Supplementary Material: Stabilizing the Optimization of Neural Signed Distance Functions and Finer Shape Representation

Anonymous Author(s)
Affiliation
Address
email

1 Additional experiment results and details

1.1 Initialization of quadratic neurons

One layer in our network is represented by:

$$\mathbf{z}(\mathbf{x}) = \sigma\left[l_1(\mathbf{x}) \cdot l_2(\mathbf{x}) + l_3(\mathbf{x}^2)\right], \tag{1}$$

where $\cdot$ denotes the elementwise product. σ is the activation function, where we use the sinusoidal function as in SIREN[1]. l_i represents the ith linear layer, which could be implemented by a standard linear layer module in PyTorch. There are two initializations proposed in DiGS[2] for shape INRs with linear neuron and sinusoidal activation, geometric initialization and multi-frequency geometric initialization (MFGI). In order to utilize the initializations designed for linear neurons, we initialize a quadratic neuron to approximately a linear neuron by setting $\mathbf{w}_1$, $\mathbf{w}_3$, $\mathbf{b}_3$ to a very small value and $\mathbf{b}_1$ to 1. Then we apply the above-mentioned initializations to the l_2 layer. In this way, the initialization of a quadratic neuron is approximate to the initialization of a single linear neuron.

1.2 Testing process

We use the marching cube algorithm[3] to extract the zero level set of the shape INR. The resolution is 512 and we use the same mesh generation procedure as in IGR[4].

1.3 Surface Reconstruction Benchmark(SRB)

1.3.1 Training details

We use the preprocessing and evaluation method from DiGS[2] for the daraset. First, the input clouds are centered to zero and normalized the largest norm to 1. The bounding box is 1.1 times the size of the shape. In 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 10k iterations with a learning rate of 1e-4. The weights for loss terms are $[50, 2000, 100, 100]$ for $[\alpha_e, \alpha_m, \alpha_n, \alpha_l]$. We use the annealing strategy for the weight of second-order regularization so that it will drop linearly to zero from the 2kth to the 4kth iteration. The network has 5 hidden layers and 128 channels. The initialization for the l_2 neuron is MFGI. The experiment is done on a single Tesla V100 16G GPU.

1.3.2 Additional quantitative results

In Table 1 we provide a quantitative result of our method for each shape in the SRB dataset and compare it against other SoTA methods. we report the result for SAL from [5], IGR+FF and PHASE+FF from [6], IGR wo n/SIREN wo n/ DiGS from [2]. It shows that we achieve overall improved performance that other methods.

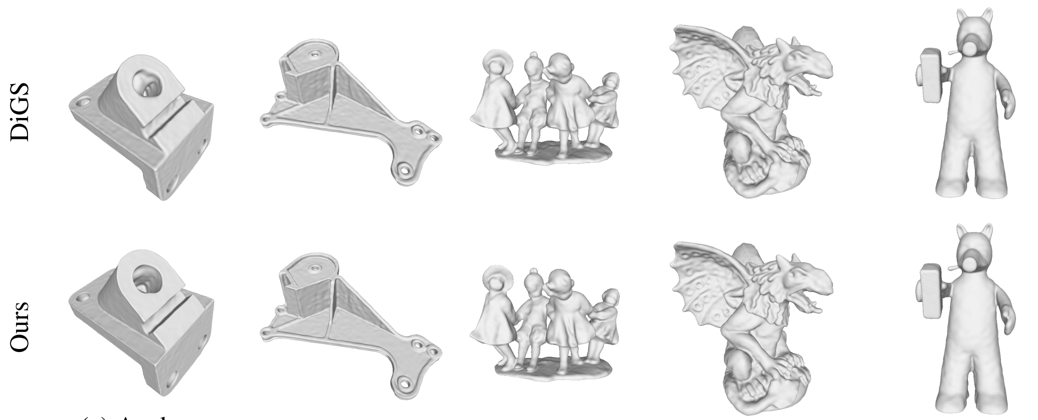
Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

		Ground Truth		Scans	
Model	Method	d_C	d_H	d_C	d_H
Overall	IGR wo n	1.38	16.33	0.25	2.96
	SIREN wo n	0.42	7.67	0.08	**1.42**
	SAL	0.36	7.47	0.13	3.50
	IGR+FF	0.96	11.06	0.32	4.75
	PHASE+FF	0.22	4.96	**0.07**	1.56
	DiGS	0.19	3.52	0.08	1.47
	Our StEik	**0.18**	**2.80**	0.10	1.45
Anchor	IGR wo n	0.45	7.45	0.17	4.55
	SIREN wo n	0.72	10.98	0.11	1.27
	SAL	0.42	7.21	0.17	4.67
	IGR+FF	0.72	9.48	0.24	8.89
	PHASE+FF	0.29	7.43	**0.09**	1.49
	DiGS	0.29	7.19	0.11	1.17
	Our StEik	**0.26**	**4.26**	0.13	**1.12**
Daratech	IGR wo n	4.9	42.15	0.7	3.68
	SIREN wo n	0.21	4.37	0.09	1.78
	SAL	0.62	13.21	0.11	2.15
	IGR+FF	2.48	19.6	0.74	4.23
	PHASE+FF	0.35	7.24	**0.08**	**1.21**
	DiGS	0.20	3.72	0.09	1.80
	Our StEik	**0.18**	**1.72**	0.10	1.77
DC	IGR wo n	0.63	10.35	0.14	3.44
	SIREN wo n	0.34	6.27	0.06	**2.71**
	SAL	0.18	3.06	0.08	2.82
	IGR+FF	0.86	10.32	0.28	3.98
	PHASE+FF	0.19	4.65	**0.05**	2.78
	DiGS	**0.15**	**1.70**	0.07	2.75
	Our StEik	0.16	1.73	0.08	2.77
Gargoyle	IGR wo n	0.77	17.46	0.18	2.04
	SIREN wo n	0.46	7.76	0.08	**0.68**
	SAL	0.45	9.74	0.21	3.84
	IGR+FF	0.26	5.24	0.18	2.93
	PHASE+FF	**0.17**	4.79	**0.07**	1.58
	DiGS	**0.17**	**4.10**	0.09	0.92
	Our StEik	0.18	4.49	0.10	0.87
Lord Quas	IGR wo n	0.16	4.22	0.08	1.14
	SIREN wo n	0.35	8.96	0.06	**0.65**
	SAL	0.13	4.14	0.07	4.04
	IGR+FF	0.49	10.71	0.14	3.71
	PHASE+FF	**0.11**	**0.71**	**0.05**	0.74
	DiGS	0.12	0.91	0.06	0.70
	Our StEik	0.13	1.81	0.07	0.73

Table 1: Additional quantitative results on the Surface Reconstruction Benchmark[7] using only point data (no normals).

1.3.3 Additional visual results

In Figure 1 we provide visualization results for all shapes in SRB. The improvement is not so dramatic compared to DiGS because this SRB is a relatively easy task without many thin structures and complex structures, and DiGS already has a good performance. In the Anchor shape, which is the most difficult one, the edges are much sharper, and the hole is recovered much better in our reconstruction result.



(a) Anchor (b) Daratech (c) DC (d) Garagoyle (e) Lord Quas

Figure 1: Visual results of SRB.

1.4 ShapeNet

1.4.1 Training details

We use the preprocessing and evaluation method from [8]. They first preprocess using the method from [9], 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[10], 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. In 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 10k iterations with a learning rate of 5e-5. The weights for loss terms are $[50, 5000, 100, 100]$ for $[\alpha_e, \alpha_m, \alpha_n, \alpha_l]$. We use the annealing strategy for the weight of second-order loss so that it will drop linearly to zero from the 2kth to the 4kth iteration. The network has 5 hidden layers and 128 channels. The initialization for the l_2 neuron is MFGI. The experiment is done on a single Tesla A100 80G GPU.

1.4.2 Additional quantitative results

In Table 2, we provide a quantitative result of our method for each shape in the ShapeNet dataset and compare it against other SoTA methods. we report the result for SAL from [5], SIREN wo n and DiGS from [2]. It shows that we achieve the best performance for most of the shapes.

1.4.3 Additional visual results

In Figure 2, we provide visualization results for some shapes in ShapeNet. Our method could remove some ghost geometries in lamps and benches, and recover complex topological structures like chair feet.

1.5 Scene Reconstruction

1.5.1 Training details

In 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 100k iterations with a learning rate of 8e-6. The weights for loss terms are $[50, 5000, 100, 10]$ for $[\alpha_e, \alpha_m, \alpha_n, \alpha_l]$. We use the annealing strategy for the weight of second-order loss so that it will drop linearly to zero from the 10kth to the 30kth iteration. The network has 8 hidden layers and 256 channels. The initialization for the l_2 neuron is the initialization method proposed in SIREN[1]. The experiment is done on a single Tesla A100 80G GPU.

Squared Chamfer

	Overall			airplane			bench		
Methods	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std
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
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
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
Ours	**6.86e-5**	**6.33e-6**	3.34e-4	**3.33e-6**	**2.59e-6**	1.78e-6	**7.90e-6**	**5.27e-6**	9.63e-6

	cabinet			car			chair		
Methods	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std
SIREN wo n	3.16e-4	2.72e-4	1.72e-4	2.67e-4	2.58e-4	4.78e-5	2.63e-4	2.60e-4	1.31e-4
SAL	2.81e-4	1.86e-4	1.81e-4	4.51e-4	2.74e-4	4.36e-4	1.28e-3	2.92e-4	2.05e-3
DiGS	4.07e-4	4.45e-5	9.25e-4	7.89e-5	3.97e-5	1.10e-4	3.72e-4	2.73e-5	1.05e-3
Ours	**2.81e-5**	**1.01e-5**	3.90e-5	**3.69e-5**	**1.11e-5**	8.68e-5	**1.24e-5**	**6.51e-6**	1.37e-5

	display			lamp			loudspeaker		
Methods	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std
SIREN wo n	2.49e-4	2.20e-4	8.45e-4	6.10e-4	3.49e-4	1.04e-3	3.29e-4	3.04e-4	1.31e-4
SAL	2.56e-4	8.86e-4	4.99e-4	5.86e-3	1.29e-3	9.35e-3	4.04e-4	2.63e-4	4.50e-4
DiGS	**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
Ours	4.62e-5	**6.97e-6**	1.69e-4	**5.75e-5**	**4.94e-6**	1.59e-4	3.12e-4	**2.79e-5**	5.56e-4

	rifle			sofa			table		
Methods	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std
SIREN wo n	5.44e-4	5.56e-4	1.44e-4	2.72e-4	2.66e-4	6.75e-5	**2.29e-4**	2.38e-4	8.40e-5
SAL	2.18e-3	1.15e-4	5.17e-3	3.75e-4	1.93e-4	4.31e-4	1.82e-3	5.10e-4	4.31e-3
DiGS	9.10e-6	5.26e-6	1.03e-5	5.76e-5	3.27e-5	5.39e-5	2.94e-4	2.98e-5	6.76e-4
Ours	**2.37e-6**	**2.03e-6**	1.40e-6	**1.23e-5**	**8.00e-6**	1.27e-5	3.62e-4	**9.80e-6**	8.76e-4

	telephone			watercraft		
Methods	Mean	Median	Std	Mean	Median	Std
SIREN wo n	2.10e-4	1.86e-4	6.60e-5	2.97e-4	2.43e-4	1.26e-4
SAL	1.04e-4	6.81e-5	7.99e-5	8.08e-4	2.06e-4	1.75e-3
DiGS	1.77e-5	1.74e-5	4.49e-6	6.10e-5	2.43e-5	9.03e-5
Ours	**5.53e-6**	**4.63e-6**	2.61e-6	**6.13e-6**	**4.25e-6**	6.53e-6

IoU

	Overall			airplane			bench		
Methods	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std
SIREN wo n	0.3085	0.2952	0.2014	0.2248	0.1735	0.1103	0.4020	0.4231	0.1953
SAL	0.4030	0.3944	0.2722	0.1908	0.1693	0.0955	0.2260	0.2311	0.1401
DiGS	0.9390	0.9754	0.1262	0.9613	0.9577	0.0164	0.9061	0.9536	0.1413
Ours	**0.9671**	**0.9841**	0.0878	**0.9814**	**0.9827**	0.0073	**0.9607**	**0.9756**	0.0493

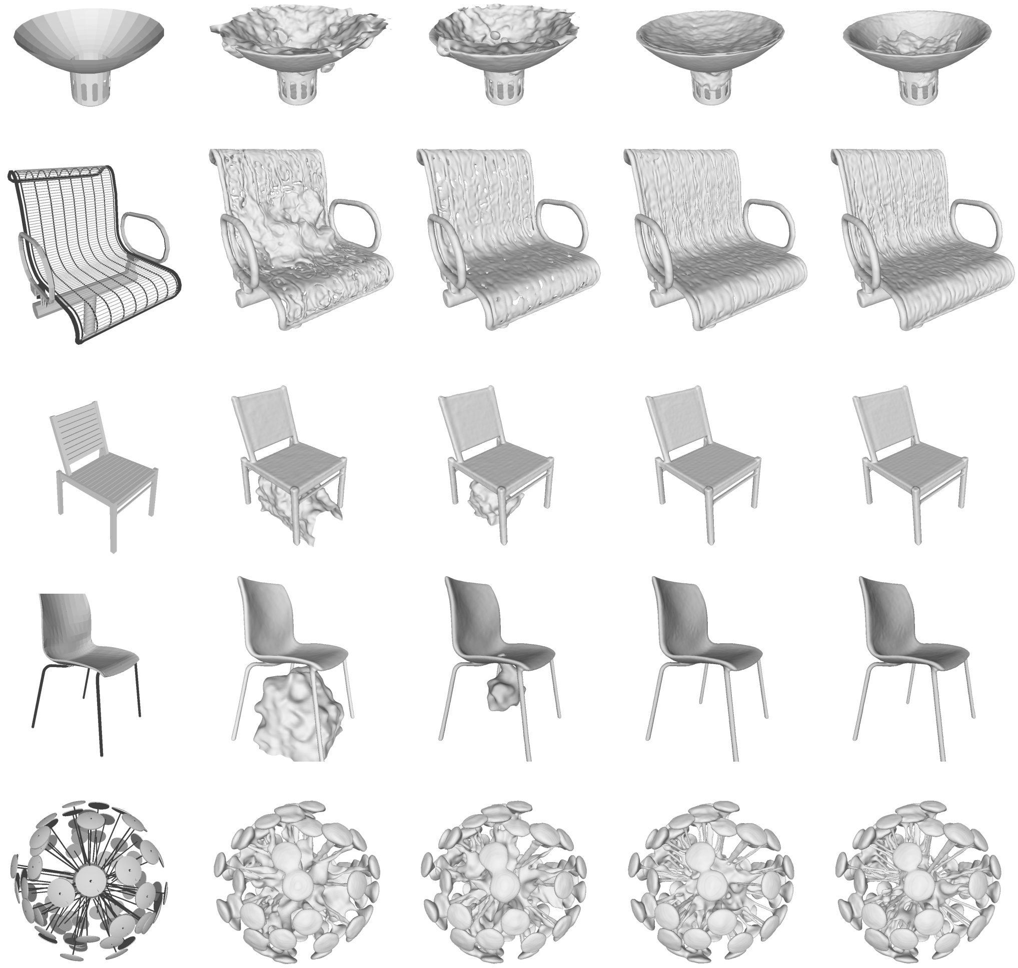
	cabinet			car			chair		
Methods	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std
SIREN wo n	0.3014	0.2564	0.1275	0.3336	0.3030	0.0997	0.4208	0.3748	0.2322
SAL	0.6923	0.7224	0.1637	0.6261	0.6561	1525	0.2589	0.1491	0.2213
DiGS	0.9261	0.9853	0.2137	0.9455	0.9765	0.0699	0.9082	0.9650	0.1523
Ours	**0.9889**	**0.9902**	0.0053	**0.9624**	**0.9842**	0.0621	**0.9754**	**0.9767**	0.0150

	display			lamp			loudspeaker		
Methods	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std
SIREN wo n	0.3566	0.3123	0.1790	0.3055	0.2573	0.2598	0.2229	0.1724	0.1575
SAL	0.5067	0.5801	0.2474	0.1689	0.0698	0.1994	0.6702	0.7264	0.1976
DiGS	0.9839	0.9886	0.0102	0.8776	0.9646	0.1943	0.9632	0.9851	0.0978
Ours	**0.9850**	**0.9870**	0.0084	**0.9290**	**0.9776**	0.1337	**0.9710**	**0.9877**	0.0681

	rifle			sofa			table		
Methods	Mean	Median	Std	Mean	Median	Std	Mean	Median	Std
SIREN wo n	0.0265	0.0092	0.0554	0.3397	0.3444	0.1206	0.3797	0.3603	0.1528
SAL	0.2835	0.2821	0.1530	0.4844	0.4530	0.1404	0.0965	0.0320	0.1502
DiGS	0.9486	0.9567	0.0281	0.9572	0.9807	0.0896	**0.8943**	0.9720	0.1996
Ours	**0.9772**	**0.9830**	0.0123	**0.9859**	**0.9894**	0.0089	0.8830	**0.9742**	0.2446

	telephone			watercraft		
Methods	Mean	Median	Std	Mean	Median	Std
SIREN wo n	0.3778	0.3806	0.2590	0.3190	0.3007	0.1877
SAL	0.6025	0.6704	0.2203	0.4170	0.4728	0.2367
DiGS	0.9854	0.9876	0.0071	0.9522	0.9735	0.0504
Ours	**0.9866**	**0.9883**	0.0051	**0.9858**	**0.9894**	0.0090

Table 2: Additional quantitative results on the ShapeNet dataset[10] using only point data (no normals).



(a) Ground Truth (b) DiGS (c) Linear + $L_{\text{L.n.}}$ (d) Quadratic + L_{div} (e) Ours

Figure 2: Additional visual results of ShapeNet.

1.5.2 Additional visual result

In Figure 3, we provide more visual results for the scene reconstruction from different angles in a higher resolution. It's clear to see that our method could recover more thin structures and fine details.

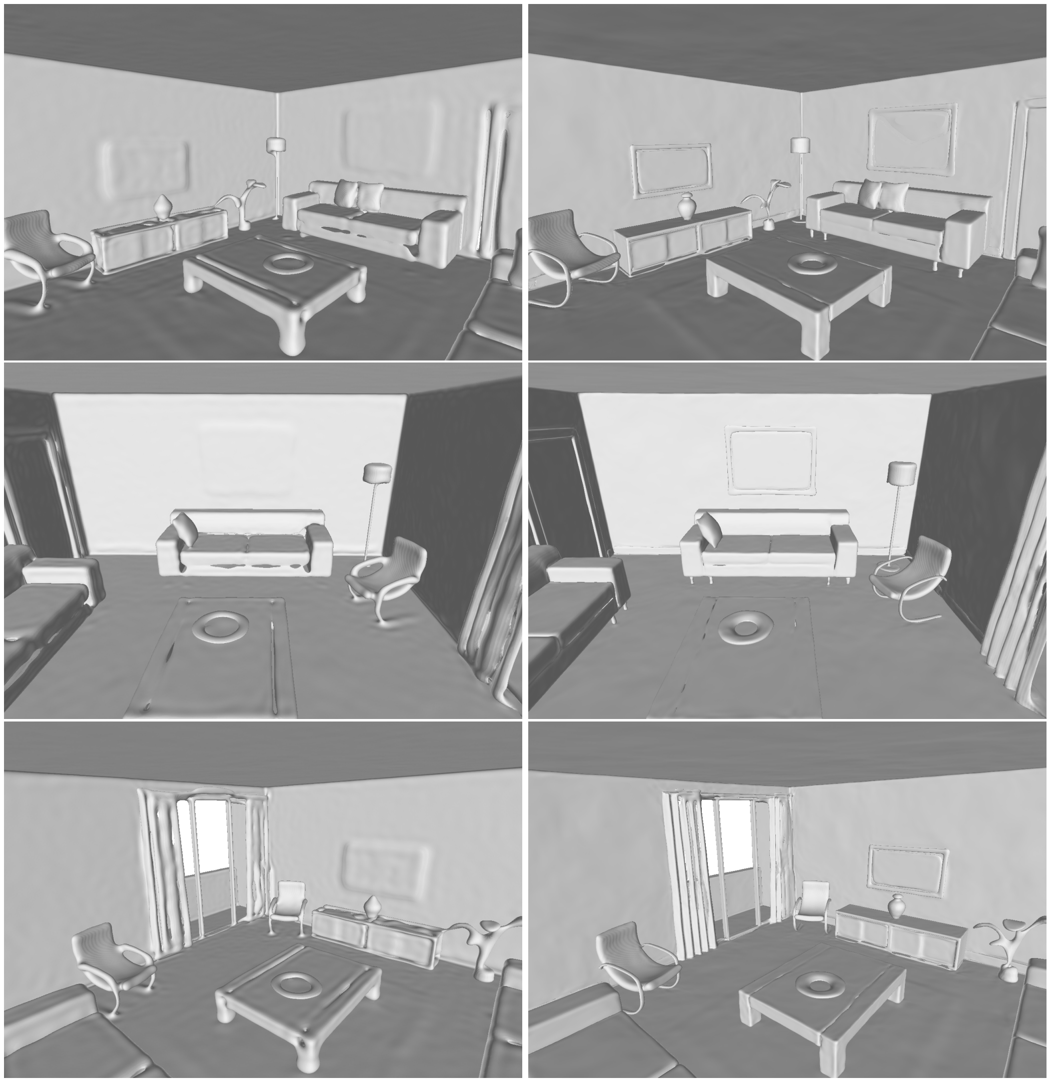
2 Derivations of functional gradients

2.1 Gradients for $L_{\text{div}}(u) = \int_\Omega |\Delta u(x)|^p \, \mathrm{d}x$

When $p = 2$ and adding a factor $\frac{1}{2}$, we have

$$\begin{aligned} -\nabla_u L_{\text{div}} &= -\nabla^2 \cdot \frac{\partial L}{\partial (D^2 u)} \\ &= -\nabla^2 \cdot (\Delta u \mathbf{I}) \\ &= -\Delta[\Delta u] \end{aligned} \tag{2}$$

where L denotes the integrand and $\mathbf{I}$ an identity matrix. The first equations comes from the Euler-Lagrange equation and the first zero and first order parts are eliminated.



(a)DiGS (b)Ours

Figure 3: Visual results of scene reconstruction.

For $p = 1$, the derivation is similar as follows,

$$\begin{aligned} -\nabla_u L_{\text{div}} &= -\nabla^2 \cdot \frac{\partial L}{\partial (D^2 u)} \\ &= -\nabla^2 \cdot \left(\frac{\Delta u \mathbf{I}}{|\Delta u|} \right) \\ &= -\Delta[\operatorname{sgn}(\Delta u)] \end{aligned} \tag{3}$$

2.2 Gradient for $L_{\text{L.n.}}(u) = \int_\Omega |\nabla u(x)^T D^2 u(x) \nabla u(x)| \, \mathrm{d}x$

In the implementation, we normalize the gradient of u to reduce the weight tunning overheads. This formula is converted as

$$L_{\text{L.n}}(u) = \int_\Omega \left| \frac{\nabla u(x)^T D^2 u(x) \nabla u(x)}{\|\nabla u\|^2} \right| \mathrm{d}x \tag{4}$$

However, we note that these two expressions are equivalent when the eikonal loss is minimized. We use the unnormalized version to compute the gradient for simplicity. One may notice that the inner part of equation (4) computes the second order derivative along the normal direction, which equals to the divergence subtracting the orthonormal tangential components

$$\Delta u - \sum_{i}^{n-1} t_i^T D^2 u t_i$$

where $t_i,\ i = 1, ..., n-1$ denotes the orthonormal tangent vectors that span the tangent subspace. Hence we could rewrite equation (4) as

$$L_{\text{L.n.}}(u) = \int_\Omega \left| \Delta u - \sum_{i}^{n-1} t_i^T D^2 u t_i \right| \,\mathrm{d}x \tag{5}$$

The negative gradient can be computed using Euler-Lagrange equation as follows,

$$-\nabla_u L_{\text{L.n.}} = \underbrace{\nabla \cdot \frac{\partial L}{\partial(\nabla u)}}_{1} - \underbrace{\nabla^2 \cdot \frac{\partial L}{\partial(D^2 u)}}_{2} \tag{6}$$

Only the term 2 in equation (6) would contain a fourth order term. Therefore we expand term 2 in the following,

$$\begin{aligned}
\nabla^2 \cdot \frac{\partial L}{\partial(D^2 u)} &= \nabla^2 \cdot \left(\frac{\Delta u - \sum_i^{n-1} t_i^T D^2 u t_i}{|\Delta u - \sum_i^{n-1} t_i^T D^2 u t_i|} \left(\frac{\partial(\Delta u)}{\partial(D^2 u)} - \frac{\partial\left(\sum_i^{n-1} t_i^T D^2 u t_i\right)}{\partial(D^2 u)} \right) \right) \\
&= \nabla^2 \cdot \left(\frac{\Delta u - \sum_i^{n-1} t_i^T D^2 u t_i}{|\Delta u - \sum_i^{n-1} t_i^T D^2 u t_i|} \left(\mathbf{I} - \frac{\partial\left(\sum_i^{n-1} t_i^T D^2 u t_i\right)}{\partial(D^2 u)} \right) \right)
\end{aligned} \tag{7}$$

From Δu and $\mathbf{I}$, we can get $\nabla^2 \cdot (\Delta u) = \Delta[\Delta u]$ as mentioned in the paper, factored by $\frac{1}{|\Delta u - \sum_i^{n-1} t_i^T D^2 u t_i|}$.

2.3 Gradients for $L_{\text{eik}}(u) = \frac{1}{2}\int_\Omega |\|\nabla u\| - 1|^p \,\mathrm{d}x$

The negative gradient for the above equation (for $p = 2$) is

$$\begin{aligned}
-\nabla_u L_{\text{eik}} &= -\frac{\partial L}{\partial u} + \nabla \cdot \frac{\partial L}{\partial(\nabla u)} \\
&= \nabla \cdot \frac{\partial L}{\partial(\nabla u)} \\
&= \nabla \cdot \left(\frac{\|\nabla u\| - 1}{\|\nabla u\|} \nabla u \right) \\
&= \nabla \cdot \left(\left(1 - \frac{1}{\|\nabla u\|} \right) \nabla u \right) \\
&= \left(1 - \frac{1}{\|\nabla u\|} \right) \Delta u + \nabla u \cdot \nabla \left(1 - \frac{1}{\|\nabla u\|} \right) \\
&= \left(1 - \frac{1}{\|\nabla u\|} \right) \Delta u - \nabla u \cdot \nabla \frac{1}{\|\nabla u\|} \\
&= \left(1 - \frac{1}{\|\nabla u\|} \right) \Delta u + \frac{1}{\|\nabla u\|^2} \nabla u \cdot \nabla \|\nabla u\| \\
&= \left(1 - \frac{1}{\|\nabla u\|} \right) \Delta u + \frac{1}{\|\nabla u\|} \left(\frac{\nabla u^T}{\|\nabla u\|} \left[\nabla^2 u \right] \frac{\nabla u}{\|\nabla u\|} \right) \\
&= \left(1 - \frac{1}{\|\nabla u\|} \right) (u_{\eta\eta} + \sum_i^{n-1} u_{\xi_i \xi_i}) + \frac{1}{\|\nabla u\|} u_{\eta\eta}
\end{aligned} \tag{8}$$

$$
= u_{\eta\eta} + \left(1 - \frac{1}{\|\nabla u\|}\right) \sum_{i}^{n-1} u_{\xi_i \xi_i} \tag{9}
$$

where the first equality is from the Euler-Lagrange equation. We remove the variable x for simplicity. Equation (8) is demonstrated in the paper and the remaining part decomposes the second order derivatives into the normal direction η and the tangential directions ξ_i. Equation (9) shows that minimizing the squared eikonal loss comes down to a stable diffusion along the normal direction and instable diffusion in all tangential directions.

Similarly, for $p = 1$, we have

$$
\begin{aligned}
-\nabla_u L_{\text{eik}} &= -\frac{\partial L}{\partial u} + \nabla \cdot \frac{\partial L}{\partial (\nabla u)} \\
&= \nabla \cdot \left(\frac{\|\nabla u\| - 1}{|\|\nabla u\| - 1|} \frac{\nabla u}{\|\nabla u\|} \right) \\
&= \nabla \cdot \left(\frac{\text{sgn}\left(\|\nabla u\| - 1\right)}{\|\nabla u\|} \nabla u \right)
\end{aligned} \tag{10}
$$

$$
\begin{aligned}
&= \frac{\|\nabla u\| - 1}{|\|\nabla u\| - 1|\,\|\nabla u\|} \Delta u + \nabla u \cdot \nabla \left(\frac{\|\nabla u\| - 1}{|\|\nabla u\| - 1|\,\|\nabla u\|} \right) \\
&= \frac{\left(\|\nabla u\| - 1\right) \Delta u + \nabla u \cdot \left(\nabla \|\nabla u\| - \frac{\|\nabla u\| - 1}{\|\nabla u\|} \nabla \|\nabla u\| - \frac{(\|\nabla u\| - 1)^2}{|\|\nabla u\| - 1|^2} \nabla \|\nabla u\| \right)}{|\|\nabla u\| - 1|\,\|\nabla u\|} \\
&= \frac{\left(\|\nabla u\| - 1\right) \Delta u - \frac{\|\nabla u\| - 1}{\|\nabla u\|} \nabla u \cdot \nabla \|\nabla u\|}{|\|\nabla u\| - 1|\,\|\nabla u\|} \\
&= \frac{\left(\|\nabla u\| - 1\right) \left(\Delta u - \frac{\nabla u}{\|\nabla u\|} \cdot \nabla \|\nabla u\| \right)}{|\|\nabla u\| - 1|\,\|\nabla u\|} \\
&= \frac{\text{sgn}\left(\|\nabla u\| - 1\right)}{\|\nabla u\|} \left(\Delta u - \frac{\nabla u^T}{\|\nabla u\|} \left[\nabla^2 u\right] \frac{\nabla u}{\|\nabla u\|} \right) \\
&= \frac{\text{sgn}\left(\|\nabla u\| - 1\right)}{\|\nabla u\|} \left(\left(\sum_{i}^{n-1} u_{\xi_i \xi_i} + u_{\eta\eta}\right) - u_{\eta\eta} \right) \\
&= \frac{\text{sgn}\left(\|\nabla u\| - 1\right)}{\|\nabla u\|} \sum_{i}^{n-1} u_{\xi_i \xi_i}
\end{aligned} \tag{11}
$$

Comparing against $p = 2$, the absolute value of the eikonal loss ($p = 1$) leads to instable diffusion along all tangential directions and no constraints in the normal direction.

2.4 Gradients for $L_{\text{norm.}}(u) = \int_{\Omega_o} \|\nabla u(x) - N_{gt}\|^p \, \mathrm{d}x$

For $p = 2$, we could get

$$
\begin{aligned}
-\nabla_u L_{\text{norm.}} &= -\frac{\partial L}{\partial u} + \nabla \cdot \frac{\partial L}{\partial (\nabla u)} \\
&= 2 \nabla \cdot \left(\nabla u - N_{gt} \right) \\
&= 2 \Delta u - 2 \text{div}\left(N_{gt}\right)
\end{aligned} \tag{12}
$$

Not that factor 2 was omitted in the full paper for simplicity. For $p = 1$, we have

$$\begin{aligned}
-\nabla_u L_{\text{norm.}} &= -\frac{\partial L}{\partial u} + \nabla \cdot \frac{\partial L}{\partial(\nabla u)} \\
&= \nabla \cdot \left(\frac{\nabla u - N_{gt}}{\|\nabla u - N_{gt}\|} \right) \\
&= \Delta u - \nabla \left(\frac{1}{\|\nabla u - N_{gt}\|} \right) \cdot N_{gt} \\
&= \Delta u + \frac{N_{gt}^T D^2 u \nabla u}{\|\nabla u - N_{gt}\|^3}
\end{aligned} \tag{13}$$

3 Choices of p in the eikonal loss

3.1 Influence on the instability

Given the equations (11) and (9), both exhibit instability in the tangential directions. While the coefficients of the diffusion terms are different, it is not straightforward to justify the effectiveness of one over the other. However, we show empirically that $p = 1$ achieves better results on SRB, present in the next subsection.

3.2 Ablation Study of the performance

We investigate the effects of design choices made for regularization terms and report the averages over all shapes in the dataset in Table 3. We demonstrate that if we choose $p = 1$ for both first-order and second-order regularization, the algorithm will achieve the best performance.

		GT		Scans	
L_{eik}	$L_{\text{L.n.}}$	d_C	d_H	$d_{\vec{C}}$	$d_{\vec{H}}$
L1	L1	**0.180**	**2.800**	**0.096**	**1.454**
L1	L2	0.205	4.389	0.105	1.486
L2	L1	0.194	3.917	0.469	1.486
L2	L2	0.217	4.844	0.093	1.483

Table 3: Ablation study of regularization terms on SRB[7]

4 Additional results on the eikonal instability

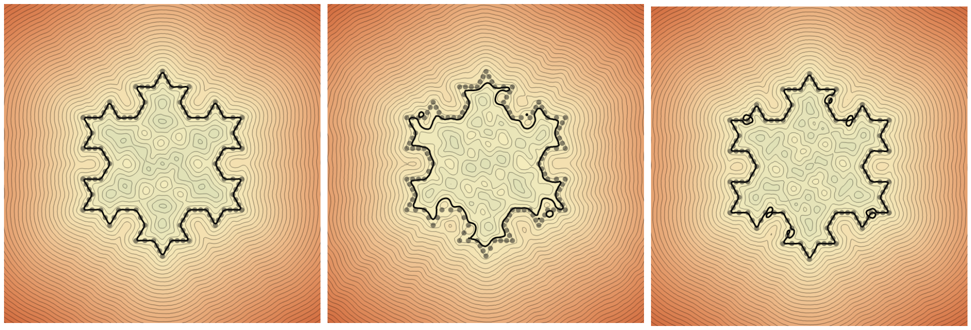
We showed in Fig. 1 (in the full paper), with quadratic networks, the instability incurred by the eikonal loss when divergence terms are removed. Linear networks, though even less complex, will encounter the eikonal instability as well according to our analysis. We demonstrate additional results in Fig. 4 with linear networks and SIREN.

5 Ablation study on regularization weight

We have conducted this experiment on SRB varying regularization weights. It shows that around the optimal weight choice, the results are not sensitive. Furthermore, increasing the weight beyond the optimal, only degrades results slightly since that simply enforces further a constraint that is true of all SDFs, without smoothing the geometry much. This is consistent Lagrange multiplier theory for enforcing a constraint into the optimization problem.

6 Ablation study on SRB dataset of Regularizations & Linear vs Quad Layers

In Table 5, we study the effectiveness of each of our novel contributions (the Laplacian normal regularization and quadratic layers) on the SRB dataset. We get a better result with linear networks



Almost converged | Intermediate after instability | Final

Figure 4: Instability on linear networks. (*left*) The evolution is almost converged. (*middle*) However, after several additional iterations, instability occurs. We show the intermediate results 50 steps after the instability. (*right*) This instability drives the network to a sub-optimal local minimizer.

	GT		Scans	
α_l	d_C	d_H	$d_{\vec{C}}$	$d_{\vec{H}}$
10	0.264	6.089	0.099	1.513
50	0.191	3.799	0.096	1.485
100*	0.180	2.800	0.096	1.453
200	0.188	3.520	0.097	1.495
300	0.192	3.497	0.102	1.535
400	0.187	3.177	0.102	1.537
500	0.194	3.557	0.098	1.499

Table 4: Varying α_l and performance. The relationship between α_l and the performance is not salient. We may mention that the weight needs to be tuned based on different tasks, but a relatively larger weight is preferred given the annealing strategy.

if we replace the divergence in [2] with our Laplacian normal. Also replacing linear with quadratic layers, irrespective of regularization leads to better results. The combination of the two produces the best results. All the experiments are run under the same hyper-parameters.

	GT		Scans	
Method	d_C	d_H	$d_{\vec{C}}$	$d_{\vec{H}}$
Lin+L_{div}	0.190	4.397	0.099	1.446
Lin+$L_{\text{L. n.}}$	0.188	4.321	0.100	1.498
Qua+L_{div}	0.187	3.597	0.098	1.496
Ours(Qua+$L_{\text{L. n.}}$)	**0.180**	**2.800**	0.096	1.454

Table 5: Ablation study on the Surface Reconstruction Benchmark[7] using only point data (no normals).

References

[1] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, *Implicit neural representations with periodic activation functions*, 2020. arXiv: 2006.09661 [cs.CV].

[2] Y. Ben-Shabat, C. H. Koneputugodage, and S. Gould, *Digs : Divergence guided shape implicit neural representation for unoriented point clouds*, 2022. arXiv: 2106.10811 [cs.CV].

[3] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 163–169, Aug. 1987, ISSN: 0097-8930. DOI: 10.1145/37402.37422. [Online]. Available: https://doi.org/10.1145/37402.37422.

[4] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, *Implicit geometric regularization for learning shapes*, 2020. arXiv: 2002.10099 [cs.LG].

[5] M. Atzmon and Y. Lipman, *Sal: Sign agnostic learning of shapes from raw data*, 2020. arXiv: 1911.10414 [cs.CV].

[6] Y. Lipman, *Phase transitions, distance functions, and implicit neural representations*, 2021. arXiv: 2106.07689 [cs.LG].

[7] M. Berger, J. A. Levine, L. G. Nonato, G. Taubin, and C. T. Silva, "A benchmark for surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 2, Apr. 2013, ISSN: 0730-0301. DOI: 10.1145/2451236.2451246. [Online]. Available: https://doi.org/10.1145/2451236.2451246.

[8] F. Williams, M. Trager, J. Bruna, and D. Zorin, "Neural splines: Fitting 3d surfaces with infinitely-wide neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9949–9958.

[9] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, *Occupancy networks: Learning 3d reconstruction in function space*, 2019. arXiv: 1812.03828 [cs.CV].

[10] A. X. Chang, T. Funkhouser, L. Guibas, *et al.*, *Shapenet: An information-rich 3d model repository*, 2015. arXiv: 1512.03012 [cs.GR].