GENERALIZATION ANALYSIS AND IMPROVED SHAPE REPRESENTATION WITH NEURAL SIGNED DISTANCE FUNCTIONS

Meenakshi Krishnan *
Ramani Duraiswami†

ABSTRACT

Implicit Neural Representations (INRs) that learn Signed Distance Functions (SDFs) from point cloud data are the state-of-the-art for 3D surface reconstruction. However, training these Neural SDFs often requires enforcing the Eikonal equation, an ill-posed PDE that also leads to unstable gradient flows when used as a constraint. Numerical Eikonal solvers have relied on viscosity approaches for regularization and stability. Motivated by this, we introduce ViscoReg, a regularizer that provably stabilizes Neural SDF training. Empirically, ViscoReg outperforms state-of-the-art approaches such as DiGS, StEik, and HotSpot on the Surface Reconstruction Benchmark, and 3D scene reconstruction. We also establish generalization error estimates for Neural SDFs in terms of the training error, using the theory of viscosity solutions.

1 INTRODUCTION

Reconstructing surfaces from point clouds is a long-studied problem in computer vision that is challenging due to non-uniform point sampling, noisy normal estimations, missing surface regions, and other data imperfections (Berger et al., 2017). Recently, Implicit Neural Representations (INRs) have gained popularity in volumetric representation due to their high resolution and compactness (Michalkiewicz et al., 2019; Mildenhall et al., 2021; Sitzmann et al., 2019a;b). Neural SDFs learn a function mapping spatial coordinates to the signed distance from a manifold, implicitly defining the surface as the zero level set. Early Neural SDF methods include DeepSDF (Park et al., 2019), SAL (Atzmon & Lipman, 2020a), SALD (Atzmon & Lipman, 2020b), and PHASE (Lipman, 2021).

Neural SDF are trained on the input point cloud data by constraining the signed distance to be zero on the surface, and optionally, using surface normal information. In the absence of normal information, previous methods suffer a severe degradation in reconstruction quality. Enforcing the Eikonal partial differential equation (PDE):

$$\|\nabla u(x)\|_2 = 1 \text{ for } x \in \Omega,\ u(x) = 0 \text{ for } x \in \partial\Omega, \tag{1}$$

via the *Eikonal loss* ensures that the network learns a valid SDF (Gropp et al., 2020). Here, Ω is a bounded domain, and $\partial\Omega$ is the sufficiently smooth boundary surface we aim to reconstruct. However, the Eikonal loss alone may not be enough for good reconstruction (Ben-Shabat et al., 2022), and it presents two fundamental challenges. First, training with this regularizer can cause instabilities, leading the network to converge to suboptimal local minima, as has been demonstrated both theoretically and empirically (Yang et al., 2023). Second, the Eikonal equation is inherently ill-posed, with multiple solutions. The SDF is the unique viscosity solution of the Eikonal (Crandall & Lions, 1983). This leads to an important question for Neural SDFs: With infinitely many solutions to the Eikonal equation, why is minimizing the PDE residual loss on a finite training set sufficient to ensure convergence to the unique viscosity solution (i.e., the SDF)?

Several methods have tried to address the Eikonal instability by proposing various regularizing terms. DiGS (Ben-Shabat et al., 2022) minimizes the Laplacian of the learned function. However, the SDF Laplacian corresponds to the mean curvature of the surface, and its minimization can lead to over-smoothing of fine detail. StEik (Yang et al., 2023) identified training instabilities with the Eikonal loss and proposed a directional divergence regularizer. However, empirically, this overfits noise in the input. More recently, HotSpot (Wang et al., 2025) introduced a loss function derived

---

* Perceptual Interfaces and Reality Lab (PIRL), Dept. of Mathematics, University of Maryland, College Park, `mkrishn9@umd.edu`

† Perceptual Interfaces and Reality Lab (PIRL), UMIACS & Dept. of Computer Science, University of Maryland, College Park, `ramanid@umd.edu`

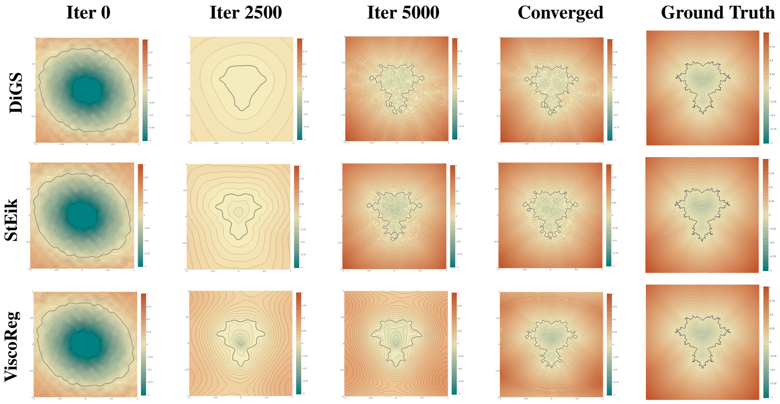


Figure 1: 2D fractal Mandelbrot reconstruction. DiGS oversmooths early on; both baselines produce disconnected, self-intersecting reconstruction. ViscoReg recovers the correct intricate boundary.

from a screened Poisson equation. For non-neural grid-based methods, ViscoGrids (Pumarola et al., 2022), proposed using a fixed, non-decaying viscosity term and a coarea loss.

To address the first challenge of the practical instability, we consider the well-posed parabolic equation, which adds a viscosity/diffusion term to the Eikonal:

$$\|\nabla u_\varepsilon\|_2 = 1 + \varepsilon \Delta u_\varepsilon. \tag{2}$$

The viscosity solution u of equation 1 is recovered in the limit $\varepsilon \to 0$ of u_ε. Motivated by these, we propose **a novel regularization technique** that incorporates a dynamically scaled viscous term into the Eikonal loss during training. This stabilizes training, and improves reconstruction quality.

To address the second challenge of theoretical ill-posedness, we establish bounds on the INR generalization error using properties of viscosity solutions, and classical PDE inequalities. Generalization theory aims to understand how well the network generalizes to unseen data given the training error. Results have been established for PINNs in abstract settings (Mishra & Molinaro, 2023; 2022; Chen et al., 2025), and for specific PDEs (De Ryck & Mishra, 2022; Hu et al., 2021; Berner et al., 2020). We extend this analysis to Neural SDFs, providing intuition on why the network should converge to the correct solution, and bounds on the worst-case deviation from the ground-truth. Our main contributions can be summarized as follows:

- Generalization error bounds are presented to validate that minimizing the PDE residual and surface data fidelity loss ensures that the estimated solution converges to the unique viscosity solution.
- We propose *ViscoReg*, a novel training regime for Neural SDF based on the vanishing viscosity method with a dynamically scaled loss. We justify this by analyzing the gradient flow of its variational formulation, and demonstrate its ability to stabilize training for high-frequency components.

2 ERROR ANALYSIS

We present results on generalization error bounds for Neural SDF. We take $\Omega \subset \mathbb{R}^3$ as an open, connected, bounded domain with sufficiently smooth boundary $\partial\Omega$. Denote, the input surface point cloud $\mathcal{P}_{\partial\Omega} := \{x_i\}_{i=1}^N \subset \partial\Omega$, and uniformly sampled non-manifold points from the domain $\mathcal{P}_\Omega := \{y_j\}_{j=1}^M \subset \Omega$. A Neural SDF u_θ is trained by minimizing the *manifold constraint* $L_m(u_\theta) = N^{-1}\sum_{x\in\mathcal{P}_{\partial\Omega}} |u_\theta(x)|$ and the *non-manifold penalization constraint* $L_{nm}(u_\theta) = M^{-1}\sum_{y\in\mathcal{P}_\Omega} e^{-\alpha|u_\theta(y)|}$ (with $\alpha \gg 1$). These ensure that u_θ is zero on the manifold, and non-zero away from it. Additionally, the *Eikonal constraint* $L_{eik}(u_\theta) = (M+N)^{-1}\sum_{z\in\mathcal{P}_{\partial\Omega}\cup\mathcal{P}_\Omega} |\|\nabla u_\theta(z)\|_2 - 1|^p$ for $p \in \{1,2\}$ specifies the norm of the gradient to be one is enforced. The optimization problem with hyperparameters $\alpha_m,\ \alpha_{nm},\ \alpha_e$ and $\alpha >> 1$ is:

$$\arg\min_\theta \left(\alpha_m \mathcal{L}_m(u_\theta) + \alpha_{nm}\mathcal{L}_{nm}(u_\theta) + \alpha_e \mathcal{L}_{eik}(u_\theta)\right). \tag{3}$$

The generalization error estimate is provided when the L^1 norm is used for the Eikonal loss (L^2 case in the appendix). Let $\theta^* \in \mathbb{R}^d$ be the minimizer of (3) and let $u_{\theta^*} \in C^\infty(\bar{\Omega})$ be the corresponding network. As we use the sine activation function, u_{θ^*} is smooth. It is as an exact solution to a perturbed Eikonal equation, where the PDE residual corresponds to $f_{\theta^*} \in C^\infty(\bar{\Omega})$ and the boundary error becomes $g_{\theta^*} \in C^\infty(\partial\Omega)$:

$$\|\nabla u_{\theta^*}(x)\|_2 = f_{\theta^*}(x),\ x \in \Omega,\ u_{\theta^*}(x) = g_{\theta^*}(x),\ x \in \partial\Omega. \tag{4}$$

We assume that the network $u_{\theta^*}(x)$ satisfies the following conditions.

Assumption 1: The gradient of u_{θ^*} is bounded away from zero. Specifically, for all $x \in \Omega$, we have $0 < \frac{1}{C_{\theta^*}} \leq \|\nabla u_{\theta^*}(x)\|_2 \leq C_{\theta^*}$, for $C_{\theta^*} > 0$.

If θ^* is a sufficiently good local minima, it is natural that Assumption 1 holds, since the ground-truth SDF u satisfies $\|\nabla u(x)\|_2 = 1 > 0, \ \forall \, x \in \Omega$.

Assumption 2: The input point cloud $\mathcal{P}_{\partial\Omega} = \{x_i\}_{i=1}^{N}$ satisfies the quadrature error bound:

$$\left| \int_{\tilde{\Omega}} |g(x)|^p dx - \frac{1}{N} \sum_{i=1}^{N} |g(x_i)|^p \right| \leq C_g N^{-\beta}, \tag{5}$$

for $p = 1, 2$, and $\beta > 0$. This assumption is quite general, and $\beta = 1/3$ for uniform sampling, and $\beta = 1/2$ for Monte-Carlo sampling of sufficiently smooth functions (Mishra & Molinaro, 2023).

Note Ω is bounded, and the network has bounded weights, $\|u_\theta\|_{C^k(\Omega)} \leq C_k < \infty$ for all finite k. This implies that the network (and its derivatives) is bounded in $W^{k,p}$ norm for finite k and $1 \leq p \leq \infty$. Denote $\|f_{\theta^*}\|_{W^{6,1}(\Omega)}, \ \|g_{\theta^*}\|_{W^{6,1}(\partial\Omega)} \leq M_{\theta^*}$ for $M_{\theta^*} > 0$. Here, the choice of $k = 6$ is determined by the requirements of the interpolation inequality used in the proof of Theorem 1.

Theorem 1 *Suppose Assumptions 1-2 hold. Consider the minimizer $\theta^* \in \mathbb{R}^d$ of (3) and let $u_{\theta^*} \in C^\infty(\bar{\Omega})$ be the network parametrized by θ^*. Let $u \in C(\bar{\Omega})$ be the solution to equation 1. Then, the generalization error is bounded as:*

$$\|u - u_{\theta^*}\|_\infty \lesssim M_{\theta^*} (L_m(u_{\theta^*}))^{\frac{1}{2}} + M_{\theta^*} C_{\theta^*}^{-2} \left(L_{eik}(u_{\theta^*})\right)^{\frac{1}{2}} + \mathcal{O}((M+N)^{-1/6}) + \mathcal{O}(N^{-\frac{\beta}{2}}). \tag{6}$$

The constants in $\lesssim$ depend only on $\bar{\Omega}$.

The proof is in the appendix. The generalization bound may seem expected, as it suggests that small training error leads to better generalization. However, this result is non-trivial in the context of PDE solutions, where there is no fundamental reason why minimizing the PDE residual and boundary loss at finitely many points should lead the network to converge to a solution of the continuous formulation of a nonlinear PDE. This is particularly insightful for the ill-posed Eikonal equation, which admits infinitely many solutions, only one of which is the viscosity solution.

3 VISCOREG

Gradient flow defines the continuous evolution of a function $u : \Omega \to \mathbb{R}$ along the path of steepest descent to minimize an energy functional $E(u)$. It is obtained in the continuum limit of the gradient descent for the minimization problem, and is given by $u_t = -\nabla E(u)$. Here, t is an artificial time parameter, and $\nabla E(u)$ is the Fréchet derivative of E with respect to u. When u is constrained to neural networks parametrized by weights, the optimization is in the finite-dimensional parameter space. Here, we study the unconstrained gradient flow to provide insight into the training process.

As shown by Yang et al. (2023), the gradient flow of the Eikonal loss is unstable. To stabilize Neural SDF training, we propose adding a decaying viscosity term to the Eikonal loss : $L(u_\theta) = \alpha_m L_m(u_\theta) + \alpha_{nm} L_{nm}(u_\theta) + \alpha_v L_{veik}(u_\theta)$. Here, L_{veik} represents the viscous Eikonal loss that we refer to as *ViscoReg* given by:

$$\mathcal{L}_{veik}(u_\theta) = \sum_{x \in \mathcal{P}_\Omega} |\|\nabla u_\theta(x)\|_2 - 1 - \varepsilon \Delta u_\theta|^p \ dx, \ \ p = 1, \ 2, \tag{7}$$

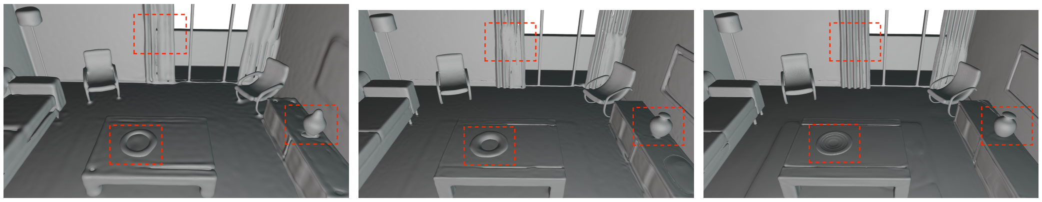
where $\varepsilon > 0$ is a hyperparameter decayed to zero during training. This is motivated by the fact that the Eikonal viscosity solution is the limit of the parabolic equation 2 (Theorem 2, Appendix).

The viscosity term lends stability to the Eikonal loss. Let $r(u) = \|\nabla u\|_2 - 1 - \varepsilon \Delta u$ be the viscous residual. For $p = 1$, computing the Frèchet derivative of $\mathcal{L}_{veik}$ gives the gradient flow equation:

$$u_t = \nabla \cdot (\text{sign}(r) \nabla u / \|\nabla u\|_2) + \varepsilon \Delta(\text{sign}(r)). \tag{8}$$

For analytical ease, we replace $\text{sign}(r)$ function with a smooth approximation (such as the sigmoid function) $S(r)$ such that $S(0) = 0$ and $S'(0) = c > 0$. Linearising the resulting non-linear PDE around its stationary solution $u_0 = \mathbf{a} \cdot x$ (with $\|\mathbf{a}\|_2 = 1$), and taking its Fourier transform:

$$\hat{u}_t = -c[(\mathbf{a} \cdot \omega)^2 + \varepsilon^2 \|\omega\|_2^4] \hat{u} \implies \hat{u} = \exp\left(-ct[(\mathbf{a} \cdot \omega)^2 + \varepsilon^2 \|\omega\|_2^4]\right). \tag{9}$$



(a) DiGS (b) StEik (quad) (c) ViscoReg (ours)

Figure 2: Scene reconstruction benchmark. DiGS (a) misses fine details (sofa legs, vase shape, picture frame), as does StEik (curtains and plate). ViscoReg (c) reconstructs details with high fidelity.

The Fourier symbol $-c[(\mathbf{a} \cdot \omega)^2 + \varepsilon^2 \|\omega\|_2^4]$ is non-positive for all wavevectors since $c > 0$. This implies that as $t \to \infty$, $\hat{u} \to 0$, and that the equation is unconditionally stable.

Similar results are presented for $p = 2$ in the appendix. Enforcing the viscous Eikonal PDE in the initial phases of training thus stabilizes training. As proof of concept, we demonstrate the boundary reconstruction of a complex Mandelbrot fractal with different methods in Fig. 1. Unlike other methods, ViscoReg smoothly converges to the highly curved boundary.

4 RESULTS

Implementation Details: We evaluate ViscoReg on the Surface Reconstruction Benchmark (Berger et al., 2013), and a scene reconstruction task from Sitzmann et al. (2019b). Our main point of comparison involves DiGS (Ben-Shabat et al., 2022), StEik (Yang et al., 2023), HotSpot (Wang et al., 2025), Neural Singular Hessian (Wang et al., 2023) and NeurCADRecon (Dong et al., 2024).

Table 1: Results on SRB. d_C : Chamfer and d_H: Hausdorff distance.

Method	$d_C \downarrow$	$d_H \downarrow$
SIREN wo n	0.42	7.67
NSH	0.20	3.73
NeurCADRecon	0.20	3.70
VisCoGrids wo n	0.34	4.39
HotSpot	0.19	3.17
DiGS	0.19	3.52
StEik (lin)	0.20	4.56
Ours (lin)	**0.18**	**2.76**

Table 2: Ablation on ε decay for mean Chamfer and Hausdorff metrics in SRB. BL$\times x$= baseline scaled by x.

Method	$d_C \downarrow$	$d_H \downarrow$
BL	**0.18**	2.76
BL $\times 2$	0.18	3.17
BL $\times 0.5$	0.19	5.06
Fast decay (0 @ 20%)	0.19	3.51
Slow decay(0 @ 90%)	0.18	3.28
Piecewise Const.	0.20	5.17
Quintic	0.19	3.89
$\varepsilon = 0$ (SIREN [47])	0.42	7.67

Surface Reconstruction Benchmark (SRB): We evaluate on the five noisy shapes in SRB, comparing againt normal-free versions of SoTA. Results for the Chamfer and Hausdorff distances between ground-truth meshes are in Tab. 1. Our method improves upon all others, with considerable improvement in the Hausdorff distance, even though we use approximately 50% fewer parameters than DiGS. StEik combines a directional divergence regularizer with quadratic layers. To isolate the effect of the regularizer for a fair comparison, we evaluate StEik using standard linear layers.

Viscosity parameter decay ablation: Baseline decay for all shapes is initial $\varepsilon = 0.5$, decayed linearly at 20/40/60/80% iterations to 0.4/0.04/0.005/0 for ViscoReg (linear). See Tab. 2 for ablation. Many reasonable decays work well. Ablation decay schedules are provided in the appendix.

Scene Reconstruction from Sitzmann et al. (2019b): We use 8 layers and 512 neurons with 10M sample points as in the original dataset. Qualitative results are in Fig. 2. Due to the smoothing effect of the Laplacian term, DiGS does not recover fine details such as sofa legs, vase and picture frames. StEik recovers details somewhat better but still struggles with more intricate detailing like picture frames, curtains, and the plate rim. Our method recovers the fine details reconstructing the surface with greater fidelity, even though we do not use normal information or quadratic layers.

5 CONCLUSION

We leverage classical PDE theory to provide an estimate on the worst case error when using neural networks to approximate the SDF. We also propose a physically-motivated regularizing term (ViscoReg) and show improved performance compared to the SoTA.

ACKNOWLEDGEMENTS

MK and RD were supported by ONR Award N00014-23-1-2086.

REFERENCES

Matan Atzmon and Yaron Lipman. SAL: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2565–2574, 2020a.

Matan Atzmon and Yaron Lipman. SALD: Sign agnostic learning with derivatives. *arXiv preprint arXiv:2006.05400*, 2020b.

Yizhak Ben-Shabat, Chamin Hewa Koneputugodage, and Stephen Gould. DiGS: Divergence guided shape implicit neural representation for unoriented point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19323–19332, 2022.

Matthew Berger, Joshua A Levine, Luis Gustavo Nonato, Gabriel Taubin, and Claudio T Silva. A benchmark for surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(2):1–17, 2013.

Matthew Berger, Andrea Tagliasacchi, Lee M Seversky, Pierre Alliez, Gael Guennebaud, Joshua A Levine, Andrei Sharf, and Claudio T Silva. A survey of surface reconstruction from point clouds. In *Computer Graphics Forum*, volume 36, pp. 301–329. Wiley Online Library, 2017.

Julius Berner, Philipp Grohs, and Arnulf Jentzen. Analysis of the generalization error: Empirical risk minimization over deep artificial neural networks overcomes the curse of dimensionality in the numerical approximation of black–scholes partial differential equations. *SIAM Journal on Mathematics of Data Science*, 2(3):631–657, 2020.

Jeff Calder. Lecture notes on viscosity solutions. *Lecture notes*, 2018.

Ke Chen, Meenakshi Krishnan, and Haizhao Yang. Error analysis for learning the time-stepping operator of evolutionary pdes. *arXiv preprint arXiv:2509.04256*, 2025.

Michael G Crandall and Pierre-Louis Lions. Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American mathematical society*, 277(1):1–42, 1983.

Tim De Ryck and Siddhartha Mishra. Error analysis for physics-informed neural networks (PINNs) approximating Kolmogorov PDEs. *Advances in Computational Mathematics*, 48(6):79, 2022.

Qiujie Dong, Rui Xu, Pengfei Wang, Shuangmin Chen, Shiqing Xin, Xiaohong Jia, Wenping Wang, and Changhe Tu. Neurcadrecon: Neural representation for reconstructing cad surfaces by enforcing zero gaussian curvature. *arXiv preprint arXiv:2404.13420*, 2024.

Lawrence C Evans. *Partial differential equations*, volume 19. American Mathematical Society, 2022.

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020.

Zheyuan Hu, Ameya D Jagtap, George Em Karniadakis, and Kenji Kawaguchi. When do extended physics-informed neural networks (XPINNs) improve generalization? *arXiv preprint arXiv:2109.09444*, 2021.

Yaron Lipman. Phase transitions, distance functions, and implicit neural representations. *arXiv preprint arXiv:2106.07689*, 2021.

William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353. 1998.

Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4743–4752, 2019.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Siddhartha Mishra and Roberto Molinaro. Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs. *IMA Journal of Numerical Analysis*, 42(2):981–1022, 2022.

Siddhartha Mishra and Roberto Molinaro. Estimates on the generalization error of physics-informed neural networks for approximating PDEs. *IMA Journal of Numerical Analysis*, 43(1):1–43, 2023.

Louis Nirenberg. On elliptic partial differential equations. *Annali della Scuola Normale Superiore di Pisa-Scienze Fisiche e Matematiche*, 13(2):115–162, 1959.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 165–174, 2019.

Albert Pumarola, Artsiom Sanakoyeu, Lior Yariv, Ali Thabet, and Yaron Lipman. VisCo grids: Surface reconstruction with viscosity and coarea grids. *Advances in Neural Information Processing Systems*, 35:18060–18071, 2022.

Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3D feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2437–2446, 2019a.

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019b.

Zimo Wang, Cheng Wang, Taiki Yoshino, Sirui Tao, Ziyang Fu, and Tzu-Mao Li. Hotspot: Signed distance function optimization with an asymptotically sufficient condition. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 1276–1286, 2025.

Zixiong Wang, Yunxiao Zhang, Rui Xu, Fan Zhang, Peng-Shuai Wang, Shuangmin Chen, Shiqing Xin, Wenping Wang, and Changhe Tu. Neural-singular-hessian: Implicit neural representation of unoriented point clouds by enforcing singular hessian. *ACM Transactions on Graphics (TOG)*, 42(6):1–14, 2023.

Huizong Yang, Yuxin Sun, Ganesh Sundaramoorthi, and Anthony Yezzi. StEik: Stabilizing the optimization of neural signed distance functions and finer shape representation. *Adv. Neural Inform. Process. Syst.*, 36:13993–14004, 2023.

A APPENDIX

In this section, we provide supplementary details for our paper.

Let $\Omega \subseteq \mathbb{R}^n$ be an open, bounded domain and let $u : \Omega \to \mathbb{R}$ be a sufficiently regular function. The following norms are defined (Evans, 2022).

FUNCTION SPACE NORMS

L^p Norm For $1 \leq p < \infty$, the $L^p(\Omega)$ norm is defined as:

$$\|u\|_{L^p(\Omega)} = \left(\int_\Omega |u(x)|^p \, dx \right)^{1/p}$$

For $p = \infty$, the $L^\infty(\Omega)$ norm is defined by the essential supremum:

$$\|u\|_{L^\infty(\Omega)} = \operatorname*{ess\,sup}_{x \in \Omega} |u(x)|$$

$W^{k,p}$ (Sobolev) Norm Let $k \in \mathbb{N}$ and $1 \leq p \leq \infty$. The Sobolev norm for the space $W^{k,p}(\Omega)$ is defined using multi-index notation for weak derivatives $D^\alpha u$, where $|\alpha| \leq k$. For $1 \leq p < \infty$, the norm is:

$$\|u\|_{W^{k,p}(\Omega)} = \left(\sum_{|\alpha| \leq k} \|D^\alpha u\|_{L^p(\Omega)}^p \right)^{1/p}$$

For $p = \infty$, the norm is:

$$\|u\|_{W^{k,\infty}(\Omega)} = \max_{|\alpha| \leq k} \|D^\alpha u\|_{L^\infty(\Omega)}$$

C^k Norm For a function $u \in C^k(\bar{\Omega})$, which is k times continuously differentiable on the closure of Ω, the C^k norm is defined as the sum of the supremum norms of all its partial derivatives up to order k:

$$\|u\|_{C^k(\bar{\Omega})} = \sum_{|\alpha| \leq k} \sup_{x \in \bar{\Omega}} |D^\alpha u(x)|$$

A.1 VISCOSITY SOLUTIONS

Denote by $\mathrm{USC}(\bar{\Omega})$ and $\mathrm{LSC}(\bar{\Omega})$, the space of upper and lower semi-continuous functions, respectively. The viscosity solution of the Eikonal equation is defined rigourously below.

Definition A.1 (Viscosity Solution) *A function $u \in \mathrm{USC}(\bar{\Omega})$ is a* viscosity subsolution *of (10) if for all $x_0 \in \bar{\Omega}$ and all $\phi \in C^\infty(\mathbb{R}^3)$ such that $u - \phi$ has a local maximum at x_0, we have:*

$$\begin{cases} \|\nabla \phi(x_0)\|_2 - f(x_0) \leq 0, & \textit{if } x_0 \in \Omega \\ \min\{\|\nabla \phi(x_0)\|_2 - f(x_0), u(x_0) - g(x_0)\} \leq 0, & \textit{if } x_0 \in \partial\Omega \end{cases}$$

Similarly, $u \in \mathrm{LSC}(\bar{\Omega})$ is a viscosity supersolution *of equation 10 if for all $x_0 \in \bar{\Omega}$ and all $\phi \in C^\infty(\mathbb{R}^3)$ such that $u - \phi$ has a local minimum at x_0, the following inequality holds:*

$$\begin{cases} \|\nabla \phi(x_0)\|_2 - f(x_0) \geq 0, & \textit{if } x_0 \in \Omega \\ \max\{\|\nabla \phi(x_0)\|_2 - f(x_0), u(x_0) - g(x_0)\} \geq 0, & \textit{if } x_0 \in \partial\Omega \end{cases}$$

Then, $u \in C(\Omega)$ is a viscosity solution *of (10) if it is both a viscosity subsolution and a supersolution.*

Next, we state formally the result for covergence of solutions of the parabolic equation 2 to the solution of equation 1 in the limit (Crandall & Lions, 1983).

Theorem 2 *For each $\varepsilon > 0$, let $u_\varepsilon \in C^2(\bar{\Omega}) \cap C(\bar{\Omega})$ denote the unique solution to 2. Then $u_\varepsilon \to u$ uniformly, as $\varepsilon \to 0^+$, where ε is the unique viscosity solution of (1).*

A.2 MATHEMATICAL PROOFS

The computational domain is often chosen as a bounding box tightly fitted to the surface, enclosing the shape. For analysis, we consider the domain to be the volume enclosed by the surface. Since the trained network will not exactly satisfy the Eikonal equation 1, consider the more general formulation of the boundary value problem (BVP):

$$\|\nabla u(x)\|_2 = f(x),\ x \in \Omega, \quad u(x) = g(x),\ x \in \partial\Omega. \tag{10}$$

where $f \in C^\infty(\bar{\Omega})$, $g \in C(\partial\Omega)$, for $\bar{\Omega} = \Omega \cup \partial\Omega$. Let $u \in C(\bar{\Omega})$ denote the viscosity solution, see the appendix for a rigorous definition. When $f \not\equiv 1$, u is not the SDF, but rather the shortest arrival time of a wavefront propagating from $x \in \bar{\Omega}$ to $\partial\Omega$. The function f represents the "slowness" (reciprocal of the speed) in the medium, while g acts as an exit-time penalty.

To obtain the required bounds, we establish a few preliminary results for viscosity solutions.

A.2.1 LEMMA 1

Lemma 1 *Let $u_1, u_2 \in C(\bar{\Omega})$ be viscosity solutions of the Eikonal equation $\|\nabla u\|_2 = f$, subject to the respective boundary conditions $u_{1|\partial\Omega} = g_1$, $u_{2|\partial\Omega} = g_2$, for $g_1, g_2 \in C(\partial\Omega)$. Then:*

$$\|u_1 - u_2\|_\infty \leq \|g_1 - g_2\|_\infty. \tag{11}$$

PROOF

Let $C = \max_{\partial\Omega} \|u - v\|$. Then by definition:

$$u(x) - C \leq v(x),\ x \in \partial\Omega. \tag{12}$$

The function $u(x) - C$ is also a solution to the equation $\|\nabla(u - C)\| = f$. The comparison principle for Hamilton-Jacobi equations (see Corollary 3.2 in Calder (2018)) then implies that:

$$u(x) - C \leq v(x)\ x \in \Omega \tag{13}$$

$$\implies u(x) - v(x) \leq \max_{\partial\Omega}(u - v),\ \forall\, x \in \Omega \tag{14}$$

This bound may also be obtained for $v - u$ by flipping u and v. It follows that,

$$\|u - v\|_\infty \leq \max_{\partial\Omega} \|u - v\| = \|g_1 - g_2\|_\infty \tag{15}$$

Lemma 1 shows that equation 10 has at most one continuous viscosity solution. Next, we provide a stability estimate that shows the sensitivity of the viscosity solution to the slowness function.

A.2.2 LEMMA 2

Lemma 2 *Let $u_1,\ u_2$ be unique viscosity solutions of $\|\nabla u\|_2 = f_1$, $\|\nabla u\|_2 = f_2$, respectively, with $u_{1|\partial\Omega} = u_{2|\partial\Omega} = 0$. Here, $f_1,\ f_2 \in C^\infty(\mathbb{R}^3)$, and assume, $\exists\, C_f > 0$ such that $0 < \frac{1}{C_f} \leq f_1, f_2 < C_f$. Then the solutions satisfy:*

$$\|u_1 - u_2\|_\infty \leq C_\Omega C_f^{-2} \|f_1 - f_2\|_\infty. \tag{16}$$

where C_Ω is a constant corresponding to the diameter of Ω.

Let $u_1,\ u_2$ be viscosity solutions of $\|\nabla u\|_2 = f_1$, $\|\nabla u\|_2 = f_2$, respectively. The comparison principle states that if $f_1 \leq f_2$ in $\bar{\Omega}$, and $u_{1|\partial\Omega} \leq u_{2|\partial\Omega}$ then $u_1 \leq u_2$ in $\bar{\Omega}$. We prove Lemmas 1 and 2 using this theory (Calder, 2018).

PROOF

Let $\hat{f}_1 = \lambda f_1$ where $\lambda = \max_\Omega \frac{f_2}{f_1}$. By construction, this ensures that $\hat{f}_1 \geq f_2$. Note that λu_1 is the viscosity solution to the Eikonal equation with slowness $\hat{f}_1$. Since $\lambda u_1, u_2$ are the viscosity solutions, they obey the maximum principle, and hence $\lambda u_1 \geq u_2$. This leads to the following inequality:

$$\begin{aligned} u_2 - u_1 \leq (\lambda - 1)u_1 &\leq \max_\Omega \frac{f_2 - f_1}{f_1} u_1 \\ &\leq \frac{1}{C_f} \|f_1 - f_2\|_\infty u_1. \end{aligned} \tag{17}$$

Since u_1 is the signed distance function, it can be bounded by the maximum time to travel between two points in the domain, and hence,

$$\|u_1\|_\infty \leq C(\Omega) C_f^{-1}. \tag{18}$$

Inequality equation 17 may also be derived for $u_1 - u_2$ by swapping u_1 and u_2. Consequently:

$$\|u_1 - u_2\|_{L^\infty(\Omega)} \leq C_\Omega C_f^{-2} \|f_1 - f_2\|_\infty \tag{19}$$

A.2.3 PROOF OF THEOREM 1

First, we state the following classical result that follows from the Gagliardo–Nirenberg interpolation inequality relating different function norms Nirenberg (1959).

Theorem 3 *Let $\Omega \subset \mathbb{R}^3$ be an open, smooth, bounded and connected domain. For $u \in L^1(\Omega) \cap W^{6,1}(\Omega)$, we have:*

$$\|u\|_\infty \le C'_\Omega \|u\|^{1/2}_{W^{6,1}(\Omega)} \|u\|^{1/2}_1. \tag{20}$$

Here C'_Ω is a constant depending only on Ω.

Note that this result also holds for compact Riemannian manifolds Nirenberg (1959).

Proof of Theorem 1 By Lemma 1, equation 1 can have at most one continuous viscosity solution. Since u_{θ^*} is smooth, it is the unique viscosity solution to equation 4. Define an auxillary function $\hat{u}_{\theta^*} \in C(\bar{\Omega})$ such that it is the unique viscosity solution of the PDE:

$$\|\nabla \hat{u}_{\theta^*}(x)\|_2 = 1,\ x \in \Omega, \quad \hat{u}_{\theta^*}(x) = g_{\theta^*}(x),\ x \in \partial\Omega. \tag{21}$$

By the regularity of $\partial\Omega$ and $g_{\theta^*} : \partial\Omega \to \mathbb{R}$, we have $\hat{u}_{\theta^*} \in C(\Omega)$. Using the triangle inequality:

$$\|u - u_{\theta^*}\|_\infty \le \|u - \hat{u}_{\theta^*}\|_\infty + \|\hat{u}_{\theta^*} - u_{\theta^*}\|_\infty. \tag{22}$$

Using Lemma 1 and Lemma 2 to bound the first and second term, respectively:

$$\|u - u_{\theta^*}\|_\infty \le \|g_{\theta^*}\|_\infty + C_\Omega C_{\theta^*}^{-2} \|1 - f_{\theta^*}\|_\infty. \tag{23}$$

where C_Ω is a constant depending only on the domain. Using the Gagliardo-Nirenberg interpolation inequality (see Theorem 4 in the appendix) for the open bounded set Ω and compact Riemannian manifold $\partial\Omega$, along with Assumption 3:

$$\|u - u_{\theta^*}\|_\infty \lesssim M_{\theta^*} \|g_{\theta^*}\|_1^{\frac{1}{2}} + M_{\theta^*} C_{\theta^*}^{-2} \|1 - f_{\theta^*}\|_1^{\frac{1}{2}}, \tag{24}$$

where the hidden constants in $\lesssim$ only depends on $\bar{\Omega}$. Observe that both $\|g_{\theta^*}\|_1$ and $\|1 - f_{\theta^*}\|_1$ can be approximated by their sample norms. The neural SDF method samples uniformly in the domain for the Eikonal loss $\|1 - f_{\theta^*}\|_1$ and hence the $L^1(\Omega)$ quadrature error is $\mathcal{O}(N^{-1/3})$, where N is the number of sample points. Assumption 2 can be used to bound the boundary loss $\|g_{\theta^*}\|_1$. This gives:

$$\begin{aligned}\|u - u_{\theta^*}\|_\infty \lesssim{}& M_{\theta^*} \left(\frac{\sum_{i=1}^N |g_{\theta^*}(x_i)|}{N} \right)^{\frac{1}{2}} + M_{\theta^*} C_{\theta^*}^{-2} \left(\frac{\sum_{i=1}^{M+N} |1 - f_{\theta^*}|}{M+N} \right)^{\frac{1}{2}} \\ &+ \mathcal{O}(N^{-\frac{\beta}{2}}) + \mathcal{O}((M+N)^{-1/6}).\end{aligned} \tag{25}$$

Since the first term can be represented using the boundary loss, and the second term by the Eikonal loss, we obtain the required result. □

The L^2 error estimate may be obtained in a similar setting as Theorem 2 by using a more general version of Theorem 1 that we state below.

Theorem 4 *Nirenberg (1959) Let $\Omega \subset \mathbb{R}^3$ be an open smooth connected domain. Let $1 \le r, m \le \infty$ and $\alpha \in [0, 1]$ such that:*

$$(1 - \alpha)\left(\frac{m}{3} - \frac{1}{r}\right) = \frac{\alpha}{p} \tag{26}$$

for $p = 1, 2$. Then for $u \in L^2(\Omega) \cap W^{m,r}(\Omega)$, we have:

$$\|u\|_\infty \le \|u\|^{1-\alpha}_{W^{m,r}(\Omega)} \|u\|^\alpha_p. \tag{27}$$

By following the proof of Theorem 2, with this inequality, we can provide a similar result.

A.3 DERIVATION OF THE GRADIENT FLOW

For $p = 1$, the first variation of the energy $\delta\mathcal{L}_{veik}(u)[v]$ in the direction of a test function v is:

$$\delta\mathcal{L}_{veik}(u)[v] = \int_\Omega \text{sign}(r(u)) \left(\frac{\nabla u \cdot \nabla v}{\|\nabla u\|_2} - \varepsilon \Delta v \right) dx \tag{28}$$

To obtain the Frèchet derivative, we apply integration by parts (with vanishing boundary terms as SDF is zero on the boundary). Let $r(u) = \|\nabla u\|_2 - 1 - \varepsilon \Delta u$ denote the viscous residual.

- Eikonal term: $\int \text{sign}(r) \frac{\nabla u \cdot \nabla v}{\|\nabla u\|} dx = - \int v \nabla \cdot \left(\text{sign}(r) \frac{\nabla u}{\|\nabla u\|} \right) dx.$
- Viscous term: $-\varepsilon \int \text{sign}(r) \Delta v \, dx = -\varepsilon \int v \Delta(\text{sign}(r)) dx.$

The Frèchet derivative is:

$$\frac{\delta\mathcal{L}_{veik}}{\delta u} = -\nabla \cdot \left(\text{sign}(r) \frac{\nabla u}{\|\nabla u\|_2} \right) - \varepsilon \Delta(\text{sign}(r)) \tag{29}$$

and the gradient flow equation may be obtained as $u_t = -\frac{\delta\mathcal{L}}{\delta u}$.

A.4 PROOF OF STABILITY FOR $p = 2$.

For $p = 2$, the first variation of the energy $\delta\mathcal{L}_{veik}(u)[v]$ is:

$$\delta\mathcal{L}_{veik}(u)[v] = \int_\Omega r(u) \left(\frac{\nabla u \cdot \nabla v}{\|\nabla u\|_2} - \varepsilon \Delta v \right) dx \tag{30}$$

Applying integration by parts as before

- Eikonal term: $\int r \frac{\nabla u \cdot \nabla v}{\|\nabla u\|} dx = - \int v \nabla \cdot \left(r \frac{\nabla u}{\|\nabla u\|} \right) dx.$
- Viscosity term: $-\varepsilon \int r \Delta v \, dx = -\varepsilon \int v \Delta r \, dx.$

The Frèchet derivative is:

$$\frac{\delta\mathcal{L}_{veik}}{\delta u} = -\nabla \cdot \left(r \frac{\nabla u}{\|\nabla u\|_2} \right) - \varepsilon \Delta r \tag{31}$$

The gradient flow equation is then $u_t = -\frac{\delta\mathcal{L}}{\delta u}$. Linearizing around $u_0 = \mathbf{a} \cdot x$ (with $\|\mathbf{a}\|_2 = 1$):

$$u_t = (\mathbf{a} \cdot \nabla)^2 u - \varepsilon^2 \Delta^2 u \tag{32}$$

In the Fourier domain:

$$\hat{v}_t = -\left((\mathbf{a} \cdot \omega)^2 + \varepsilon^2 \|\omega\|_2^4 \right) \hat{u} \tag{33}$$

The Fourier symbol $\lambda(\omega) = -[(\mathbf{a} \cdot \omega)^2 + \varepsilon^2 \|\omega\|_2^4]$ is always non-positive. Frequency damping and thus stability is again guaranteed by the $\varepsilon^2 \|\omega\|^4$ term.

A.5 IMPLEMENTATION DETAILS

All the methods are evaluated on a single Nvidia RTX A6000 GPU. For testing for all shapes, we use the Marching Cubes algorithm Lorensen & Cline (1998) with resolution 512 and the same mesh extraction process as Yang et al. (2023), Ben-Shabat et al. (2022) and other methods.

We use the sine activation function proposed in SIREN to compute the second derivatives needed for our task. For all our experiments, we find a linear decay of ε to be sufficient. Further implementation details are listed in the appendix.

Table 3: Quantitative comparison of SDF reconstruction on 2D Mandelbrot Set.

	Overall		Near Surface (< 0.05)	
Method	RMSE	MAE	RMSE	MAE
SIREN	0.055	0.036	0.0150	0.0091
DiGS	0.052	0.031	0.0110	0.0074
StEik	0.034	0.023	0.0120	0.0076
ViscoReg	**0.024**	**0.015**	**0.0091**	**0.0062**

A.5.1 2D MANDELBROT SET

All methods were evaluated with the same 4 layer 128 neuron architecture, and optimized for 10000 iterations. We compute the ground truth SDF using a dense sampling of the 2D domain and report the RMSE and MAE of the learned SDF.

All methods utilized a 4-layer SIREN architecture (sine activations) with 128 hidden units, initialized via Multi-Frequency Geometric Initialization (MFGI). The models were optimized using a learning rate of $\mathbf{5 \times 10^{-5}}$. At each step, we sampled $\mathbf{15,000}$ domain points. The loss terms were balanced with weights $[\alpha_m, \alpha_{nm}, \alpha_e] = [3000, 100, 50]$. To decouple spurious errors from far-field extrapolation, we report RMSE and MAE within a bounded domain (all points with SDF <0.5). As the object is normalized to have a bounding box half-width of 0.5, this evaluation threshold of $\delta = 0.5$ is equivalent to 100% of the object's maximal spatial extent from the center and includes all interior points. We additionally report "Near Surface" error (i.e abs(SDF)<0.05) to highlight accuracy near the zero-level set. See Table 5 for results.

A.5.2 SURFACE RECONSTRUCTION BENCHMARK

First, we center the input point clouds at the origin and normalize them so that it is inside the unit cube. The bounding box is scaled to 1.1 times the size of the shape. At each iteration, we sample 15,000 points from the original point cloud and an additional 15,000 points uniformly from the bounding box. Training is conducted for 10,000 iterations with a learning rate of 10^{-4}. The weights were taken to be $[\alpha_m, \alpha_{nm}, \alpha_e] = [3000, 100, 50]$. Baseline decay for all shapes is initial $\varepsilon = 0.5$, decayed linearly at 20/40/60/80% iterations to 0.4/0.04/0.005/0. We used 5 hidden layers, and 128 nodes. MFGI with sphere initial parameters was taken to be $(1.6, 0.1)$.

To train the network we used 5 hidden layers and 128 neurons. For ε, we used an annealing strategy, setting $\varepsilon = 0.5$ initially and decaying to zero through piece-wise linear schedule. This decay schedule does not add extra hyperparameters because a similar annealing strategy was used for the divergence terms in DiGS and StEik. We used the MFGI initialization from DiGS.

Additional quantitative results for each individual shape are presented in Table 12.

Adaptive Decay Strategy: We test a method to adaptively control the ε parameter based on the absolute error of the eikonal constraint. The update for the viscosity weight ε_t at iteration t is defined as:

$$\varepsilon_t = \begin{cases} \max\left(\varepsilon_{\min}, \varepsilon_{\text{target\_decay}}(t) + S \cdot \text{EMA}_{t-1}\right) & \text{if } t < 0.5 \cdot N \\ 0 & \text{if } t \geq 0.5 \cdot N \end{cases}$$

Here $\varepsilon_{\text{target\_decay}}$ is the baseline target decay. The baseline provides a monotonically decaying target for ε_t. The progress p is normalized over the first 50% of training: $p = t/(0.5 \cdot N)$, with $\varepsilon_{\text{target\_decay}}(t) = \varepsilon_{\text{initial}} \cdot (\gamma_{\text{factor}})^p$. Then we use Exponential Moving Average (EMA) to track the recent residual of the pure Eikonal constraint, L_{Eik}, to assess the need for regularization.

$$\text{EMA}_t = \beta \cdot \text{EMA}_{t-1} + (1 - \beta) \cdot L_{\text{Eik},t}$$

For the SRB dataset we use $\varepsilon_{\text{initial}} = 0.4$, $\gamma_{\text{factor}} = 0.01$, $\beta = 0.999$, $S = 0.5$ (Sensitivity factor) and $\varepsilon_{\min} = 0.0$.

Table 4: Adaptive Decay for SRB. d_C : Chamfer and d_H: Hausdorff distance.

Shape	**Piecewise Linear**		**Adaptive Decay**	
	d_C	d_H	d_C	d_H
`anchor`	0.23	4.35	0.23	4.14
`dc`	0.16	1.33	0.15	1.53
`daratech`	0.18	2.96	0.19	1.87
`gargoyle`	0.18	3.81	0.18	4.31
`lord_quas`	0.13	1.37	0.13	1.13
Mean	**0.18**	**2.76**	**0.18**	**2.60**

Loss	d_C	d_H
L^1 ViscoReg	0.18	2.76
L^2 ViscoReg	0.19	4.49

Table 5: L^1 vs L^2 ViscoReg loss terms

A.5.3 L^1 VS L^2 VISCOREG

We tested the model on SRB using $p = 2$ for the ViscoReg parameter in Table 5. Empirically, we see better performance for $p = 1$. The improved performance of $p = 1$ is seen across a range of method such as DiGS and StEik as well. It is perhaps unsurprising as L^2 tends to suppress the large gradients required for sharp edges, which can cause over-smoothing. Whereas the L^1 norm is more tolerant to these geometric features.

A.5.4 COMPUTATIONAL COST

The bulk of the computational cost comes from calculating the Laplacian. Hence, the computational cost is just slightly lower than that of DiGS and higher than that of StEik (who computes the directional derivative with more efficient calculations). Timings are presented in Table 6 for the SRB dataset with experiments conducted on A100 GPU.

Table 6: Runtime comparison in ms. All models utilize a 4×256 architecture with 0.26M parameters.

Method	**DiGS**	**StEik**	**ViscoReg**	**Siren**
Runtime (ms)	36.05	27.68	35.50	15.63

A.5.5 ABLATION

For the ablation studies, the decay schedules are as follows. BL$\times x$= baseline decay of 0.5/0.4/0.04/0.005/0 at 0/20/40/60/80%4 iterations scaled by x. Fast decay corresponds to a quick decay to 0, of 0.5/0.0 at 0/20% iterations. Slow decay corresponds to extended decay at 90 percent of iterations with a schedule 0.5/0.4/0.04/0.005/0 at 0/20/40/60/90%. We also test piecewise constant and piecewise quintic decay as opposed to piecewise linear. Ablation studies per shape are provided in Tab.7-11.

A.5.6 FASTER CONVERGENCE

We demonstrate ViscoReg's faster convergence to better minima (in terms of the Eikonal constraint) than SIREN Sitzmann et al. (2019b) with unstable Eikonal loss (see Fig 4).

A.5.7 SCENE RECONSTRUCTION

For this experiment, we used an architecture of 8 hidden layers, and 512 channels. At each iteration, we sample 15,000 points from the original point cloud and another 15,000 points uniformly at random within the bounding box. Training is performed for 100,000 iterations with a learning rate

Table 7: Ablation on ε decay for `anchor`.

Method	$d_C \downarrow$	$d_H \downarrow$
BL	**0.23**	4.35
BL ×2	**0.25**	5.36
BL ×0.5	0.26	4.70
Piecewise Const.	0.29	7.97
Quintic	0.26	6.35
Fast decay (0 @ 20%)	0.31	7.33
Slow decay(0 @ 90%)	0.25	5.71
$\varepsilon = 0$ (SIREN [47])	0.72	10.98
SoTA best	0.26	**4.26**

Table 8: Ablation on ε decay for `dc`.

Method	$d_C \downarrow$	$d_H \downarrow$
BL	0.16	1.33
BL ×2	**0.15**	1.44
BL ×0.5	0.16	1.39
Piecewise Const.	0.16	1.49
Quintic	0.17	1.32
Fast decay (0 @ 20%)	0.16	1.35
Slow decay(0 @ 90%)	0.15	**1.23**
$\varepsilon = 0$ (SIREN [47])	0.34	6.27
SoTA best	0.15	1.70

Table 9: Ablation on ε decay for `daratech`.

Method	$d_C \downarrow$	$d_H \downarrow$
BL	**0.18**	1.33
BL ×2	**0.18**	1.44
BL ×0.5	0.20	1.39
Piecewise Const.	0.21	1.49
Quintic	0.19	1.32
Fast decay (0 @ 20%)	0.19	1.35
Slow decay(0 @ 90%)	0.20	**1.23**
$\varepsilon = 0$ (SIREN [47])	0.21	6.27
SoTA best	**0.18**	1.72

Table 10: Ablation on ε decay for `gargoyle`.

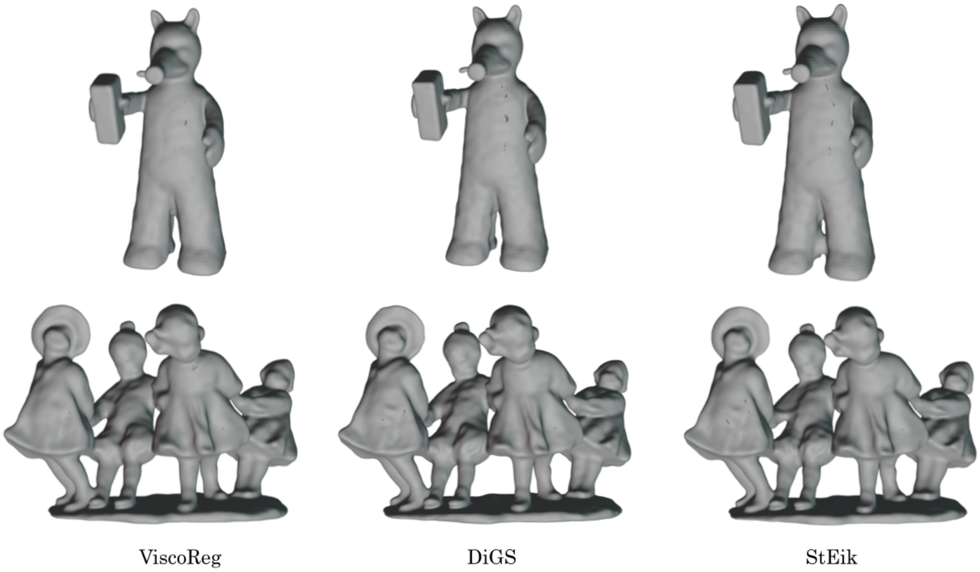
Method	$d_C \downarrow$	$d_H \downarrow$
BL	0.18	**3.81**
BL ×2	**0.17**	3.97
BL ×0.5	0.21	9.18
Piecewise Const.	0.18	4.09
Quintic	0.19	6.06
Fast decay (0 @ 20%)	0.18	3.95
Slow decay(0 @ 90%)	0.19	4.48
$\varepsilon = 0$ (SIREN [47])	0.46	7.76
SoTA best	**0.17**	4.10

Table 11: Ablation on ε decay for `lord_quas`.

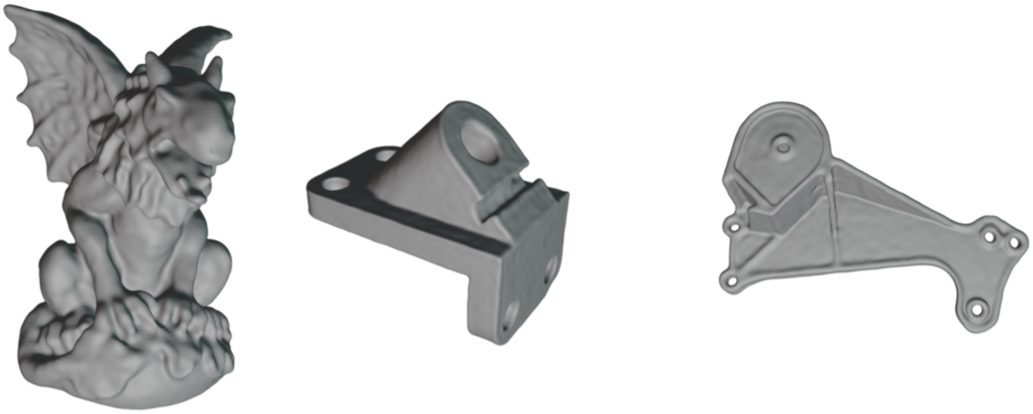
Method	$d_C \downarrow$	$d_H \downarrow$
BL	0.13	1.37
BL ×2	0.13	2.18
BL ×0.5	0.14	6.45
Piecewise Const.	0.14	3.65
Quintic	0.14	2.30
Fast decay (0 @ 20%)	0.13	2.04
Slow decay(0 @ 90%)	0.12	**1.41**
$\varepsilon = 0$ (SIREN [47])	0.35	8.96
SoTA best	**0.11**	**0.70**

Shape	Method	d_C	d_H
Overall	IGR wo n	1.38	16.33
	SIREN wo n	0.42	7.67
	SAL	0.36	7.47
	IGR+FF	0.96	11.06
	PHASE+FF	0.22	4.96
	DiGS	0.19	3.52
	StEik	**0.18**	2.80
	ViscoReg	**0.18**	2.76
	ViscoReg (quad)	**0.18**	**2.69**
Anchor	IGR wo n	0.45	7.45
	SIREN wo n	0.72	10.98
	SAL	0.42	7.21
	IGR+FF	0.72	9.48
	PHASE+FF	0.29	7.43
	DiGS	0.29	7.19
	StEik	0.26	4.26
	ViscoReg	**0.23**	**4.35**
	ViscoReg (quad)	0.26	4.90
Daratech	IGR wo n	4.9	42.15
	SIREN wo n	0.21	4.37
	SAL	0.62	13.21
	IGR+FF	2.48	19.6
	PHASE+FF	0.35	7.24
	DiGS	0.20	3.72
	StEik	0.18	1.72
	ViscoReg	0.19	2.97
	ViscoReg (quad)	**0.17**	**1.43**
DC	IGR wo n	0.63	10.35
	SIREN wo n	0.34	6.27
	SAL	0.18	3.06
	IGR+FF	0.86	10.32
	PHASE+FF	0.19	4.65
	DiGS	**0.15**	1.70
	StEik	0.16	1.73
	ViscoReg	0.16	1.33
	ViscoReg (quad)	0.16	**1.29**
Gargoyle	IGR wo n	0.77	17.46
	SIREN wo n	0.46	7.76
	SAL	0.45	9.74
	IGR+FF	0.26	5.24
	PHASE+FF	**0.17**	4.79
	DiGS	**0.17**	4.10
	StEik	0.18	4.49
	ViscoReg	0.18	**3.80**
	ViscoReg (quad)	0.18	4.15
Lord Quas	IGR wo n	0.16	4.22
	SIREN wo n	0.35	8.96
	SAL	0.13	4.14
	IGR+FF	0.49	10.71
	PHASE+FF	**0.11**	**0.71**
	DiGS	0.12	0.91
	StEik	0.13	1.81
	ViscoReg	0.14	1.37
	ViscoReg (quad)	0.13	1.69

Table 12: Additional quantitative results on the Surface Reconstruction Benchmark using point data without normals.



(a) Comparison on SRB shapes `dc` and `lord_quas`



(b) Reconstructed shapes `gargoyle`, `anchor`, and `daratech` using ViscoReg.

Figure 3: Qualitative results from SRB.

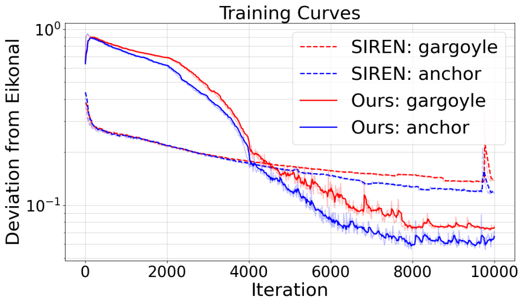
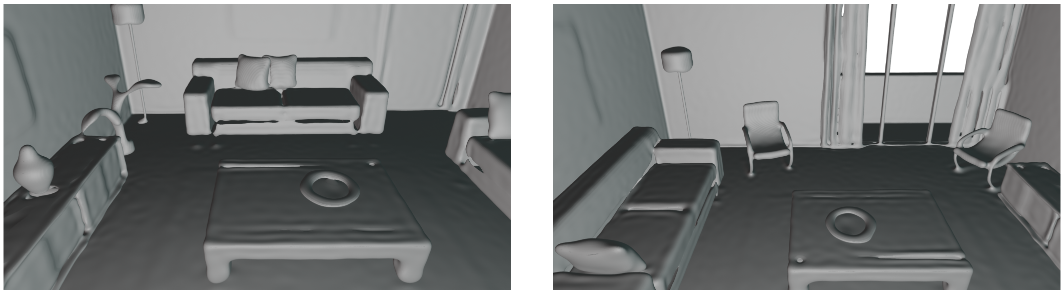


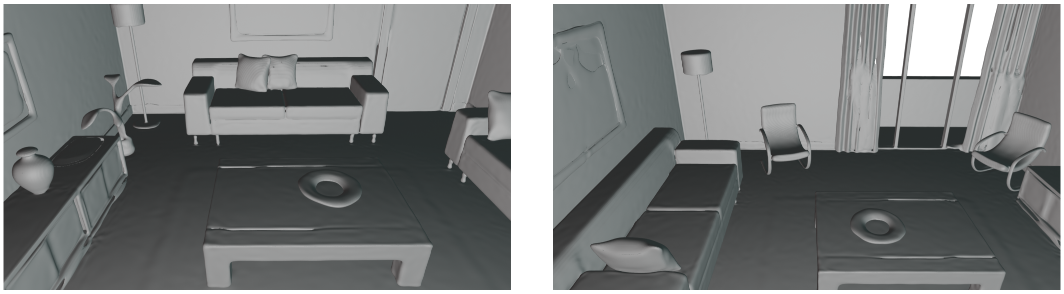
Figure 4: Deviation from Eikonal.

of 8×10^{-6}. The weights used were $[\alpha_m, \alpha_{nm}, \alpha_e] = [5000, 100, 50]$. The viscosity coefficient

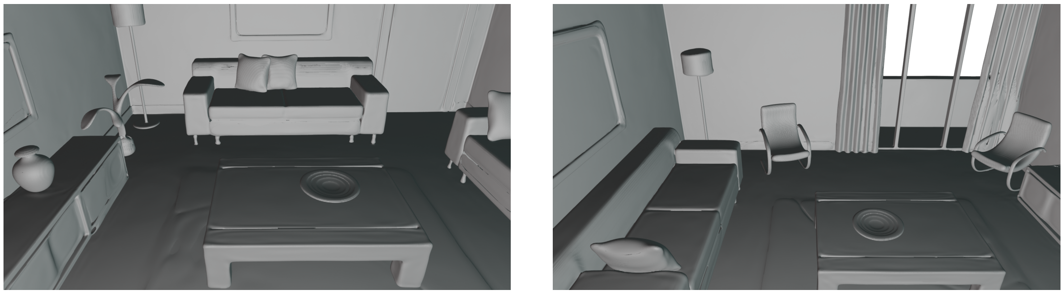
ε decayed piecewise linearly starting at 0.5, decaying to 0.01 at 50 percent iterations followed by steeply decaying to 0 at 60 percent.



(a) DiGS (d) DiGS



(b) StEik (e) StEik



(c) ViscoReg (ours) (f) ViscoReg (ours)

Figure 5: Results from the scene reconstruction benchmark from Sitzmann et al. (2019b). The left column** compares results on one view of the scene: The DiGS mesh (a) is missing fine details like the sofa legs and picture frame details. StEik (b) performs better but struggles with fine details such as the curtains. ViscoReg (c) reconstructs these fine details with high fidelity. The right column provides additional views of the scene.