DiGS : Divergence guided shape implicit neural representation for unoriented point clouds

A. Supplementary Material

Here we provide supplementary material for the proposed divergence guided shape implicit neural representation. In Section A.1 we discuss SDF properties, theory behind the proposed divergence constraint, and a second order supervised constraint. In Section A.2 we provide proofs and additional visualuzations for the proposed geometric initializations, as well as visualizations of our overall training procedure. In Section A.3 we provide additional experimental results. Finally, we provide a high resolution video showcasing the performance of our method in multiple scenarios at URL.

A.1. SDF learning theory and the divergence term

A.1.1 SDF properties

As explained in Section 5, current losses enforce two properties of SDFs, the function should be zero on the surface and the gradient of the norm should be one everywhere. In particular they are looking at two scalar fields related to the SDF function: the function itself, and the gradient norm field. On the other hand we also considered two more scalar fields, the divergence and curl of the gradient vector field of the SDF function. In Figure 10 we visualise the ground truth value of these four fields for three 2D shapes. We can see straight away that the curl is zero everywhere (as explained in Section 5), and the gradient norm is one everywhere. The divergence on the other hand has very low magnitude in most areas, spiking sharply at points such as centres, skeletons or corners of the shape, and diffusing quickly from there.

A.1.2 Understanding the divergence constraint

The divergence theorem interpretation. The divergence theorem [31] states that integrating over the outward flow in a volume using a triple integral of the divergence is equivalent to a double integral of the flux through its encapsulating surface.

$$\iiint_V \Delta\Phi(x;\theta)dV = \iint_S \nabla\Phi(x;\theta)\cdot\hat{n}dS \tag{13}$$

To intuitively understand what the divergence at a point means, consider the above equation with the volume V being a ball centered at the point, and taking the limit of the radius to 0. Then the divergence of a point is how much the vector field moves towards that point or away from that point from all directions, where mostly moving towards the point implies positive divergence (often called a sink), mostly moving away from that point implies negative divergence (often called a source), and it being balanced implies zero divergence. Thus a point having low divergence magnitude implies that the direction of the gradients are not changing much around that point. The theorem implies that divergence of a single point is heavily influenced by the surrounding region, so it incorporates a lot of local information of the (gradient) vector field.

Minimising Divergence as Regularisation & the Dirichlet Energy. A common setup in machine learning is to not only optimise for a given loss function, but to penalise model complexity by regularising towards less complex solution. This can be viewed as an Occam's Razor principle, where simpler solutions are often better explanations/predictions. We can quantify the complexity of the SDF we have solved for using the Dirichlet Energy. The Dirichlet Energy of a function Φ over a space Ω gives a notion for how smooth variable a function is [11], defined by the convex functional

$$E[\Phi] = \frac{1}{2}\int_\Omega \|\nabla\Phi(x)\|_2^2 dx. \tag{14}$$

Using Green's first identity we have that

$$E[\Phi] = \frac{1}{2}\int_{\partial\Omega}\langle\nabla\Phi(x),\mathbf{n}(x)\rangle\Phi(x)dx - \frac{1}{2}\int_\Omega \Delta\Phi(x)\Phi(x)dx \tag{15}$$

and its functional derivative is

$$DE[\Phi]\Psi = \int_{\partial\Omega}\langle\nabla\Phi(x),\mathbf{n}(x)\rangle\Psi(x)dx - \int_\Omega \Delta\Phi(x)\Psi(x)dx \tag{16}$$

where $\mathbf{n}$ is the outward normal vector to the boundary $\partial\Omega$. Thus to minimise the Dirichlet energy, the functional derivative needs to be 0 for all Ψ. As Ψ is a infinitesimal displacement of Φ, it vanishes on $\partial\Omega$, so we get $\Delta\Phi(x) = 0$. This is Laplace's equation, whose solutions are harmonic function (e.g. steady-state heat equation, which will have a unique solution under sufficiently regular boundary conditions).

However we are more restrictive in the functions Φ we want to minimise $E[\Phi]$ for, specifically we only consider Φ that interpolate our surface points within its zero level set (which can be considered as a boundary condition) and the eikonal equation must hold. As a result we would not be able to solve for a harmonic function, as most SDFs are not harmonic (see Figure 10 for the actual divergence field). However as our functional is convex, to find a function that satisfies our conditions and has minimal Dirichlet

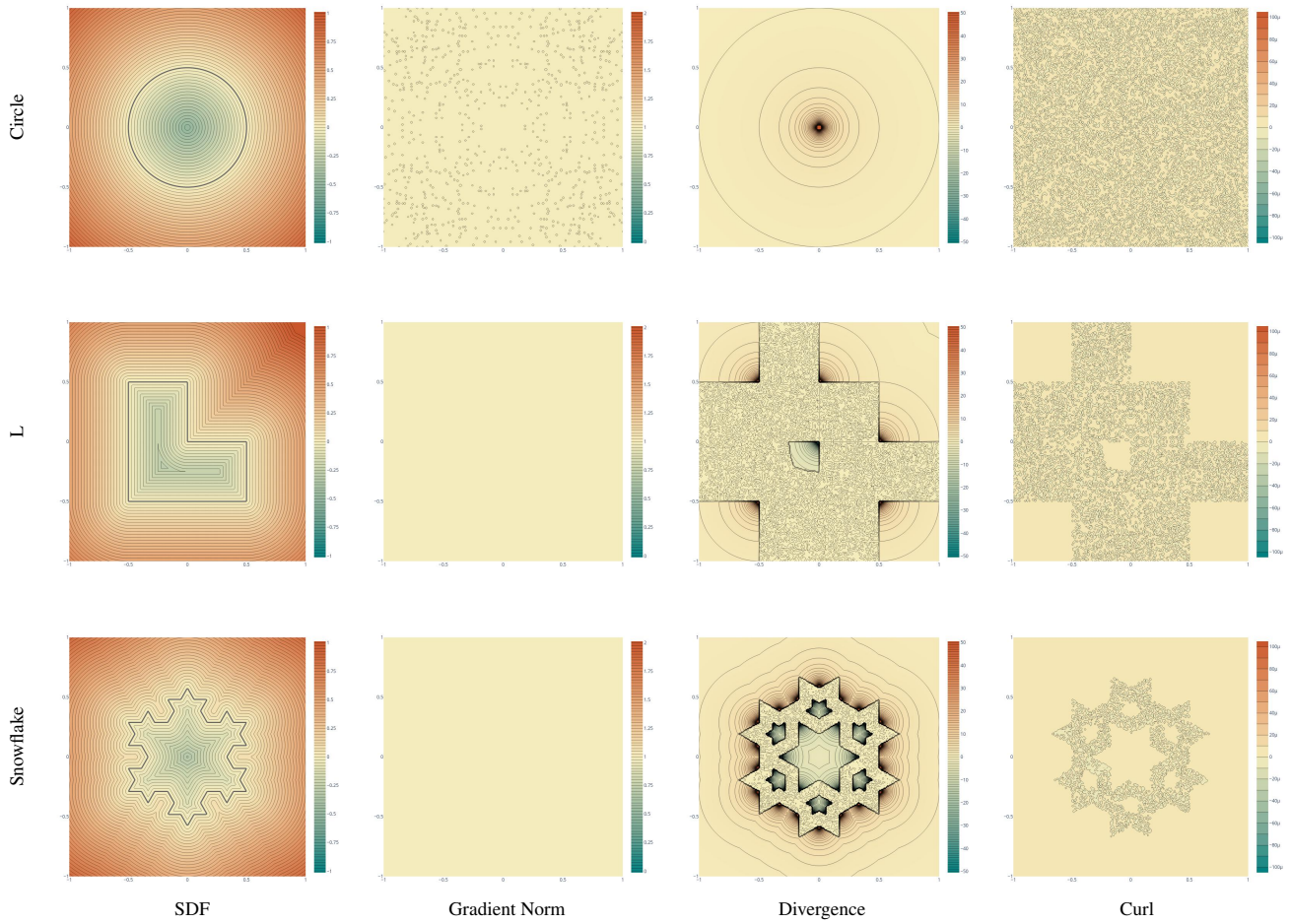

Figure 10. Ground Truth for 2D shapes. We can see that the divergence of the gradient vector field spikes sharply at points such as centres, skeletons or corners of the shape, and diffuses quickly from there.

energy, it suffices to find the function satisfying our conditions whose absolute value of the functional derivative is as small as possible, i.e. minimising our divergence term L_{div} (Equation 10).

Note that this loss term is specifically defined over $\Omega \setminus \Omega_0$, compared to (6) which is defined over Ω. The difference, Ω_0, is a set of measure zero and mathematically does not change much, but it does computationally: to evaluate our losses we have random samples over the space ($\Omega \approx \Omega \setminus \Omega_0$) and samples at ground truth surface points (i.e., heavy sampling on the surface Ω_0). Thus (6) is computed over both sets of samples, while the divergence loss term is only computer over the former. If we computed the divergence loss over the heavily sampled surface points, it would make our surface drastically smooth. In practice we want the opposite of this: a surface with fine detail will have high variability in its SDF near the surface and low variability further away from the surface.

PHASE [26] also use the Dirichlet energy term for regularisation in their method, but motivate it from the Van der Waals-Cahn-Hilliard (WCH) theory for the physical phenomenon of phase transitions, and they do not anneal the loss.

Toy Problem. We give more visualisations for the toy problem discussed in Section 5. The experiment was repeated 20 times for each of the four cases (20x20 grid without divergence, 20x20 grid with divergence, 200x200 grid without divergence and 200x200 grid with divergence), where the same randomly sampled point constraints were used for the four experiments in the same repetition. Figure 11 shows the learned functions for five repetitions. When the divergence term is present, the contour lines are more smooth and the spacing is more uniform, as desired, showing that they do a better job at maintaining the Eikonal equation. When the divergence term is absent, sometimes the sign of the function is considerably incorrect, with negative above $y = 0$ and positive below, showing it is less stable and more variable and thus that our divergence term

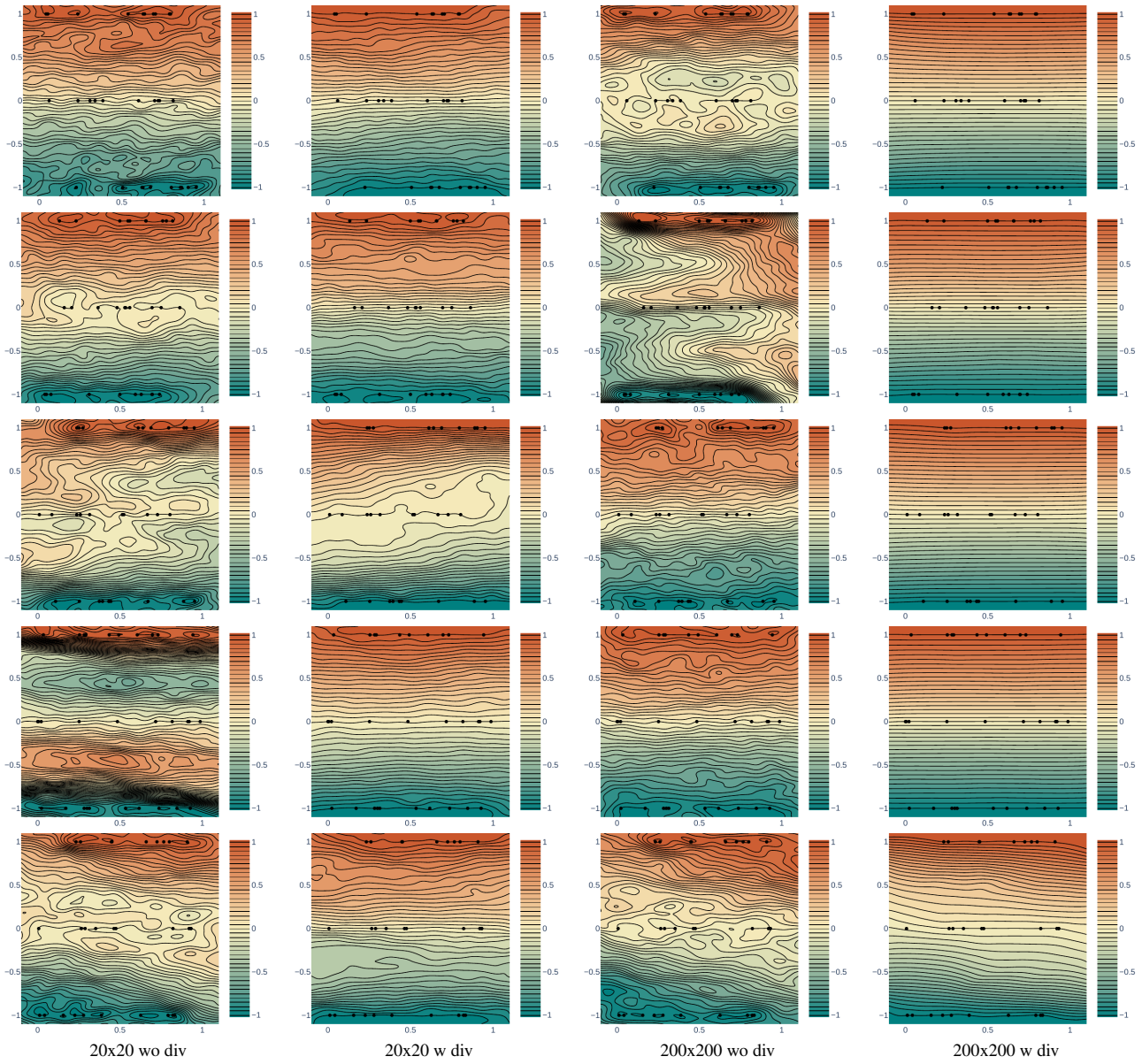

Figure 11. Five repetitions of the toy problem. Contour lines and coloring shows the learned function. Black dots on the lines $y = -1$, $y = 0$ and $y = 1$ show the point constraints. When the divergence term is present, the contour lines are more smooth and the spacing is more uniform, as desired. When the divergence term is absent, sometimes the sign of the function is considerably incorrect, with negative above $y = 0$ and positive below. Notice that when the sampling for the point constraints are not uniform, e.g., the last row where the constraints are almost clustered in a diagonal, the learned function is often biased.

is acting as regularization. Also notice that when the sampling for the point constraints are not uniform, e.g., the last row where the constraints are almost clustered in a diagonal, the learned function is often biased.

A.1.3 Understanding the divergence constraint in 2D

We perform qualitative analysis of the proposed approach on 2D simple shapes: circle, L shape polygon and Koch's snowflake polygon. In this experiment we trained a network with 4 layers and 128 elements in each layer with sine activation functions for 10K epochs, sampling a new set of points in each epoch. We compare to SIREN [37], SIREN without normal vectors, and the proposed DiGS approach with the proposed MFGI initialziation and without it. The results are shown in Figure 12 and Figure 13 where a heatmap is used to visualize the learned distance function, the eikonal term, the divergence, the curl and the dif-

ference between the unsigned predicted distance and the ground truth distance. For the circle shape, incorporating the divergence constraint without decay yields the best result since the circle is a smooth closed shape without fine detail. In contrast, Koch's snowflake is characterised with sharp edges, therefore starting with the divergence constraint and annealing it yields the best performance. In this case the divergence term guides the learning process to a smoothed version of the snowflake, and the annealing allows it to fit the geometry more tightly. SIREN without the normal vectors exhibits ghost geometries (zero level sets that should not appear), while DiGS does not.

The initialization significantly effects the sign of the distance function as well as the model's ability to properly reconstruct fine detail, particularly for the snowflake example.

A.1.4 Second order supervision constraint

Normal vectors are often not available during training and are therefore estimated using some local approximation method [5, 6, 14, 19]. Some methods also provide an approximation of the principal curvatures [5, 14]. The mean curvatures $\kappa_{mean} = \frac{1}{2}(\kappa_1 + \kappa_2)$ provides second order information that can be utilised as supervision for learning shape representation. We propose a supervised variation to DiGS that penalizes points on the surface for having a different mean curvature than the divergence of the vector field using the following constraint:

$$L_{curv} = \int_{\Omega_0} |\Delta\Phi(x;\theta)| - 2\,|\kappa_{mean}|\,dx \tag{17}$$

The new supervised loss is given by:

$$\begin{aligned} L_{DiGS+curv} =& \lambda_A L_A + \lambda_B L_B + \\ & \lambda_{C2} L_{C2} + \lambda_D L_D + \lambda_{curv} L_{curv} \end{aligned} \tag{18}$$

Note that normal and curvature estimations are noisy and highly depend on local neighboring points support size, therefore adding these supervisory signals does not guarantee improved performance.

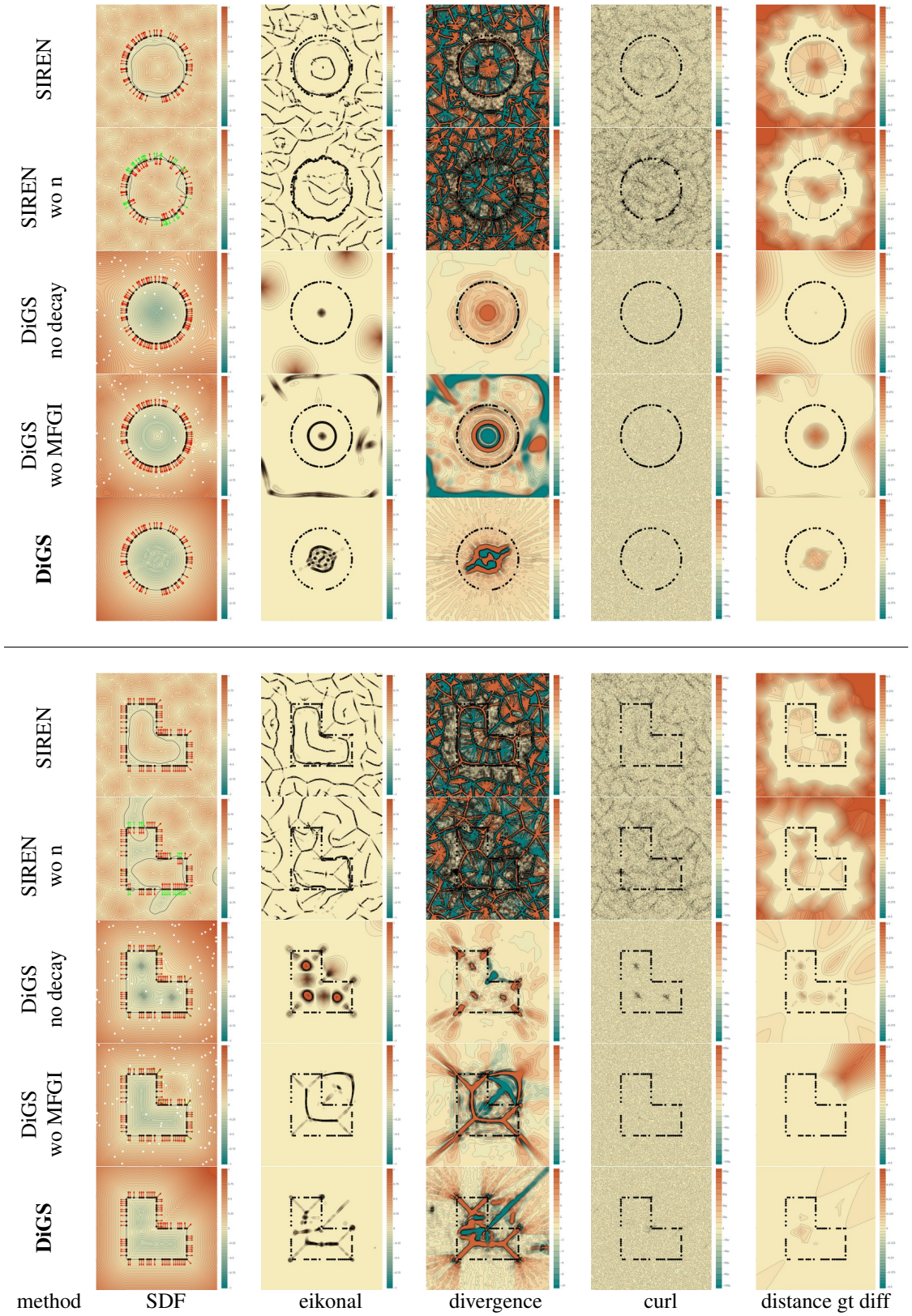

Figure 12. Qualitative 2D results. Visualizing informative quantities as heatmaps over the evaluation space: (left to right) sign distance function, divergence, eikonal term, curl, and distance difference between ground truth and inference for the Circle and L shapes.

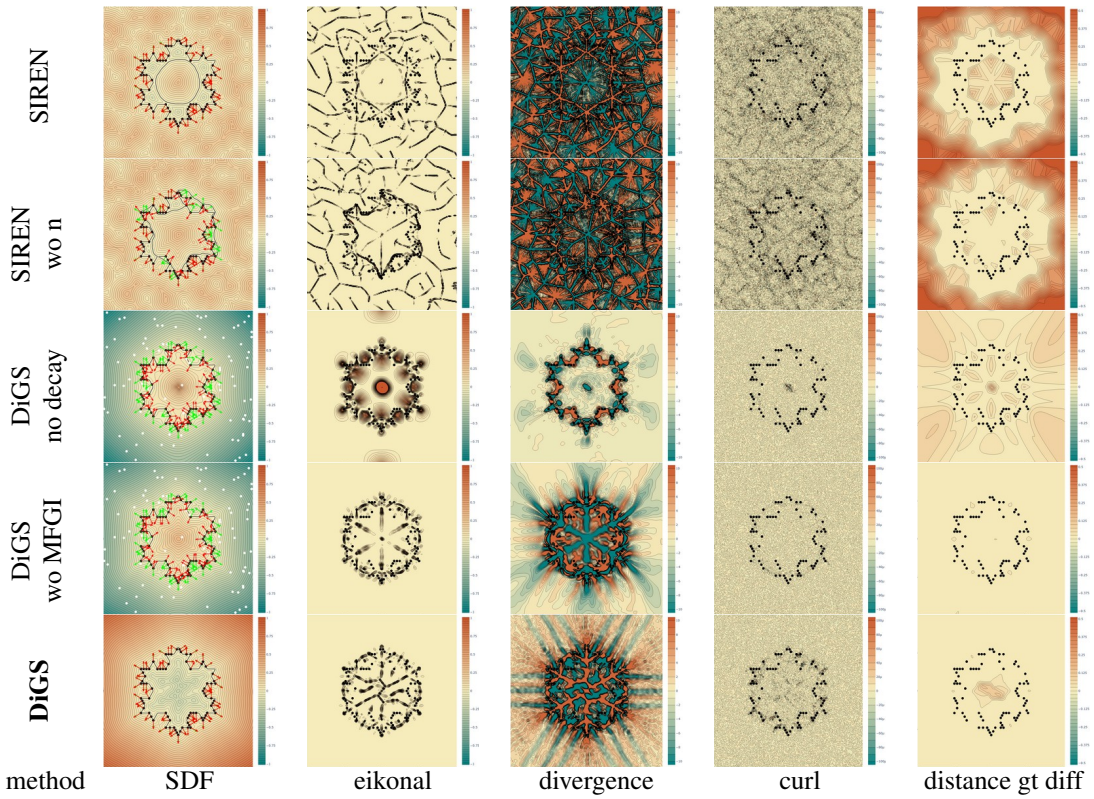

Figure 13. Qualitative 2D results. Visualizing informative quantities as heatmaps over the evaluation space: (left to right) sign distance function, divergence, eikonal term, curl, and distance difference between ground truth and inference for the Koch Snoflake shape.

A.2. Geometric initialization and training procedure

A.2.1 Further initialization visualizations

Following Section 4, we provide a visualization for the proposed geometric initialization with and without multi-frequencies in Figure 14 which shows the SDF, Eikonal term and divergence term. It shows the sphere-like (circle-like in this 2D case) level sets which provide a much smoother Eikonal and divergence terms. We qualitatively compare it to the initialization method proposed by Sitzmann et. al. [37]. This initialization plays a major role in shape space learning.

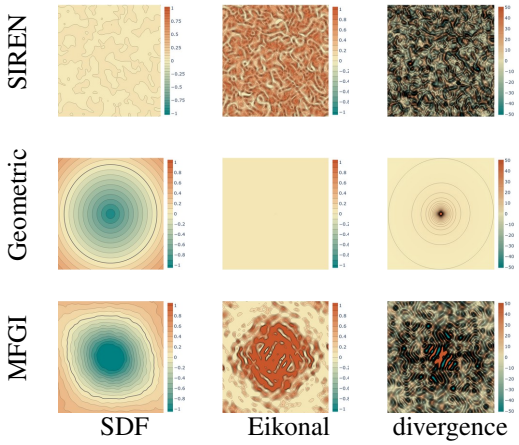

Figure 14. Visualization of the proposed geometric initialization and multi-frequency geometric initialization for sinusoidal representation networks in 2D compared to Sitzmann et. al. [37]. Depicting the sign distance function (left), eikonal (middle) and, divergence (right).

A.2.2 Proofs

The following propositions and proofs show how we initialize our network to a sphere (i.e. the SDF to the function $\Phi(x) = \|x\|_2$). Following Williams et al. [43], rather than roughly approximating the norm with our function class, we instead do a much better approximation to the squared norm, $\Phi(x) = \|x\|_2^2$, (which we do in Equation 22). We then apply the following function to the output of the SIREN:

$$\nu(x) = \textbf{sign}\,(x)\,\sqrt{|x| + \varepsilon}, \tag{19}$$

where $\textbf{sign}\,(x)$ is important as we are learning an SDF, and ε is important for both numerical stability and ensuring the function's derivative is continuous and sub-differentiable. We use $\varepsilon = 10^{-8}$.

Proposition 4.1. *Let Φ be a single hidden layer SIREN ($n = 1$ in Equation 1) of dimension M_n and let x be a point within the unit ball. Set, $\mathbf{w}_n = -\mathbf{1}$, $\mathbf{W}_{n-1} = \frac{\pi}{2} I$, $\mathbf{b}_{n-1} = \frac{\pi}{2}\mathbf{1}$ and $b_n = M_n$. Then, $\nu(\Phi(x)) \approx \|x\|_2$.*

Proof. For 1D input $z \in [-1, 1]$ we can approximate z^2 by $1 - \sin\left(\frac{\pi}{2}z + \frac{\pi}{2}\right)$ (see Figure 15).

Then

$$\Phi(x) = (-\mathbf{1})^T \sin\left(\left(\frac{\pi}{2} I\right) x + \left(\frac{\pi}{2}\mathbf{1}\right)\right) + M_n \tag{20}$$

$$= \sum_{i=1}^{M_n} 1 - \sin\left(\frac{\pi}{2} x_i + \frac{\pi}{2}\right) \tag{21}$$

$$\approx \sum_{i=1}^{M_n} x_i^2 \tag{22}$$

$$= \|x\|_2^2 \tag{23}$$

so $\nu(\Phi(x)) \approx \|x\|_2$. □

To extend this to networks with an arbitrary number of layers, we design layers ϕ_i that preserve the norm on expectation w.r.t. the weights of each layer up to the penultimate layer, i.e., $\mathbb{E}[\|\phi_i(x)\|_2] = \|x\|_2$ for $i = 1, \ldots, n-2$. We first prove the following lemmas.

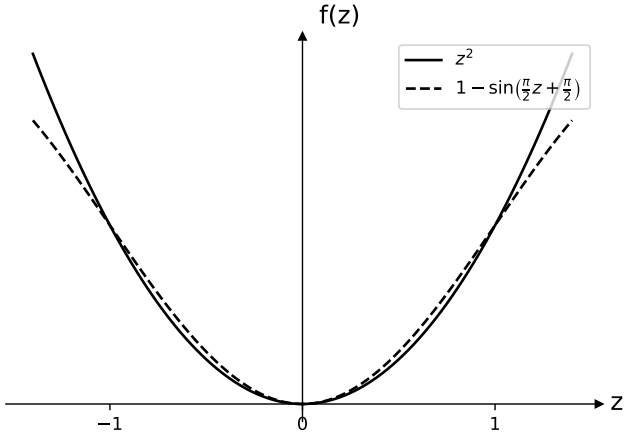

Figure 15. Approximating z^2 using $1 - \sin\left(\frac{\pi}{2}z + \frac{\pi}{2}\right)$.

Lemma A.1. Let X, Y be two random d-dimensional vectors with elements X_i and Y_i sampled i.i.d. from $\mathcal{U}\left(-\sqrt{\frac{3}{d}}, \sqrt{\frac{3}{d}}\right)$. Then $\mathbf{E}[\|X\|] = \mathbf{E}[\|Y\|] = 1$ and $\mathbf{E}[|\langle X, Y\rangle|] = 0$.

In words, random vectors generated i.i.d. from the above distribution are on average unit norm and orthogonal.

Proof. We have that $\mathbf{E}[X_i] = \mathbf{E}[Y_i] = 0$ and $\textbf{var}(X_i) = \textbf{var}(Y_i) = \frac{1}{12}\left(2\sqrt{\frac{3}{d}}\right)^2 = \frac{1}{d}$. Thus note that $\mathbf{E}\left[X_i^2\right] = \mathbf{E}\left[Y_i^2\right] = \mathbf{E}\left[(X_i - \mathbf{E}[X_i])^2\right] = \textbf{var}(X_i) = \frac{1}{d}$. By the

weak law of large numbers we have that for any $\varepsilon > 0$

$$\mathcal{P}\left(\left|\frac{X_1^2 + ... + X_d^2}{d} - \mathbf{E}\left[X_i^2\right]\right| > \varepsilon\right) \leq \frac{\mathbf{var}(X_i^2)}{d\varepsilon^2} \quad (24)$$

$$\therefore \Pr\left(\left|\frac{\|X\|_2^2}{d} - \frac{1}{d}\right| > \varepsilon\right) \leq \frac{\mathbf{var}(X_i^2)}{d\varepsilon^2} \quad (25)$$

$$\therefore \Pr\left(\left|\frac{\|X\|_2^2}{d} - \frac{1}{d}\right| \leq \varepsilon\right) \geq 1 - \frac{\mathbf{var}(X_i^2)}{d\varepsilon^2} \quad (26)$$

so $\|X\|_2^2 \approx 1$. It follows that $\mathbf{E}\left[\|X\|_2\right] = \mathbf{E}\left[\|Y\|_2\right] = 1$.

Furthermore, we have that

$$\mathbf{E}\left[\langle X, Y\rangle\right] = \mathbf{E}\left[\sum_{i=1}^{d} X_i Y_i\right] \quad (27)$$

$$= d\,\mathbf{E}\left[X_i Y_i\right] \quad (28)$$

$$= d\,\mathbf{E}\left[X_i\right]\mathbf{E}\left[Y_i\right] \quad (29)$$

$$= 0. \quad (30)$$

where Equation 29 follows from them being independent. □

Lemma A.2. Consider $A \in \mathbb{R}^{m\times p}$, $p \leq m$, such that $A_{ij} \sim \mathcal{U}\left(-\sqrt{\frac{3}{m}}, \sqrt{\frac{3}{m}}\right)$. Then for any $x \in \mathbb{R}^p$ we have that $\|Ax\|_2 \approx \|x\|_2$.

Proof. By Lemma A.1 the p columns of A, which are of dimension m, are on average of unit norm and orthogonal to each other (note $p \leq m$). As a result $A^T A \approx I_p$, so

$$\|Ax\|_2^2 = x^T A^T A x \approx x^T I x = \|x\|_2^2$$

implying that $\|Ax\|_2 \approx \|x\|_2$. □

Lemma A.3. Consider $A \in \mathbb{R}^{m\times p}$, $m >> 10$, such that $A_{ij} \sim \mathcal{U}\left(-\sqrt{\frac{3}{m}}, \sqrt{\frac{3}{m}}\right)$. Then for $x \in \mathbb{R}^p$ s.t. $\|x\|_2 \leq 1$ we have that $\sin(Ax) \approx Ax$.

Proof. By Lemma A.2 we have that $\|Ax\|_2 \approx \|x\|_2$. Furthermore since A is generated uniform randomly, the values of $Ax \in \mathbb{R}^m$ should be randomly distributed, therefore as $\|Ax\|_2 \approx \|x\|_2 \leq 1$ and $m \gg 10$, with high probability $|(Ax)_i| < 0.2$. Thus $\sin\left((Ax)_i\right) \approx (Ax)_i$ (as it is within the linear region of sine), so $\sin(Ax) \approx Ax$. □

Proposition 4.2. *Let Φ be a n-hidden layer SIREN (Equation 1) that maps from $\mathbb{R}^{M_0} \to \mathbb{R}$ and $\|x\|_2 \leq 1$. Set $\mathbf{W}_i \sim \mathcal{U}\left(-c_{wr}^i, c_{wr}^i\right)$, $c_{wr}^i = \sqrt{\frac{3}{M_{i+1}}}$, $\mathbf{b}_i = \mathbf{0}$ for $0 \leq i \leq n-2$ and $\mathbf{W}_{n-1} = \frac{\pi}{2} I$, $\mathbf{b}_{n-1} = \frac{\pi}{2}\mathbf{1}$, $\mathbf{w}_n = -\mathbf{1}$ and $b_n = M_n$. Then $\nu(\Phi(x)) \approx \|x\|_2$.*

Proof. We first prove that the input to the last hidden layer, x_{n-1}, has the property that $\|x_{n-1}\|_2 \approx \|x\|_2$. Proposition 4.1 then implies that $\nu(\Phi(x)) \approx \|x_{n-1}\|_2 \approx \|x\|_2$, as x_{n-1} is essentially the input to a one hidden layer network satisfying Proposition 4.1.

Now for for $0 \leq i \leq n-2$, if $\|x_i\|_2 \leq 1$,

$$\|x_i\|_2 \approx \|\mathbf{W}_i x_i\|_2 \quad (*)$$

$$\approx \|\sin(\mathbf{W}_i x_i + \mathbf{b}_i)\|_2 \quad (\dagger)$$

$$= \|x_{i+1}\|_2 \quad (31)$$

where $*$ holds from Lemma A.2 and † holds due to $\mathbf{b}_i = \mathbf{0}$ and Lemma A.3.

Thus as $\|x_0\|_2 = \|x\|_2 \leq 1$, by induction $\|x\| \approx \|x_{n-1}\|_2$. □

A.2.3 DiGS training procedure comparison

Following Section 3, we provide further visualizations in Figure 16 and Figure 17 to compare the difference between the training of DiGS and SIREN wo n, highlighting the four phases of DiGS. SIREN wo n fits to the shape very quickly, but does not do so in a consistent manner and thus has multiple surfaces interpolating the surface points that have differing orientations for what is inside and out (best seen with the 2D contours). It then tries to refine and improve upon this, but is stuck with the ghost geometry from its early fitting. DiGS on the other hand has a structured training procedure that prevents this: it slowly changes the SDF for the noisy sphere, allowing more and more details, which reduces incorrect orientation and thus ghost geometry from occurring. As SIREN wo n does not have a zero level set at initialization (see the contour plot for the 2D examples), there is no visualization for its initialization in 3D. Note that the initialization for DC is the same noisy sphere as the initialization for the gargoyle shape, however it may appear slightly different because it is rotated and cut due to the bounding box for the DC shape.

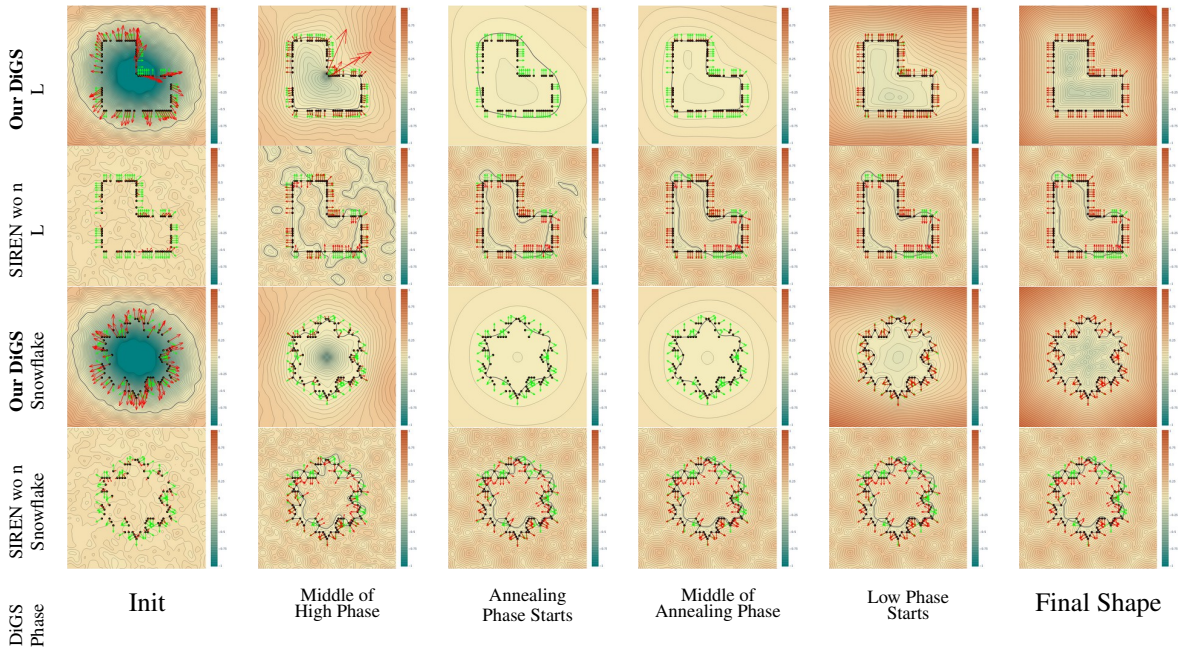

Figure 16. Visualisation of DiGS and SIREN wo n at six points during training for 2D shapes. The 6 points are labelled according to their position relative to the four phases of DiGS' training. A contour plot of the learned function is shown: the black dots are surface points, the green arrows are ground truth normals and the red arrows are the current normals at those points.

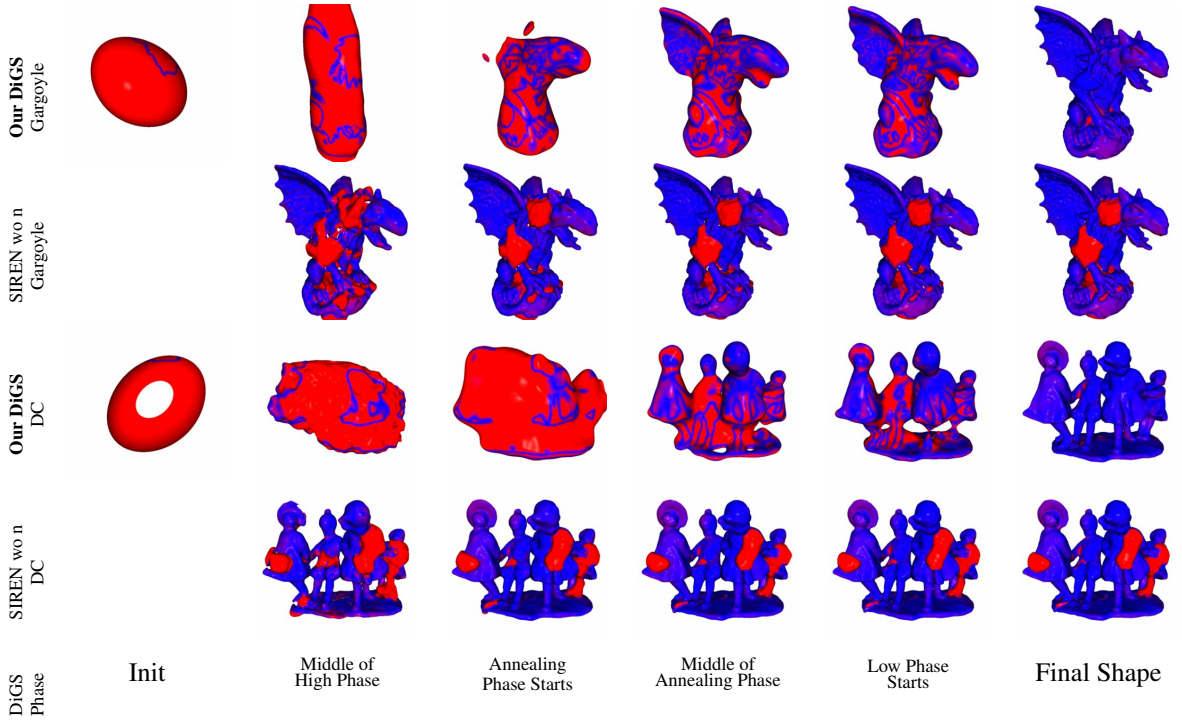

Figure 17. Visualisation of DiGS and SIREN wo n at six points during training on 3D shapes from SRB [7]. The 6 points are labelled according to their position relative to the four phases of DiGS' training. The current zero level set at the point is shown and colored by the distance to the closest ground truth point (thresholded at one). As SIREN wo n does not have a zero level set at initialization (see the contour plot for the 2D examples), there is no visualization for its initialization in 3D.

A.3. Additional experiments and detail

Code and model parameters available at our project page https://chumbyte.github.io/DiGS-Site/.

A.3.1 Evaluation metrics

To compare between two point sets $\mathcal{X}_1, \mathcal{X}_2 \subset \mathbb{R}^3$, we use the Chamfer ($d_C$) and Hausdorff ($d_H$) distances from Williams et al. [44]. For the ShapeNet dataset, instead of this Chamfer distance we report the squared Chamfer to be consistent with previous works [34, 45]. Note that these metrics compares the accuracy of the predicted surface of the shape. To compare between the implicit function learnt and the ground truth mesh, we use the volumetric Intersection over Union (IoU) of the interior of the shapes as per Mescheder et al. [29]. Note that this metric compares the underlying occupancy (interior) predicted by the method.

We define the

$$d_C(\mathcal{X}_1, \mathcal{X}_2) = \frac{1}{2}(d_{\vec{C}}(\mathcal{X}_1, \mathcal{X}_2) + d_{\vec{C}}(\mathcal{X}_2, \mathcal{X}_1)) \tag{32}$$

$$d_H(\mathcal{X}_1, \mathcal{X}_2) = \max(d_{\vec{H}}(\mathcal{X}_1, \mathcal{X}_2), d_{\vec{H}}(\mathcal{X}_2, \mathcal{X}_1)) \tag{33}$$

$$\tag{34}$$

where

$$d_{\vec{C}}(\mathcal{X}_1, \mathcal{X}_2) = \frac{1}{|\mathcal{X}_1|} \sum_{x_1 \in \mathcal{X}_1} \min_{x_2 \in \mathcal{X}_2} \|x_1 - x_2\|_2 \tag{35}$$

$$d_{\vec{H}}(\mathcal{X}_1, \mathcal{X}_2) = \max_{x_1 \in \mathcal{X}_1} \min_{x_2 \in \mathcal{X}_2} \|x_1 - x_2\|_2 \tag{36}$$

are the one directional Chamfer distance and one directional Hausdorff distance respectively.

We define the Chamfer distance used in Park et al. [34] as the squared Chamfer distance, given by

$$d_C^{sq}(\mathcal{X}_1, \mathcal{X}_2) = d_{\vec{C}}^{sq}(\mathcal{X}_1, \mathcal{X}_2) + d_{\vec{C}}^{sq}(\mathcal{X}_2, \mathcal{X}_1) \tag{37}$$

$$d_{\vec{C}}^{sq}(\mathcal{X}_1, \mathcal{X}_2) = \frac{1}{|\mathcal{X}_1|} \sum_{x_1 \in \mathcal{X}_1} \min_{x_2 \in \mathcal{X}_2} \|x_1 - x_2\|_2^2 . \tag{38}$$

$$\tag{39}$$

For the volumetric IoU, following Mescheder et al. [29] we obtain unbiased estimates of the occupied (interior) volume of the shapes by randomly sampling 100k points $\mathcal{X}$ in the space. Thus given the known occupancy of the ground truth mesh $O_{GT}(x) \in \{0, 1\}$ and the predicted SDF $\Phi(x) \in \mathbb{R}$ for a points $x \in \mathcal{X}$, the occupancy of the SDF is given by

$$O_\Phi(x) = \begin{cases} 1 & \Phi(x) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{40}$$

method	SIREN	IGR	DiGS
time [ms]	5.2	17.5	12.0
# parameters	66.5K	2.1M	66.5K

Table 5. Time performance results per iteration. Comparing standard SIREN (excluding the normal estimation stage time) to DiGS. Results are reported in milliseconds (ms).

and the IoU is given by

$$IoU_{\mathcal{X}}(O_{GT}, O_\Phi) = \frac{\sum_{x \in \mathcal{X}} O_{GT}(x) \text{ and } O_\Phi(x)}{\sum_{x \in \mathcal{X}} O_{GT}(x) \text{ or } O_\Phi(x)}. \tag{41}$$

A.3.2 Experimental setup and timing performance

The surface reconstruction experiments on the Surface Reconstruction Benchmark dataset [7] were implemented in PyTorch and trained on a single Nvidia RTX 2080 GPU. The architecture for 2D experiments our method is a 4 layer MLP with sinusoidal activation (SIREN) with 128 nodes in each layer. The architecture for 3D reconstruction experiments our method is a 4 layer MLP with sinusoidal activation (SIREN) with 256 nodes in each layer, and for scene reconstruction we increase that to 8 layers and 512 nodes. For shapespace we used an 8 layer MLP with sinusoidal activation (SIREN) with 512 nodes in each layer. Our method does not add parameters to the standard SIREN approach. Note that due computing the divergence term, there is an increase in computation time which is mainly attributed to computing (and back propagating) the gradient of the gradient. At inference time SIREN and DiGS have the same time and computation performance. Note that to train the standard SIREN, a preprocessing normal estimation stage is required. The number of parameters and timing performance are reported in Table 5 for the 2D reconstruction networks. It shows the increase in training time (per iteration) for DiGS compared to SIREN, however DiGS is still faster and has fewer parameters compared to IGR.

A.3.3 Surface reconstruction on SRB

Implementation details. The input point cloud is first centered to zero and scaled to have maximum norm of one. Then a bounding box that is 1.1 times the size of the shape is selected. Each iteration we sample 15,000 points from the original point cloud and sample 15,000 points uniformly randomly in a bounding box. We train for 10,000 iterations with a learning rate of 5e-5.

We report the values for DGP [44], NSP [45] and PHASE/PHASE+FF [26] from their respective works, and report the results for FFN [39] from Williams *et al.* [45] and IGR+FF from Lipman et al. [26]. We report the results for

	Mean		Anchor				Daratech				DC				Gargoyle				Lord Quas			
	GT		GT		Scans		GT		Scans		GT		Scans		GT		Scans		GT		Scans	
Method	d_C	d_H	d_C	d_H	$d_{\vec{C}}$	$d_{\vec{H}}$	d_C	d_H	$d_{\vec{C}}$	$d_{\vec{H}}$	d_C	d_H	$d_{\vec{C}}$	$d_{\vec{H}}$	d_C	d_H	$d_{\vec{C}}$	$d_{\vec{H}}$	d_C	d_H	$d_{\vec{C}}$	$d_{\vec{H}}$
DGP	0.21	5.18	0.33	8.82	**0.08**	2.79	0.20	3.14	**0.04**	1.89	0.18	4.31	**0.04**	**2.53**	0.21	5.98	**0.06**	3.41	0.14	3.67	**0.04**	2.03
IGR	0.19	2.99	0.23	4.71	0.12	1.32	0.25	4.01	0.14	1.59	0.17	2.22	0.09	2.61	0.18	**2.85**	0.1	1.29	0.12	1.17	0.07	0.98
SIREN	0.19	3.86	0.31	7.32	0.11	1.23	0.21	4.74	0.09	1.85	0.15	2.37	0.07	2.71	0.17	4.26	0.09	**0.82**	0.12	**0.62**	0.08	0.81
NSP	0.17	2.85	0.22	4.65	0.11	**1.11**	0.21	4.35	0.08	**1.14**	**0.14**	**1.35**	0.06	2.75	**0.16**	3.20	0.08	2.75	0.12	0.69	0.05	**0.62**
PHASE	**0.16**	**2.77**	**0.21**	**4.29**	0.09	1.23	**0.18**	**2.92**	0.08	1.80	0.15	2.52	0.05	2.78	**0.16**	3.14	0.07	1.09	**0.11**	0.96	0.04	0.96
DiGS + n	0.18	3.55	0.28	5.71	0.11	1.14	0.21	5.02	0.09	1.75	0.15	2.13	0.06	2.74	**0.16**	3.81	0.09	0.90	0.12	1.1	0.06	0.77
IGR wo n	1.38	16.33	0.45	7.45	0.17	4.55	4.9	42.15	0.7	3.68	0.63	10.35	0.14	3.44	0.77	17.46	0.18	2.04	0.16	4.22	0.08	1.14
SIREN wo n	0.42	7.67	0.72	10.98	0.11	1.27	0.21	4.37	0.09	1.78	0.34	6.27	0.06	**2.71**	0.46	7.76	0.08	**0.68**	0.35	8.96	0.06	**0.65**
SAL	0.36	7.47	0.42	7.21	0.17	4.67	0.62	13.21	0.11	2.15	0.18	3.06	0.08	2.82	0.45	9.74	0.21	3.84	0.13	4.14	0.07	4.04
IGR+FF	0.96	11.06	0.72	9.48	0.24	8.89	2.48	19.6	0.74	4.23	0.86	10.3	0.28	3.98	0.26	5.24	0.18	2.93	0.49	10.7	0.14	3.71
PHASE+FF	0.22	4.96	**0.29**	7.43	**0.09**	1.49	0.35	7.24	**0.08**	**1.21**	0.19	4.65	**0.05**	2.78	**0.17**	4.79	**0.07**	1.58	**0.11**	**0.71**	**0.05**	0.74
Our DiGS	**0.19**	**3.52**	**0.29**	**7.19**	0.11	**1.17**	**0.20**	**3.72**	0.09	1.80	**0.15**	**1.70**	0.07	2.75	**0.17**	**4.10**	0.09	0.92	0.12	0.91	0.06	0.70

Table 6. Results on the Surface Reconstruction Benchmark using Chamfer d_C, Hausdorff distance d_H. We compare methods with normal supervision above the line and without normal supervision below the line. The *scans* column reports the one sided distances $(d_{\vec{C}}, d_{\vec{H}})$ between the reconstruction and the simulated scans which give a measure of the reconstruction's overfit to the noisy input.

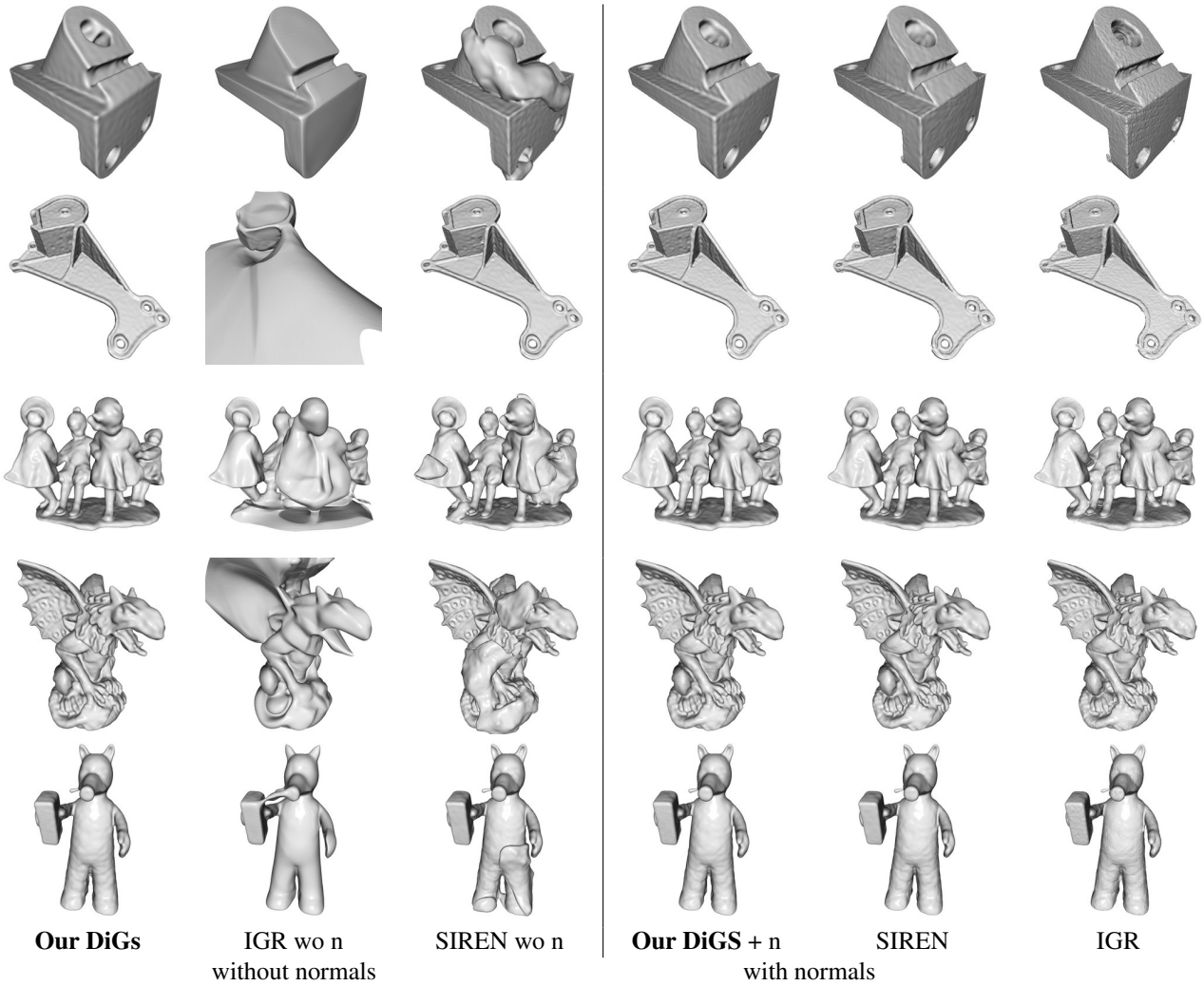

Figure 18. Qualitative results of surface reconstruction on the anchor and gargoyle shapes from the Surface Reconstruction Benchmark [7] compared to state of the art approaches (IGR, SIREN) that use normal vectors as ground truth.

Method	GT d_C	GT d_H	Scans $d_{\vec{C}}$	Scans $d_{\vec{H}}$
SIREN wo n	0.42	7.67	0.08	**1.42**
DiGS L_1	0.20	4.47	**0.07**	1.77
DiGS L_2	0.25	5.18	**0.07**	1.67
DiGS no-decay	0.21	4.45	0.09	2.58
DiGS lin-decay	0.20	4.41	**0.07**	1.64
DiGS curv	0.30	4.83	0.10	1.57
DiGS L_1 MFGI	**0.19**	**3.52**	0.08	1.47

Table 7. DiGS ablation study on the Surface Reconstruction Benchmark [7]. We compare design choices such as annealing method (linear, step and no annealing), divergence term penalties (L_1 vs L_2), initialization method (MFGI vs SIREN) and curvature ground truth. We use the Chamfer d_C and Hausdorff distance d_H metrics.

SAL [2], IGR/IGR wo n [17] and SIREN/SIREN wo n [37] using their code.

Additional quantitative results. We provide additional quantitative results for surface reconstruction on the Surface Reconstruction Benchmark [7]. Table 6 is an extended version of Table 2 from the main paper that includes comparison to additional methods with normal vector supervision. The results show that we are able to achieve the best reconstruction compared to other methods without normal supervision. Additionally,we achieve improved performance when using normal vector supervision compared to a vanilla SIREN and show comparable results to other methods.

Additional qualitative results. We provide additional qualitative results for surface reconstruction on the Surface Reconstruction Benchmark [7]. Figure 18 shows visualizations of the output reconstruction of different methods that use normal vector as ground truth as well as methods that do not. It is clear that whenever ground truth normal vectors are not available, DiGS presents a significant improvement and yields comparable results to normal based methods.

Ablation study. We investigate the effects of several design choices made for DiGS and report the averages over all shapes in the dataset in Table 7 and individual shapes in Table 8. First we investigate the influence of the annealing function τ. We compare between the case of

1. No annealing:
$$\tau = 1. \tag{42}$$

2. Linear annealing:
$$\tau_{Lin}(t_0, t_1, \tau_1) = \begin{cases} 1 & t < t_0 \\ 1 + (\tau_1 - 1)\,\frac{(t-t_0)}{t_1 - t_0} & t_0 \leq t \leq t_1 \\ \tau_1 & t > t_1 \end{cases} \tag{43}$$

3. Step annealing:
$$\tau_{Step}(t_0, \tau_1) = \begin{cases} 1 & t < t_0 \\ \tau_1 & t \geq t_0 \end{cases} \tag{44}$$

	Method	GT d_C	GT d_H	Scans $d_{\vec{C}}$	Scans $d_{\vec{H}}$
anchor	SIREN wo n	0.72	10.98	0.11	1.27
	DiGS l1	0.33	8.71	0.11	2.52
	DiGS l2	0.43	8.24	**0.10**	2.23
	DiGS no-decay	0.35	8.84	0.12	4.38
	DiGS lin-decay	0.33	8.71	**0.10**	2.05
	DiGS curv	0.78	11.63	0.13	1.21
	DiGS l1 MFGI	**0.29**	**7.19**	0.11	**1.17**
daratech	SIREN wo n	0.21	4.37	0.09	1.78
	DiGS l1	0.21	3.50	**0.07**	1.81
	DiGS l2	0.20	3.47	**0.07**	1.78
	DiGS no-decay	0.20	3.35	0.09	1.79
	DiGS lin-decay	**0.19**	**2.97**	**0.07**	1.77
	DiGS curv	0.22	4.27	0.12	**1.76**
	DiGS l1 MFGI	0.20	3.72	0.09	1.80
dc	SIREN wo n	0.34	6.27	**0.06**	**2.71**
	DiGS l1	**0.15**	2.25	**0.06**	2.76
	DiGS l2	0.19	4.04	**0.06**	2.76
	DiGS no-decay	0.16	2.58	0.07	2.78
	DiGS lin-decay	0.16	2.72	**0.06**	**2.71**
	DiGS curv	0.16	1.76	0.08	2.82
	DiGS l1 MFGI	**0.15**	**1.70**	0.07	2.75
gargoyle	SIREN wo n	0.46	7.76	**0.08**	**0.68**
	DiGS l1	**0.17**	5.12	**0.08**	0.81
	DiGS l2	**0.17**	5.02	**0.08**	0.75
	DiGS no-decay	0.18	5.18	0.09	3.10
	DiGS lin-decay	**0.17**	5.09	**0.08**	0.81
	DiGS curv	0.19	**3.90**	0.12	1.30
	DiGS l1 MFGI	**0.17**	4.10	0.09	0.92
lord quas	SIREN wo n	0.35	8.96	**0.06**	**0.65**
	DiGS l1	**0.12**	2.77	**0.06**	0.94
	DiGS l2	0.25	5.12	**0.06**	0.85
	DiGS no-decay	0.13	2.30	**0.06**	0.85
	DiGS lin-decay	0.13	2.59	**0.06**	0.88
	DiGS curv	0.14	2.61	0.07	0.75
	DiGS l1 MFGI	**0.12**	**0.91**	**0.06**	0.70

Table 8. DiGS ablation study on the Surface Reconstruction Benchmark. We compare design choices such as annealing method (linear, step and no annealing), divergence term penalties (L_1 vs L_2), initialization method (MFGI vs SIREN) and curvature ground truth. We use the Chamfer d_C and Hausdorff distance d_H metrics.

Here t_0 and t_1 are expressed as a fraction of training iterations. In our experiments we used the following values: $(t_0, t_1, \tau_1) = (0.5, 0.75, 0)$. Results show that annealing improves performance with little difference between linear and step annealing.

We also investigate the choice of penalty function over the divergence term (with step decay) and compare between L_1 and L_2. Here, L_1 allows for sparse spatial locations to have high divergence (which is desired because there may be some source or sink point in the gradient vector field) while L_2 provides an average low value over the volume. As expected, the results show an advantage for using L_1 penalty on the divergence.

Furthermore, we evaluate the second order supervision approach presented in Section A.1.4. For that, we estimate the mean curvature using DeepFit [5] (a recent state-of-the-art method for estimating normals and curvatures) and introduce the mean curvature as ground truth information during training. The results show that, surprisingly, curvature supervision does not provide any benefit. This is due to the

noisiness and local support dependency of curvature estimation which may have over smoothed or jittery values that make training inconsistent.

Finally, we investigate the effect of the multi-frequency geometric initialization (with step decay and L_1 penalty) and show that it produces consistently better performance than all other ablations. This variant is denoted throughout the paper as DiGS, *i.e.*, L_1 penalty on the divergence term with step decay and MFGI (Section 4). Note that all ablations were trained without normal vector information.

Use of DiGS Loss with other activation functions. Our divergence loss is general and can be applied to any network that has second order derivatives defined everywhere. Note that this means it cannot be used with ReLU activations, though we can use smooth approximations such as the SoftPlus activation function (which IGR [17] uses).
However, our loss is targeted at high frequency architectures such as SIRENs. A drawback of such architectures is the trade off between high fidelity detail and ghost geometries, especially without normal vector supervision. Our loss is particularly beneficial to alleviate this trade-off.
We experimented with using our loss paired with SoftPlus and found that it does not provide a boost in performance since the network is already biased towards low frequency solutions (performance reduction of 0.99 d_C, and 6.34 d_H on SRB) which *SIREN wo n* outperforms even without the divergence loss performance boost.

A.3.4 Surface reconstruction on ShapeNet

Implementation details. We use the preprocessing and evaluation method from Williams et al. [45]. They first preprocess using the method from Mescheder et al. [29], then report on the first 20 shapes of the test set for each shape class. The preprocessing extracts ground truth surface points from the shapes of ShapeNet v1 [15], and extracts random samples within the space with their labelled occupancy values. The evaluation method uses the ground truth points to calculate squared Chamfer distance, and uses the labelled random samples to calculate IoU. Note that the initial ShapeNet data has inconsistent normal orientation and many non-manifold surfaces due to its nature as CAD models, and this preprocessing helps orient the normals and remove most of the non-manifold surfaces.

For our method, given the input point cloud from the preprocessing, we continue as we did for SRB (Section A.3.3). We report the values for SPSR [23], IGR [17], SIREN [37], FFN [39] and NSP [45] from Williams et al. [45]. We report the results for SIREN wo n [37] and SAL [2] using their code.

Additional quantitative results. We give the breakdown of squared Chamfer distance and IoU per shape class in Table 9. As with the summary over all shape classes, for the individual breakdowns we can see that with squared Chamfer distance we get better means and medians without normals, and with normals we get better medians but often not better means. A particular class to note is loudspeaker, it is the only class that we do not get better medians for when comparing with normals, and we can see that it does much worse with normals than without. As stated in the main paper, we find that this is due to there being significant internal ghost geometry due to trying to match the internal parts of the loudspeakers, and having normal vector supervision causes even more of such ghost geometry. For IoU we do similarly: we get better means means and medians without normals, but with normals we get better medians but sometimes not better means.

Additional qualitative results. We show additional visualisations in Figure 19 and Figure 20. A comparison between the ground truth mesh, DiGS, SIREN wo n, SAL and NSP is done for a shape from each shape class. SIREN wo n and SAL, which do not use normal information, are able to reconstruct the shapes, but the former has a lot of ghost geometry while the latter is overly smoothened or missing thin surfaces. On the other hand, DiGS is able to perform on par with the best method with normal information, NSP, where failure cases are usually extra thin surfaces.

A.3.5 Scene Reconstruction

In Section 6.1 we presented qualitative results for scene reconstruction. Attached to this supplemental material, we provide a low resolution video depicting the scene from multiple angles. A high resolution video is available an external URL. For this task we use eight layers with 512 units and train on the scene from Sitzmann *et al*. [37] which includes 10M oriented points. We train for $100K$ iterations, sampling random $15K$ points in each iteration.

A.3.6 Shape Space Learning

We use the setup and evaluation procedure of Gropp et al. [17]. The DFaust dataset [10] has scans of 10 humans, each of which do multiple action sequences and have a scan at different points during the sequence. The data set include both the scan done, which are high-resolution triangle soups, and their own registration for the complete mesh of the human which they gained from using extra data (e.g., colour) and the temporal information during the action sequence.

We use the random split setting of Gropp et al. [17], a 75%-25% split between a significant subset of all scans (8566 scans). For the shapespace experiment, a single model is trained on all training scans. During training, a latent code is assigned to each action sequence of each human, which is allowed to be trained. During test time the latent code for the test shape is optimised using the input

point cloud data, after which the optimised latent code is used to evaluate the full shape on a $512 \times 512 \times 512$ grid.

Following the evaluation procedure of Gropp et al. [17] we report the mean and median of the total one-sided Chamfer distances between the reconstruction and the input scans, and the reconstructions and the ground truth.

For our network we use a 8 layer SIREN with hidden layers of dimension 256. We also use 256 dimensional latent codes. At test time we optimise for the latent vectors for 800 iterations using Adam with a learning rate of 10^{-3}. The latent code and the original input ((x, y, z) coordinates) are concatenated together for a 259 dimensional input, and the latent codes are intialised to zero. As is standard in an autodecoder we use latent code regularization [34], with a scaling weight of 1.0.

For IGR we use their model (autodecoder with 8 layers, hidden dimension 512, latent dimension 256) and use the same latent optimisation procedure. For the version with normals we use their trained weights, for the version without normals we train their network using their provided training script.

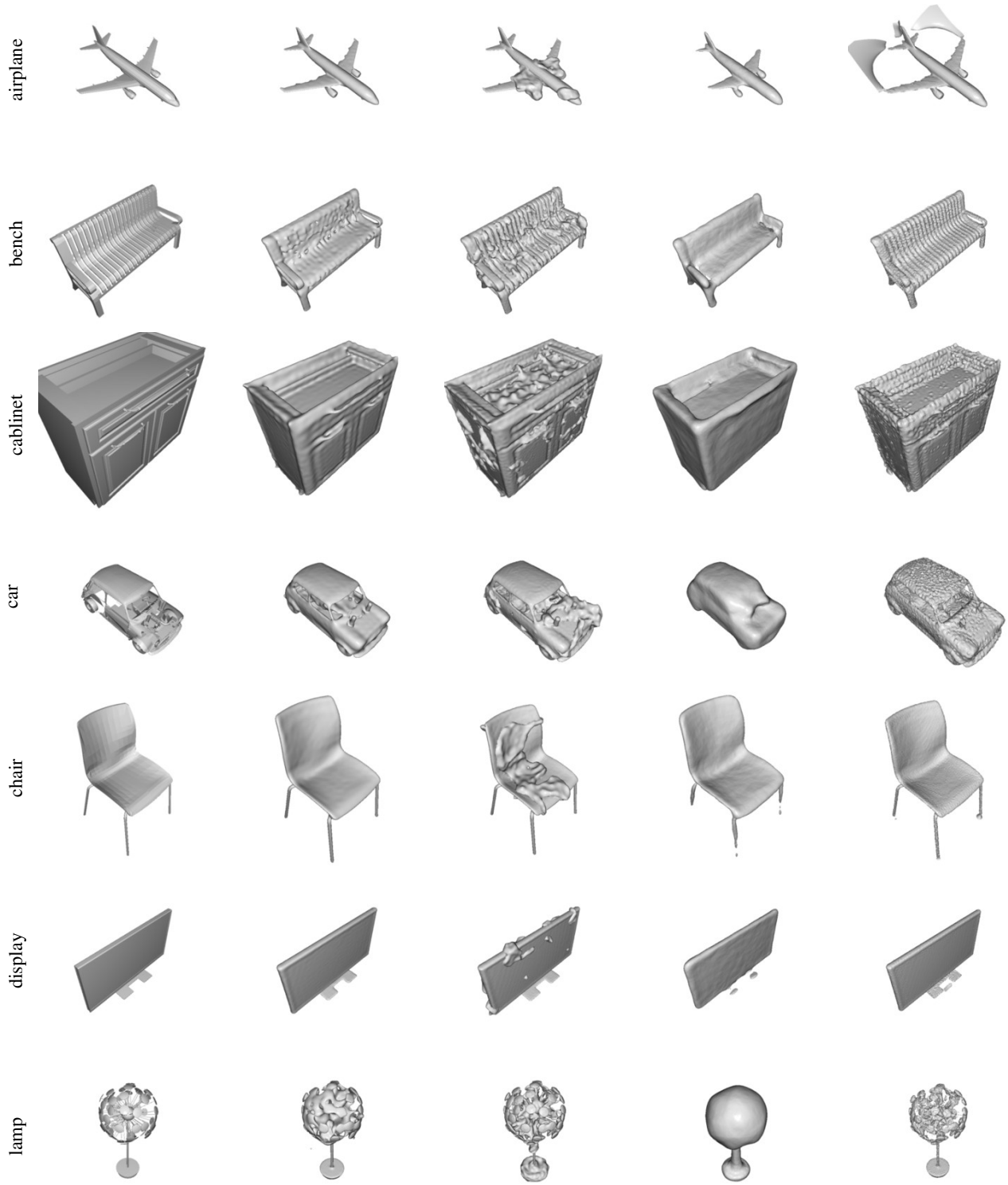

Figure 19. Qualitative results on ShapeNet. One shape from each class is shown compared to other methods.

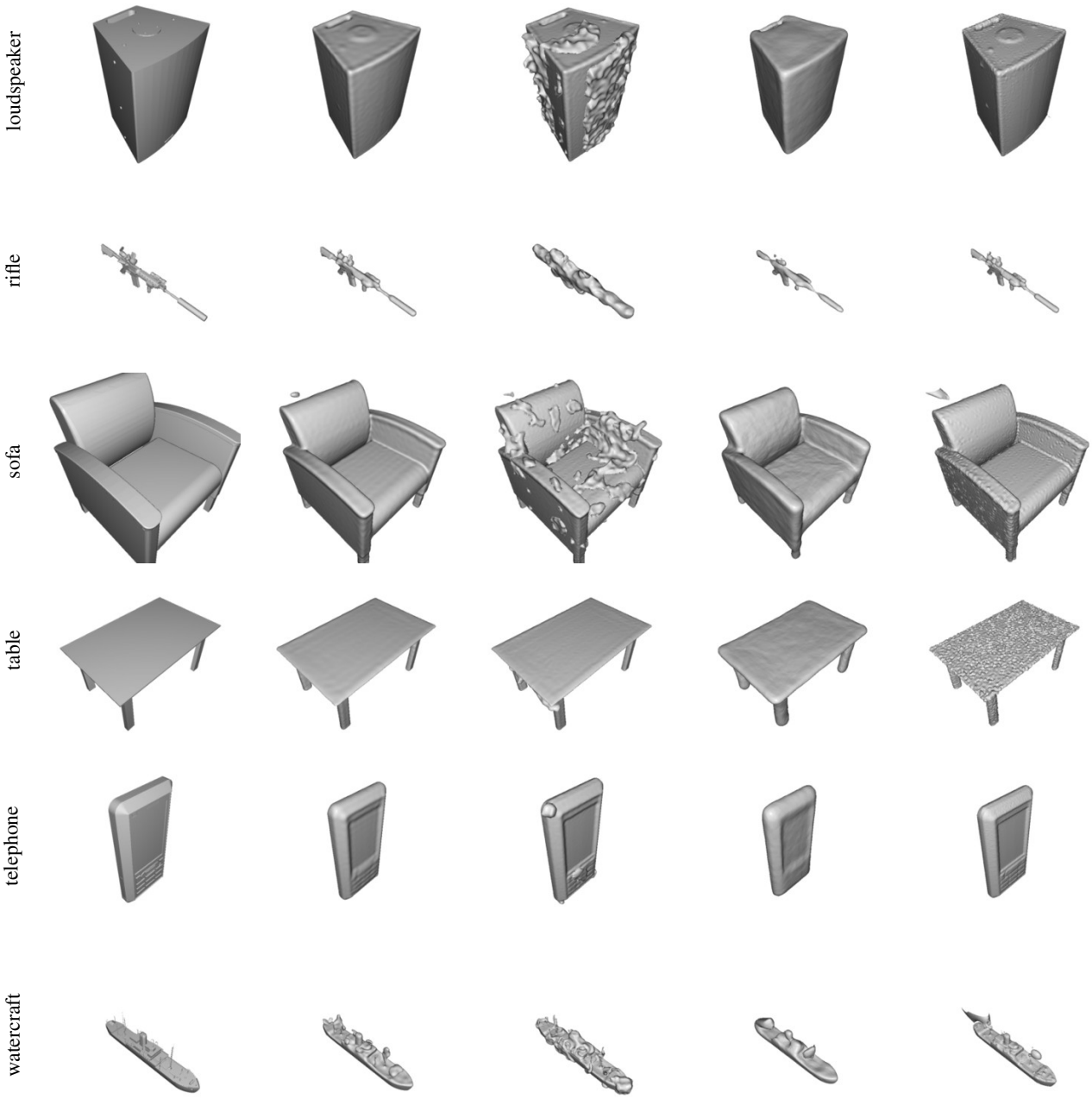

Figure 20. Qualitative results on ShapeNet. One shape from each class is shown compared to other methods.

Squared Chamfer

Method	All			airplane			bench			cabinet			car		
	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std
IGR	6.66e-4	1.07e-4	4.69e-3	3.04e-4	1.74e-4	3.47e-4	4.48e-4	2.58e-4	4.33e-4	1.56e-4	9.39e-5	1.23e-4	2.60e-4	2.82e-4	9.80e-5
SIREN	1.03e-4	5.28e-5	1.93e-4	4.15e-5	3.87e-5	8.57e-6	9.63e-5	8.12e-5	5.41e-5	1.51e-4	6.69e-5	1.77e-4	1.39e-4	9.07e-5	1.03e-4
NSP	**5.36e-5**	4.06e-5	3.64e-5	3.55e-5	3.44e-5	2.45e-6	5.66e-5	4.82e-5	2.09e-5	**6.98e-5**	4.69e-5	4.34e-5	**8.21e-5**	7.18e-5	3.60e-5
DiGS + n	2.74e-4	**2.32e-5**	9.90e-4	**1.05e-5**	**9.29e-6**	3.93e-6	**3.11e-5**	**2.17e-5**	3.98e-5	6.92e-4	**4.28e-5**	1.10e-3	3.96e-4	**3.87e-5**	1.52e-3
SIREN wo n	3.08e-4	2.58e-4	3.26e-4	2.42e-4	2.50e-4	5.92e-5	1.93e-4	1.67e-4	9.09e-5	3.16e-4	2.72e-4	1.72e-4	2.67e-4	2.58e-4	4.78e-5
SAL	1.14e-3	2.11e-4	3.63e-3	5.98e-4	2.38e-4	9.22e-4	3.55e-4	1.71e-4	4.26e-4	**2.81e-4**	1.86e-4	1.81e-4	4.51e-4	2.74e-4	4.36e-4
Our DiGS	**1.32e-4**	**2.55e-5**	4.73e-4	**1.32e-5**	**1.01e-5**	7.56e-6	**7.26e-5**	**2.21e-5**	1.74e-4	4.07e-4	**4.45e-5**	9.25e-4	**7.89e-5**	**3.97e-5**	1.10e-4

Method	chair			display			lamp			loudspeaker			rifle		
	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std
IGR	9.25e-4	9.88e-5	3.11e-3	9.99e-5	7.49e-5	8.44e-5	1.72e-3	1.28e-4	6.24e-3	3.77e-3	1.15e-4	1.49e-2	9.62e-5	5.29e-5	1.25e-4
SIREN	1.05e-4	6.34e-5	1.18e-4	6.98e-5	5.68e-5	3.86e-5	6.26e-5	5.07e-5	3.35e-5	2.77e-4	6.88e-5	5.54e-4	3.62e-5	3.50e-5	4.03e-6
NSP	**5.62e-5**	4.21e-5	4.32e-5	**4.36e-5**	3.99e-5	1.28e-5	4.19e-5	3.91e-5	1.00e-5	**8.41e-5**	**4.54e-5**	7.54e-5	3.26e-5	3.15e-5	2.79e-6
DiGS + n	8.55e-5	**2.43e-5**	1.43e-4	8.67e-4	**2.52e-5**	2.45e-3	**3.34e-5**	**1.70e-5**	4.80e-5	1.05e-3	7.13e-4	1.14e-3	**4.80e-6**	**4.73e-6**	1.74e-6
SIREN wo n	**2.63e-4**	2.60e-4	1.31e-04	2.49e-4	2.20e-4	8.45e-05	6.10e-4	3.49e-4	1.04e-03	3.29e-4	3.04e-4	1.31e-04	5.44e-4	5.56e-4	1.44e-04
SAL	1.28e-3	2.92e-4	2.05-3	2.56e-4	8.86e-5	4.99-4	5.86e-3	1.29e-3	9.35-3	4.04e-4	2.63e-4	4.50-4	2.18e-3	1.15e-4	5.17-3
Our DiGS	3.72e-4	**2.73e-5**	1.05e-3	**3.16e-5**	**2.53e-5**	2.32e-5	**1.70e-4**	**2.18e-5**	3.96e-4	**1.18e-4**	**6.18e-5**	2.15e-4	**9.10e-6**	**5.26e-6**	1.03e-5

Method	sofa			table			telephone			watercraft		
	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std
IGR	2.86e-4	1.02e-4	5.30e-4	3.40e-4	1.95e-4	3.33e-4	1.03e-4	4.43e-5	1.54e-4	1.47e-4	1.12e-4	1.23e-4
SIREN	7.88e-5	6.99e-5	3.90e-5	1.92e-4	8.32e-5	2.32e-4	3.88e-5	3.58e-5	9.64e-6	5.57e-5	4.21e-5	2.95e-5
NSP	**5.11e-5**	4.80e-5	1.24e-5	**6.60e-5**	4.88e-5	4.17e-5	**3.34e-5**	3.19e-5	3.60e-6	4.41e-5	3.84e-5	1.42e-5
DiGS + n	6.83e-5	**2.77e-5**	9.39e-5	1.68e-4	**3.26e-5**	3.50e-4	1.15e-4	**1.75e-5**	3.05e-4	**2.77e-5**	**1.57e-5**	3.30e-5
SIREN wo n	2.72e-4	2.66e-4	6.74e-05	**2.29e-4**	2.38e-4	8.40e-05	2.10e-4	1.86e-4	6.60e-05	2.97e-4	2.43e-4	1.26e-04
SAL	3.75e-4	1.93e-4	4.31-4	1.82e-3	5.10e-4	4.31-3	1.04e-4	6.81e-5	7.99-5	8.08e-4	2.06e-4	1.75-3
Our DiGS	**5.76e-5**	**3.27e-5**	5.39e-5	2.94e-4	**2.98e-5**	6.76e-4	**1.77e-5**	**1.74e-5**	4.49e-6	**6.10e-5**	**2.43e-5**	9.03e-5

IoU

Method	All			airplane			bench			cabinet			car		
	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std
IGR	0.8102	0.8480	0.1519	0.7851	0.8193	0.0977	0.5812	0.5923	0.2487	0.8709	0.8857	0.0924	0.8026	0.8664	0.1300
SIREN	0.8268	0.9097	0.2329	0.8045	0.9080	0.2696	0.6109	0.7442	0.3258	0.8706	0.9263	0.1621	0.8036	0.9241	0.2753
NSP	0.8973	0.9230	0.0871	0.8165	0.8998	0.1551	0.7872	0.8370	0.1236	**0.9274**	0.9291	0.0422	0.8954	0.9288	0.0740
DiGS + n	**0.9200**	**0.9774**	0.1992	**0.9693**	**0.9718**	0.0151	**0.9428**	**0.9655**	0.0644	0.8323	**0.9867**	0.3076	**0.9147**	**0.9754**	0.2126
SIREN wo n	0.3085	0.2952	0.2014	0.2248	0.1735	0.1103	0.4020	0.4231	0.1953	0.3014	0.2564	0.1275	0.3336	0.3030	0.0997
SAL	0.4030	0.3944	0.2722	0.1908	0.1693	0.0955	0.2260	0.2311	0.1401	0.6923	0.7224	0.1637	0.6261	0.6526	0.1525
Our DiGS	**0.9390**	**0.9764**	0.1262	**0.9613**	**0.9577**	0.0164	**0.9061**	**0.9536**	0.1413	**0.9261**	**0.9853**	0.2137	**0.9455**	**0.9765**	0.0699

Method	chair			display			lamp			loudspeaker			rifle		
	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std
IGR	0.8049	0.8320	0.1022	0.8741	0.8917	0.0533	0.7865	0.8259	0.1318	0.8867	0.9324	0.1017	0.8279	0.8267	0.0542
SIREN	0.8721	0.8807	0.0495	0.9014	0.9146	0.0440	0.8392	0.8995	0.2025	0.8458	0.9618	0.2404	0.7329	0.9132	0.3662
NSP	0.8841	0.9034	0.0825	**0.9309**	0.9316	0.0251	**0.9037**	0.9178	0.0512	**0.9323**	0.9627	0.0599	0.9299	0.9313	0.0215
DiGS + n	**0.9719**	**0.9759**	0.0140	0.8367	**0.9855**	0.3485	0.9024	**0.9637**	0.1991	0.8798	**0.9747**	0.2424	**0.9569**	**0.9571**	0.0207
SIREN wo n	0.4208	0.3748	0.2322	0.3566	0.3123	0.1790	0.3055	0.2573	0.2598	0.2229	0.1724	0.1575	0.0265	0.0092	0.0554
SAL	0.2589	0.1491	0.2213	0.5067	0.5801	0.2474	0.1689	0.0698	0.1994	0.6702	0.7264	0.1976	0.2835	0.2821	0.1530
Our DiGS	**0.9082**	**0.9650**	0.1523	**0.9839**	**0.9886**	0.0102	**0.8776**	**0.9646**	0.1943	**0.9632**	**0.9851**	0.0978	**0.9486**	**0.9567**	0.0281

Method	sofa			table			telephone			watercraft		
	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std
IGR	0.8891	0.9139	0.0708	0.6852	0.7260	0.2004	0.9148	0.9372	0.0639	0.8146	0.8445	0.0931
SIREN	0.9251	0.9411	0.0390	0.7280	0.8058	0.2089	0.9427	0.9514	0.0310	0.8722	0.9279	0.1990
NSP	0.9387	0.9473	0.0264	0.8414	0.8427	0.0534	**0.9569**	0.9625	0.0260	0.9207	0.9231	0.0402
DiGS + n	**0.9624**	**0.9859**	0.0696	**0.9284**	**0.9784**	0.1743	0.8880	**0.9855**	0.2935	**0.9747**	**0.9789**	0.0168
SIREN wo n	0.3397	0.3444	0.1206	0.3797	0.3603	0.1528	0.3778	0.3806	0.2590	0.3190	0.3007	0.1877
SAL	0.4844	0.4530	0.1404	0.0965	0.0320	0.1502	0.6025	0.6704	0.2203	0.4170	0.4728	0.2367
Our DiGS	**0.9572**	**0.9807**	0.0896	**0.8943**	**0.9720**	0.1996	**0.9854**	**0.9876**	0.0071	**0.9522**	**0.9735**	0.0504

Table 9. Extended results for surface reconstruction on ShapeNet [15]. For each shape class, and all shapes together, we report the squared Chamfer distance (first three tables) and the IoU (last three tables) to the ground truth mesh. Methods above the line use normal supervison, and methods below do not.

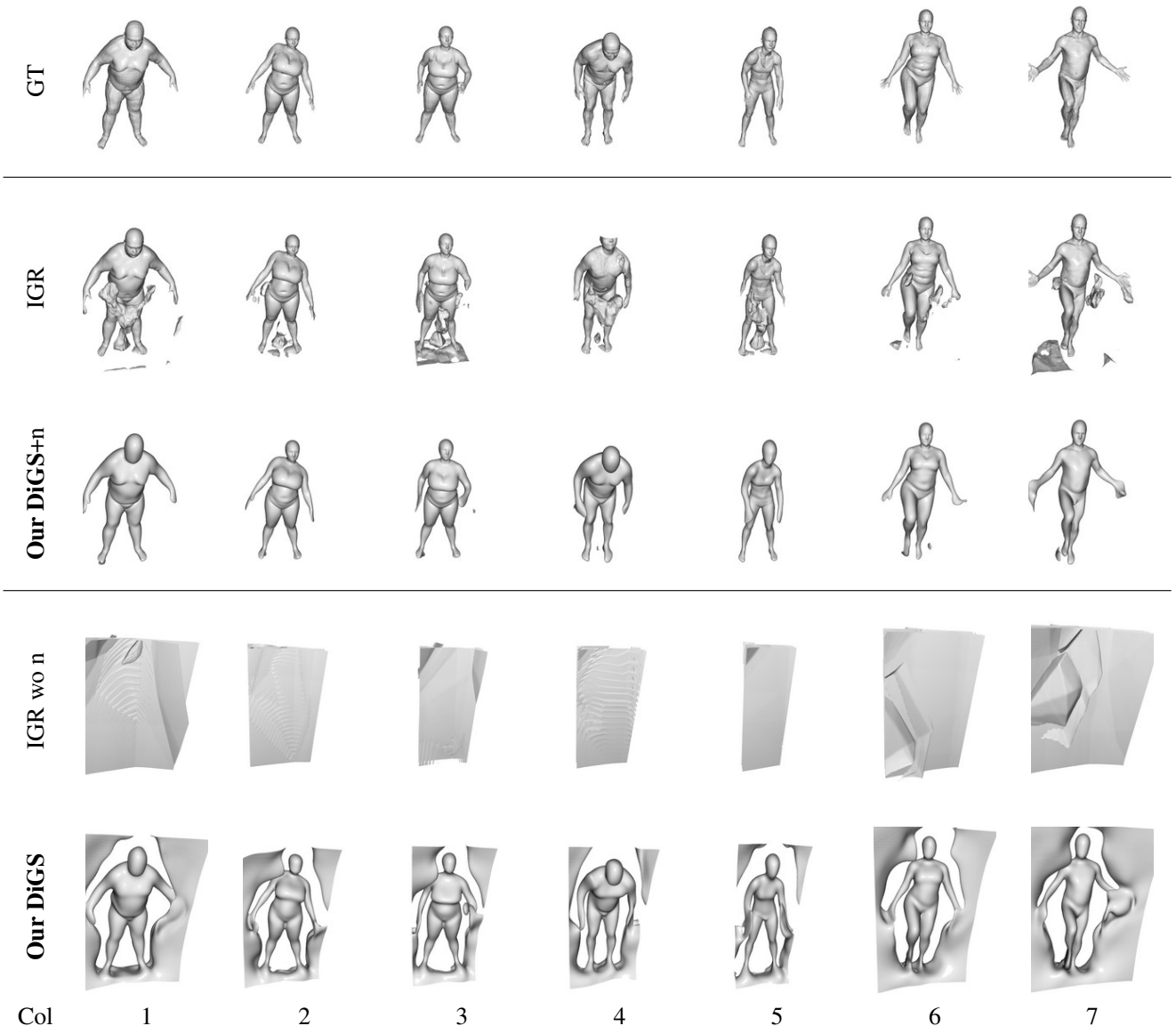

Figure 21. Qualitative results for the shape space experiment on the DFaust dataset [10]. Both methods with normals, IGR and DiGS+n, manage to capture the shape of the humans. IGR has more detail, but has a lot of ghost geometries (all columns), sometimes changes orientation that is far away from the mean of the dataset (column 4) and often misses large surfaces such as forearms (columns 3 and 5). DiGS+n captures the correct surface with minimal ghost geometry and few missing regions, but oversmooths fine details (e.g. facial features). On the other hand, for methods without normals (DiGS and IGR wo n), only DiGS is able to learn multiple human shapes whereas IGR wo n is not able to learn at all. DiGS also has large ghost geometries and oversmooths the human surfaces, but manages to capture more key regions compared to IGR.