$.

Now let $p \in \mathcal{B}_a$. As $p \in \partial\textbf{conv}(\Omega_a)$ and $\partial\textbf{conv}(\Omega_a) \subset \textbf{conv}(\Omega_a)$ (since Ω_a is compact), by the Krein-Milman theorem p is a convex combination of points $p_1, ..., p_n \in \mathcal{A}_a$ (since A_a contains the extreme points of Ω_a). However as $p \in \partial\textbf{conv}(\Omega_a)$, there is a supporting hyperplane for $\textbf{conv}(\Omega_a)$ at p, so it follows that $p_1, ..., p_n$ lie on that same supporting hyperplane. This means that the hyperplane is a supporting hyperplane for $\textbf{conv}(\Omega_a)$ at $p_1, ..., p_n$ as well. However by Lemma F.10 $p_1, ..., p_n$ have a unique supporting hyperplane given by their tangent hyperplane, so $p_1, ..., p_n$ have the same normal direction that is orthogonal to the supporting hyperplane. □

Lemma F.12. We have that $\mathcal{A}_a$ is a parallel/offset surface of $\mathcal{A}_0$, $\mathcal{B}_a$ is a parallel/offset surface of $\mathcal{B}_0$, and $\partial\textbf{conv}(\Omega_a)$ is a parallel/offset surface of the convex hull of the 0-level set $\partial\textbf{conv}(\Omega_0)$. The latter implies that $\partial\textbf{conv}(\Omega_a) = \{x + an(x) \mid x \in \partial\textbf{conv}(\Omega_0)\}$.

Proof. From Lemma F.10, $\mathcal{A}_a$ is a parallel/offset surface of $\mathcal{A}_0$.

To show that $\mathcal{B}_a$ is a parallel/offset surface of $\mathcal{B}_0$, note that by Lemma F.11 for any $x \in \mathcal{B}_0$ we have that x is a convex combination of points $x_1, ..., x_n \in \mathcal{A}_0$ that lie on the same hyperplane and have the same normal direction, and similarly for any $p \in \mathcal{B}_a$ we have that p is a convex combination of points $p_1, ..., p_n \in \mathcal{A}_a$ that lie on the same hyperplane and have the same normal direction. Thus as we have shown that $\mathcal{A}_a$ is a parallel/offset surface of $\mathcal{A}_0$, offsetting $x_1, ..., x_n$ gives $p_1, ..., p_n$ that must lie on a hyperplane that is parallel to the hyperplane $x_1, ..., x_n$ lie on, and they all have the same normal. Thus

$$\begin{aligned} x \in \mathcal{B}_0 &\iff x = c_1x_1 + ... + c_nx_n \\ &\qquad\qquad \text{(where } x_1, .., x_n \text{ obey Lemma F.11)} \\ &\iff x + an(x) = c_1(x_1 + an(x_1)) + .. \\ &\qquad + c_n(x_n + an(x_n)) \quad (\text{as } c_1 + ... + c_n = 1) \\ &\iff x + an(x) = c_1p_1 + ... + c_np_n \\ &\qquad\qquad (p_1, ..., p_n \in \mathcal{A}_{0+a} = \mathcal{A}_a) \\ &\iff x + an(x) \in \mathcal{B}_a. \\ &\qquad\qquad \text{(as } p_1, .., p_n \text{ obey Lemma F.11)} \end{aligned}$$

Finally, since Ω_a is the disjoint union of $\mathcal{A}_a$ and $\mathcal{B}_a$ for any $a \geq 0$, clearly $\partial\textbf{conv}(\Omega_a)$ is a parallel/offset surface of the convex hull of the 0-level set $\partial\textbf{conv}(\Omega_0)$. □

F.5. Proof of the Four Propositions.

F.5.1 Proof of Statement 1's Proposition

Lemma F.13. Let $x \in \Omega_0$. If $q = x + an(x)$ is a SION point for Ω_a relative to a surrounding sphere, then the first point of intersection, $p = x + kn(x) \in \Omega_a$ where $k > a$, must have that $p = x' + an(x')$ for a point $x' \in \Omega_0$ where **(1)** x' is in the open exterior half space at x, and **(2)** $s(n(x), n(x')) \leq 0$.

Proof. As $p \in \Omega_a$, $p = x' + an(x')$ for some point $x' \in \Omega_0$ by Lemma F.7.

To see (1), note that $p = x' + an(x') = x + kn(x)$ where $k > a$ (see Fig. 13 top left), so the closed ball of radius a at p (which contains x') must be contained within the open exterior tangent half space at x (see Fig. 13 top right).

To see that $s(n(x), n(x')) \leq 0$, note that as $x' + an(x') = x + kn(x)$ where $k > a$, then for any $0 < t < k$ $x + (k - t)n(x)$ must be outside Ω_0. This means that for some small $t > 0$, $x + (k - t)n(x)$ must be in the open exterior tangent half space at $x' + an(x')$ (see Fig. 13 bottom left), so $s(n(x), n(x')) \leq 0$ (as $s(n(x), n(x')) > 0$ would imply that the ray $x + \lambda n(x)$ hits x' from the interior, rather than the exterior). □

We now give the proof for Proposition F.3 which we restate here

Proposition F.3. For any sphere $\mathcal{R}$ surrounding Ω_a, SION(Ω_a; $\mathcal{R}$) is zero for all $a > \alpha$, where α depends only on Ω_0. Furthermore, $\alpha \leq d_{\partial\mathcal{V}}$.

Proof. Let $x \in \Omega_0$. If $x + an(x)$ is a SION for some $a \geq 0$, then $p = x' + an(x') = x + kn(x)$ where $k > a$ for some $x' \in \Omega_0$, and by the previous lemma x' is in the open exterior tangent half space at x and $s(n(x), n(x')) \leq 0$. Thus x, x', p form a triangle where $d(x, p) = k$, $d(x', p) = a$ and the vertex at p has an angle $\geq 90°$ (see Fig. 13 bottom right). This means that x, x' is the largest side so $a = d(x', p) < d(x, x') \leq d_{\partial\mathcal{V}}$ and $k = d(x, p) < d(x, x') \leq d_{\partial\mathcal{V}}$. Furthermore, note that $x + kn(x) \notin \Omega_k$ as its distance to Ω_0 is less than or equal to a.

Thus when $a \geq d_{\partial\mathcal{V}}$, it follows that SION($\Omega_a$; $\mathcal{R}$) $= 0$ where $\mathcal{R}$ is any sphere surrounding Ω_a. □

F.5.2 Proof of Statement 2's Proposition

We now give the proof for Proposition F.4 which we restate here

Proposition F.4. As a increases, Ω_a limits towards the exterior a-level set of $\mathcal{A}_0 = \Omega_0 \cap \partial\textbf{conv}(\Omega_0)$. In particular for any a, exterior a-level set Ω_a does not depends on points in Ω_0 that are further than $\frac{d_{\partial\mathcal{V}}}{2a}$ to $\partial\textbf{conv}(\Omega_0)$.

Proof. Let $x \in \Omega_0$. If $x \in \mathcal{A}_0$ then $p = x + an(x) \in \Omega_a$ for any a. If $x \in \mathcal{C}_0$, then we can determine an upper bound on a for when p is no longer within Ω_a. Let $z \in \Omega_0$ be the point on the shape's surface furthest away from the tangent hyperplane at x in the normal direction of x, $z = \arg\max_{y \in \Omega_0} \langle n(x), y - x \rangle$, and set $\Delta_x = \langle n(x), z - x \rangle$ (see Fig. 14 left). Note that $\langle n(x), z - x \rangle > 0$ and $\Delta_x > 0$ since $x \in \mathcal{C}_0$ implies that x does not have a tangent supporting hyperplane, and $z \in \mathcal{A}_0$ as otherwise there is a point further

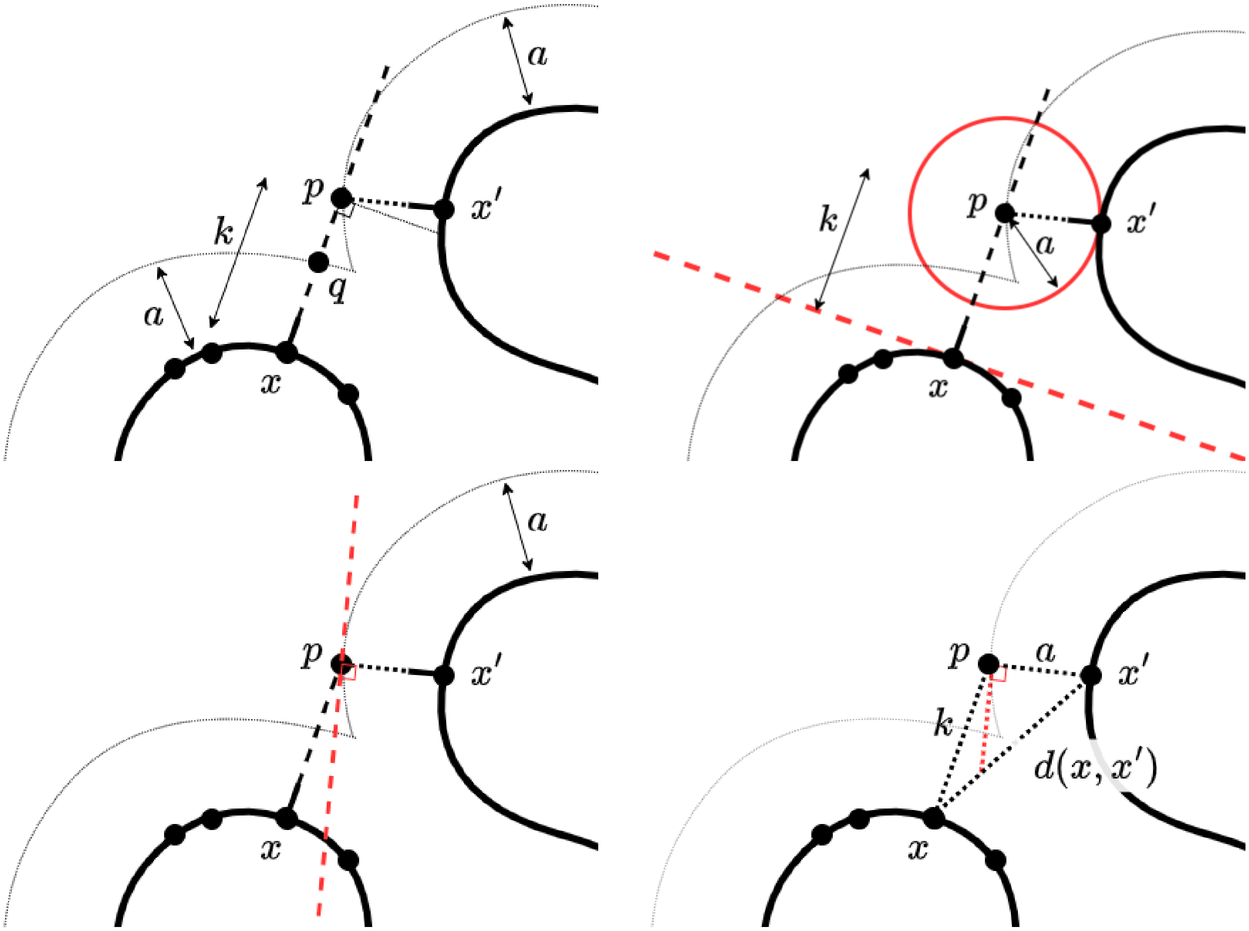

Figure 13. The cases for Statement 1.

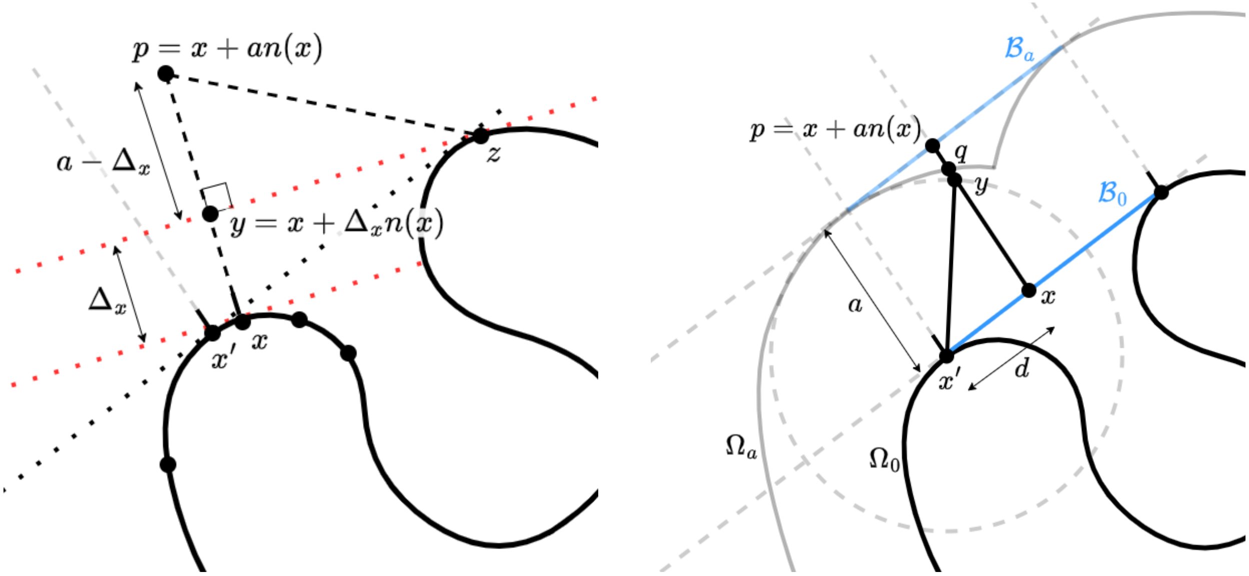

Figure 14. The cases for Statements 2 (left) and 3 (right). Note for ease of illustration, the case for Statement 3 (right) has been drawn with $a \leq d_{\partial\mathcal{V}}$.

in the normal direction. Then p is closer to z than x when

$$
\begin{aligned}
d(p,x) &= a && (32)\\
&> d(p,z) && (33)\\
&= \sqrt{d(p,y)^2 + d(y,z)^2} && (34)\\
&= \sqrt{(a-\Delta_x)^2 + \sqrt{d(x,z)^2 - d(x,y)^2}^2} && (35)\\
&= \sqrt{(a-\Delta_x)^2 + d(x,z)^2 - \Delta_x^2} && (36)\\
&= \sqrt{d(x,z)^2 + a^2 - 2a\Delta_x} && (37)
\end{aligned}
$$

which is equivalent to $a > \frac{d(x,z)^2}{2\Delta_x}$. Now $d(x,z)$ is bounded above by $d_{\partial\mathcal{V}}$, and $d(x,\mathbf{conv}(\Omega_0)) \leq \Delta_x$ since $z \in \mathcal{A}_0$ and y lies on its supporting hyperplane so the line segment between x and y has to cross $\mathbf{conv}(\Omega_0)$. Thus rearranging we get that $d(x,\mathbf{conv}(\Omega_0)) < \frac{d_{\partial\mathcal{V}}}{2a}$. □

F.5.3 Proof of Statement 3's Proposition

We now give the proof for Proposition F.5 which we restate here

Proposition F.5. As $a \to \infty$, the Hausdorff distance between Ω_a and the boundary of its convex hull $\partial\mathbf{conv}(\Omega_a)$ limits to 0. In particular, for any $a > d_{\partial\mathcal{V}}$ the Hausdorff distance is bounded by $a - \sqrt{a^2 - d_{\partial\mathcal{V}}^2}$.

Proof. Let us restrict $a > d_{\partial\mathcal{V}}$. We will bound where exactly Ω_a can be. Let $q \in \Omega_a$. Note that $q \in \mathbf{conv}(\Omega_a)$. However as we have specified $a > d_{\partial\mathcal{V}}$, if $q \in \Omega_a$ then $q \notin \mathbf{conv}(\Omega_0)$. Thus $q \in \partial\mathbf{conv}(\Omega_a) \setminus \partial\mathbf{conv}(\Omega_0)$. By Lemma F.12, as $\partial\mathbf{conv}(\Omega_0)$ and $\partial\mathbf{conv}(\Omega_a)$ are parallel/offset surfaces, this means that $q = x + kn(x)$ for some $x \in \Omega_0$ and some $k > 0$. However if $x \in \mathcal{A}_0$, then it must be that $k = a$ (as otherwise $x + an(x) \notin \Omega_a$). If $x \in \mathcal{B}_0$, then $q \in \mathcal{C}_0$ (as otherwise by the Lemma $x \in \mathcal{A}_0$) and $k < a$ (as $x \in \mathcal{B}_0$ implies $x + an(x) \notin \partial\mathbf{conv}(\Omega_a)$). Let $p = x + an(x)$, note by the Lemma $p \in \partial\mathbf{conv}(\Omega_a)$. We want want to determine where on the line segment between x and p the point q can and cannot be (see Fig. 14 (right)). Let x' be the closest point to x within $\mathcal{A}_0$, and consider a sphere of radius a centered at x'. Note that x is inside this sphere as $a > d_{\partial\mathcal{V}}$, so the surface of the sphere intersects with the line segment between x and p at some point y. By the definition of $\mathcal{C}_a$, q cannot be within the sphere as then its distance to x' would be less than a. Thus we have bounded q to be within the line segment between y and p. The length of this line segment is

$$
\begin{aligned}
d(p,y) &= d(p,x) - d(y,x) && (38)\\
&= a - \sqrt{d(y,x')^2 - d(x',x)^2} && (39)\\
&= a - \sqrt{a^2 - d(x',x)^2} && (40)\\
&\leq a - \sqrt{a^2 - d_{\partial\mathcal{V}}^2}. && (41)
\end{aligned}
$$

Thus we have determined that any $q \in \Omega_a$ is either in $\mathcal{A}_a \subset \partial\mathbf{conv}(\Omega_a)$ or is within $a - \sqrt{a^2 - d_{\partial\mathcal{V}}^2}$ of $\mathcal{B}_a \subset \partial\mathbf{conv}(\Omega_a)$ (when $q \in \mathcal{C}_a$). Note that this is also a bound on the distance to $\mathcal{C}_a$ for every $p \in \mathcal{B}_a$ as well by the same construction, thus it is a bound on the Hausdorff distance. Since $a - \sqrt{a^2 - d_{\partial\mathcal{V}}^2} \to 0$ as $a \to \infty$, it follows that the Hausdorff distance between Ω_a and $\partial\mathbf{conv}(\Omega_a)$ limits to 0 as $a \to \infty$. □

F.5.4 Proof of Statement 4's Proposition

We now give the proof for Proposition F.6 which we restate here

Proposition F.6. As a increases the principal curvatures of Ω_a either stay zero or limit to $-\frac{1}{a}$, which is the curvature of a sphere of radius a (equivalently the radii of the principal curvatures either stay ∞ or limits to a). Furthermore as $a \to \infty$, exterior a-level set Ω_a limits to a sphere.

Proof. Let $p \in \Omega_a$. Then $p = x + an(x)$ for some $x \in \Omega_0$ by Lemma F.7. Let the principal curvatures at p and x be $\kappa_i(p)$ and $\kappa_i(x)$. From Section F.3 the principal curvatures at p are given by $\kappa_i(p) = \frac{\kappa_i(x)}{1 - a\kappa_i(x)}$. Thus if $\kappa_i(x) = 0$ then $\kappa_i(p) = 0$, and if $\kappa_i(x) \neq 0$ then $\kappa_i(p) = \frac{1}{\frac{1}{\kappa_i(x)} - a}$ which limits to $-\frac{1}{a}$ as $a \to \infty$.

To see that Ω_a limits to a sphere, note that as $a \to \infty$ the distance from any $p \in \Omega_a$ to any $x \in \Omega_0$ approaches a. □

F.6. Summary

To somewhat summarise out findings, for each $x \in \Omega_0$ let us define $E(x)$ as the set of points $x' \in \Omega_0$ that are in the open exterior half space to x. Then

- **If $E(x)$ is empty:**
  - x has a tangent supporting hyperplane to Ω_0, and thus $x \in \mathcal{A}_0$. Furthermore for any $a \geq 0$, $x + an(x) \in \mathcal{A}_a \subset \Omega_a$.
  - For any $a \geq 0$, clearly $x + an(x)$ is not a SION point for Ω_a.

  Points a and b in Fig. 12 are examples of this case.
- **If $E(x) \neq \emptyset$ and $\exists x' \in E(x)$ such that $s(n(x), n(x')) \leq 0$:**
  - x does not have a tangent supporting hyperplane so $x \in \mathcal{C}_0$.
  - There exists some $a \geq 0$ such that $x + an(x) \in \Omega_a$ and is a SION point for Ω_a.
  - There exists $\alpha > 0$ such that for all $a \geq \alpha$ $x + an(x) \notin \Omega_a$.

  Points d and e in Fig. 12 are examples of this case (note e is a SION for $a = 0$ but d is not).
- **If $E(x) \neq \emptyset$ and $x' \in E(x) \implies s(n(x), n(x')) > 0$:**
  - x does not have a tangent supporting hyperplane to Ω_0 so $x \in \mathcal{C}_0$.

- For any $a \geq 0$, $x + an(x)$ will never be a SION point for Ω_a.
- There exists some $\alpha > 0$ such that for all $a > \alpha$ we have that $x + an(x) \notin \Omega_a$, and α is bounded above by $\frac{d_{\partial \mathcal{V}}}{2d(x, \partial \mathbf{conv}(\Omega_0))}$.

Point c in Fig. 12 is an example of this case.

G. More visualizations

We show visualizations in Fig. 15 and Fig. 16 using the colourmap settings from OG-INR [25]. Note this visualization works in the favour of our method, our method is highly unlikely to produce outside ghost geometry and thus have outside red surfaces in the coloured mesh. On the other hand, our main failure cases are missing surfaces, which this visualization does not show, or extra inside surfaces, which would be occluded. These both can happen when the estimated sampling radius is not conservative enough and guiding points enter the inside of the shape. The former can also happen when the shape has very thin surfaces that are hard for all methods to model, for example the railing of the ship at the bottom of Fig. 16.

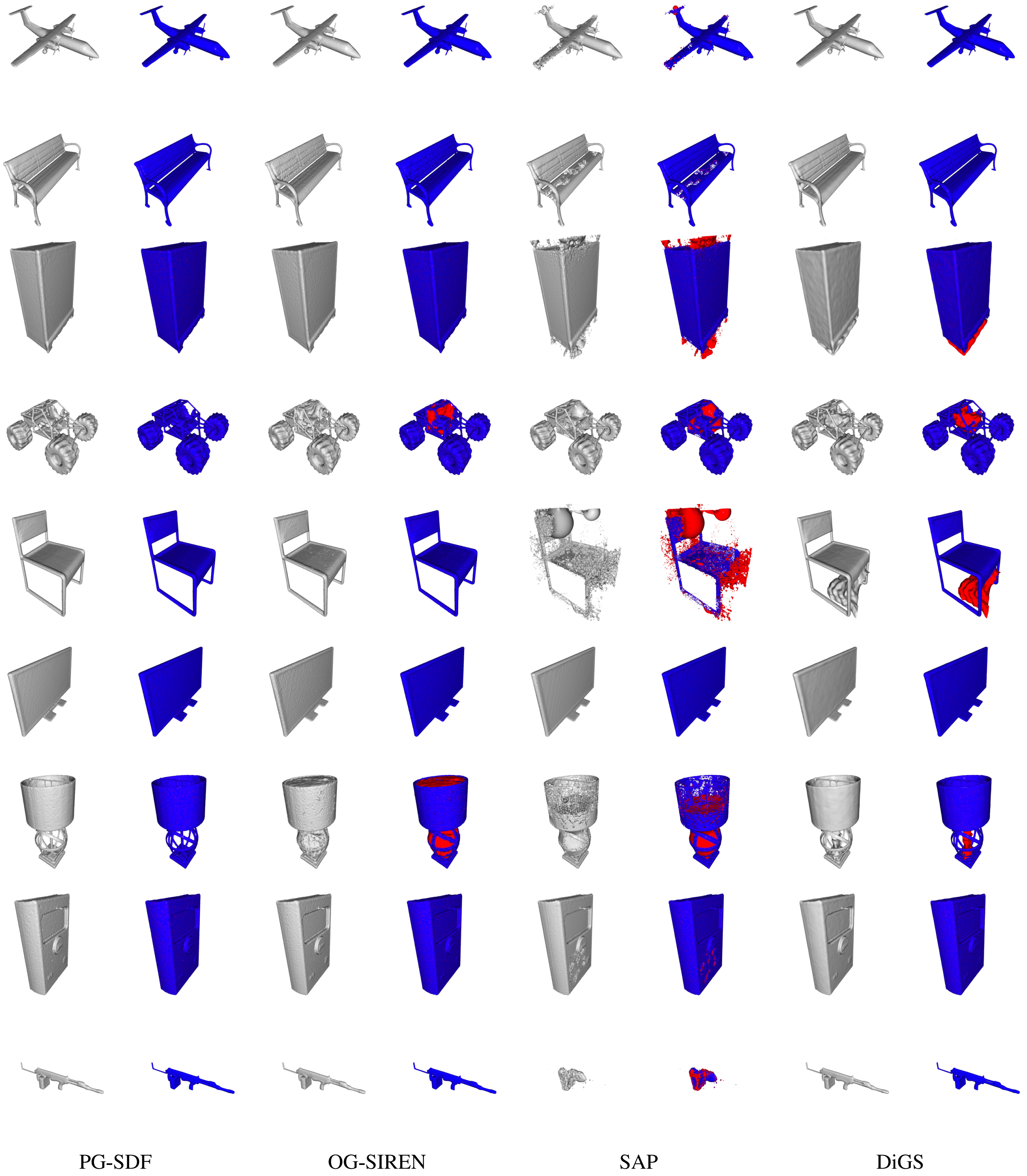

Figure 15. Qualitative comparison on one shape from each class in ShapeNet. We display the reconstructed mesh, and the mesh coloured by the distance of each vertex to the input point cloud (red is higher distance).

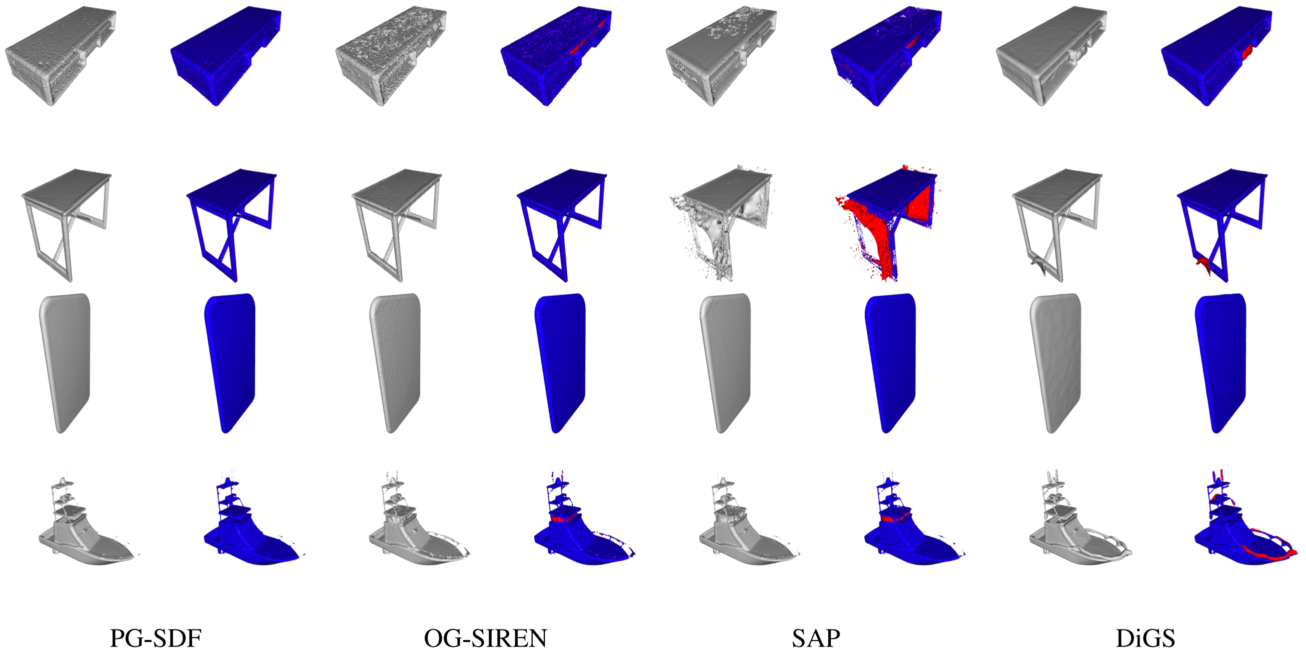

Figure 16. Qualitative comparison on one shape from each class in ShapeNet. We display the reconstructed mesh, and the mesh coloured by the distance of each vertex to the input point cloud (red is higher distance).