[Re] p-Poisson surface reconstruction in curl-free flow from point clouds

Anonymous authors
Paper under double-blind review

Abstract

This study presents a reproducibility analysis of the p-Poisson surface reconstruction method presented by Park et al. (NeurIPS 2023). The method utilizes the p-Poisson equation and a curl-free constraint for improved surface reconstruction from point clouds, claiming significant advancements over existing implicit neural representation techniques. Our objective is to replicate the results reported in the original paper, focusing on efficient implementation and evaluation using the Surface Reconstruction Benchmark (SRB) dataset. We re-implemented the neural network architecture and training procedures, emphasizing code efficiency. While our replication generally outperforms alternative methods, it falls short of reproducing the reported results. Notably, training with the authors' provided code yields results comparable to our results, but still deviating from the presented findings. Despite these challenges, our implementation demonstrates a significant improvement in training performance, achieving a five-fold acceleration in training times compared to the original code. Our code is published here [1].

1 Reproducibility Summary

1.1 Scope of Reproducibility

This reproducibility study focuses on the approach presented in the paper "p-Poisson surface reconstruction in curl-free flow from point clouds" Park et al. (2023) presented at NeurIPS 2023. The original work introduces a methodology that employs the p-Poisson equation and a curl-free constraint for enhanced surface reconstruction from points clouds, purportedly achieving significant improvements over comparable implicit neural representations that rely on supplementary information such as surface normals. This study aims to reproduce the results of the original paper, focusing on creating an efficient implementation of the PDE-based framework and the learning process, as well as the evaluation of the proposed approach's performance in terms of surface reconstruction on the Surface Reconstruction Benchmark (SRB) dataset (Berger et al., 2013).

1.2 Methodology

To reproduce the results, we re-implemented the neural network architecture and training procedures as detailed by Park et al. (2023). This includes the implementation of the auxiliary variable strategy, the enforcement of the p-Poisson equation, and the integration of a curl-free constraint to ensure the representation of a conservative vector field. A central aspect of our reproducibility study is to provide a codebase that compiles the training process and neural network using XLA through JAX Bradbury et al. (2018) for efficient numerical computations and automatic differentiation.

[1] `https://anonymous.4open.science/r/pinc-B7CD/`

1.3 Results

The quantitative metrics for all benchmarks except Daratech exhibit lower distances than other similar methods but fall short of the distance metrics presented in Park et al. (2023). Upon training the model using the original code published by the authors, we achieved results that are very similar to our replication for all benchmarks. Unfortunately, due to the lack of response from the original authors, we are unable to validate the results reported in the original paper as no trained models, reconstructed meshes, or evaluation script was published. Notably, when evaluating training time and computational resources, our implementation demonstrates significant improvements, with 5x faster training times.

1.4 What was easy

- The article provided detailed explanations of the p-Poisson equation and the incorporation of the curl-free constraint.
- The formulation of the proposed loss function, the integration of the minimal area criterion, and the explicit handling of occlusions, were relatively well articulated. This made the implementation of the loss function straightforward.
- The experimental setup section offered sufficient detail about the neural network architecture, the variable-splitting strategy, and the use of auxiliary variables, which facilitated the implementation of the model and forward pass.
- The use of Chamfer and Hausdorff distances as evaluation metrics, made it possible to benchmark the reproduced results against the original findings and compare them to the metrics reported by other comparable implicit neural representation methods for surface reconstruction.
- The reliance on the widely used Surface Reconstruction Benchmark (SRB) (Berger et al., 2013), enabled us to directly compare the performance of our reproduced model with the reported results, without the need for additional data preprocessing or acquisition.

1.5 What was difficult

Several challenges were encountered during the reproduction process:

- Multiple undocumented numerical differences were not mentioned in the original paper. For example, the geometric initialization was slightly different than in other implementations, the number of sampled global points was not mentioned in the paper, there is a mistake in the 50th nearest neighbor calculation, the division by $\sqrt{2}$ after the skip connection in the multi-layer perceptron was not mentioned. The authors did also not provide the evaluation code, which made it difficult to verify the results.
- When training a model using the original code provided by the authors, we achieved results comparable to our replication. However, these results are not on par with the results presented in Park et al. (2023).
- The evaluation of the model is done using sampling of a mesh constructed using marching cubes, where the random seed for the sampling is not mentioned and the evaluation code is not provided, which means that the evaluation metrics vary each time they are calculated.
- The original authors' code had very long training times, which meant that we had to use a lot of computational resources to run their published code for comparison after our replication results did not meet expectations.

1.6 Communication with original authors

A mail to the original authors inquiring about the availability of evaluation code, trained models, and reconstruction results has been sent but has not been answered at the time of writing.

2 Introduction

Surface reconstruction from unorganized point clouds, crucial in computer vision and graphics, has seen extensive research over two decades. While traditional representations like point clouds or meshes lack watertight surfaces and flexibility, implicit function-based approaches such as signed distance functions (SDFs) or occupancy functions offer watertight results and topology flexibility. The advent of deep learning introduced implicit neural representations (INRs), utilizing neural networks to parameterize implicit functions for efficient training and expressive reconstruction. Early INRs treated reconstruction as supervised regression, facing challenges with ground-truth distance values. Some methods employ partial differential equations (PDEs) like the eikonal equation to alleviate the need for 3D supervision, but they struggle with non-unique solutions and reliance on accurate normal vectors, which may be unavailable in raw point cloud data. Additionally, these methods are sensitive to noise and outliers, limiting their effectiveness in reconstructing fine details or realistic surfaces without normal vectors. In this study, therefore we aim to reproduce the results of the method termed p-**P**oisson equation based **I**mplicit **N**eural representation with **C**url-free constraint (PINC) introduced by Park et al. (2023) which purportedly achieves significant improvements over comparable implicit neural representations that rely on supplementary information such as surface normals.

3 Scope of reproducibility

This study aims to reproduce the results of the original paper by Park et al. (2023) and evaluate the performance of the proposed method for surface reconstruction from unorganized point clouds, focusing on the efficient implementation of the PDE-based variable-splitting strategy and training process as well as the evaluation of the proposed approach's surface reconstruction performance on the Surface Reconstruction Benchmark (SRB) dataset (Berger et al., 2013). The original paper claims that the proposed method outperforms existing implicit neural representation approaches in terms of accuracy on the metrics of Chamfer and Hausdorff distances, robustness to noise, and handling of incomplete observations. Our study seeks to validate these claims and assess the computational efficiency of the proposed method. Our reproducibility study also offers a new implementation of the proposed method using JAX which significantly improves training speed.

4 Methodology

The original paper by Park et al. (2023) introduces a novel approach to surface reconstruction from point clouds using INRs. The proposed method leverages the p-Poisson equation and a curl-free constraint to enhance the accuracy and robustness of the reconstructed surfaces. Unlike comparable methods that require additional information such as surface normals, the proposed approach learns the SDF implicitly, allowing for more flexible and accurate surface reconstructions.

4.1 Problem Formulation

Given an unorganized point cloud $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ sampled from a closed surface Γ, the goal is to learn a signed distance function $u : \mathbb{R}^3 \to \mathbb{R}$, where the zero level set of u accurately represents the surface Γ, such that $\Gamma = \left\{\boldsymbol{x} \in \mathbb{R}^3 \mid u(\boldsymbol{x}) = 0\right\}$. In the following equations, Ω represents a subset of the entire definition space of $u(\boldsymbol{x})$ for which we sample points to constrain the optimization.

p-Poisson Equation The p-Poisson equation serves as the foundation for PINC, allowing us to model the signed distance function with high precision. The equation is defined as:

$$\min_u \int_\Gamma |u| d\boldsymbol{x} + \lambda_1 \int_\Omega \left|\nabla_{\boldsymbol{x}} \cdot \left(\|\nabla_{\boldsymbol{x}} u\|^{p-2} \nabla_{\boldsymbol{x}} u\right) + 1\right| d\boldsymbol{x}, \tag{1}$$

which using the augmented neural network structure with auxiliary variable G proposed by Park et al. (2023) can be rewritten as a loss function of the form

$$\mathcal{L}_{p\text{-Poisson}} = \int_\Gamma |u| d\boldsymbol{x} + \lambda_1 \int_\Omega \|\nabla_{\boldsymbol{x}} u - G\|^2 d\boldsymbol{x} \tag{2}$$

where $\lambda_1 > 0$ is a weighting hyperparameter and G is defined from the neural network output vector Ψ and the fixed function $F(\boldsymbol{x}) = \frac{\boldsymbol{x}}{3}$, chosen such that $\nabla_{\boldsymbol{x}} \cdot F = 1$

$$G = \frac{\nabla_{\boldsymbol{x}} \times \Psi - F}{\|\nabla_{\boldsymbol{x}} \times \Psi - F\|^{\frac{p-2}{p-1}}}. \tag{3}$$

PINC integrates this equation as a hard constraint within the neural network model, ensuring the learned signed distance function adheres to the physical properties of the surface.

Curl-Free Constraint In the original paper, the authors argue that to enhance the accuracy of surface reconstructions from point clouds, a curl-free constraint should be applied to the auxiliary variable G, which represents the gradient of the signed distance function (SDF), to ensures that G forms a conservative vector field. A conservative vector field condition, expressed as $G = \nabla_{\boldsymbol{x}} u$, indicates that G is curl-free ($\nabla_{\boldsymbol{x}} \times G = 0$) if it can be written as the gradient of some scalar potential function u.

Implementing a direct penalty for the curl of G to enforce this constraint ($\int_\Omega \|\nabla_{\boldsymbol{x}} \times G\|^2 \, dx$) is claimed to introduce computational challenges and a complex loss landscape due to high-order derivatives required by automatic differentiation (Park et al., 2023). To mitigate these issues, an additional auxiliary variable $\tilde{G}$ is introduced to satisfy both $G = \tilde{G}$ and the curl-free condition $\nabla_{\boldsymbol{x}} \times \tilde{G} = 0$, through the loss function

$$\mathcal{L}_{\text{PINC}} = \mathcal{L}_{p\text{-Poisson}} + \lambda_2 \int_\Omega \left\|G - \tilde{G}\right\|^2 dx + \lambda_3 \int_\Omega \left\|\nabla_{\boldsymbol{x}} \times \tilde{G}\right\|^2 dx. \tag{4}$$

where $\lambda_2, \lambda_3 > 0$ are weighting hyperparameters.

The optimality conditions suggest that $\tilde{G}$ should have a unit norm, as dictated by the Eikonal equation. To simplify adherence to this non-convex equality constraint, it is relaxed to a convex condition $\|\tilde{G}\| \leq 1$, using a projection operator $\mathcal{P}$ that maps the auxiliary output $\tilde{\Psi}$ to $\tilde{G}$ within the three-dimensional unit sphere:

$$\tilde{G} = \frac{\tilde{\Psi}}{\max\left\{1, \|\tilde{\Psi}\|\right\}}, \tag{5}$$

How to neural network structure relates to the auxiliary variable is summarized in figure 1.

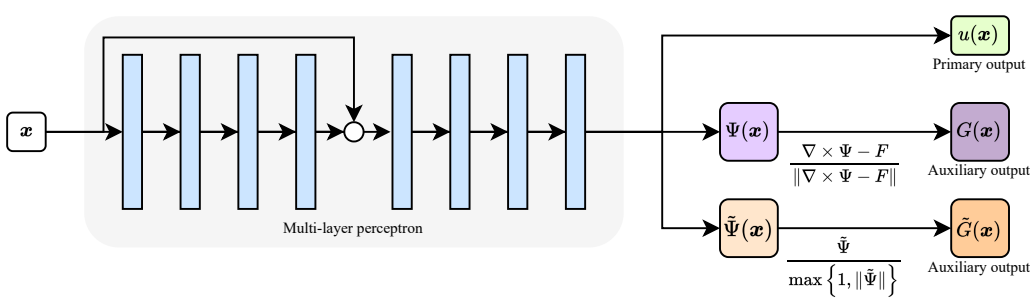

Figure 1: The visualization of the augmented network structure with two auxiliary variables.

Variable-Splitting Strategy PINC adopts a variable-splitting strategy to simplify the optimization process; splitting the network into multiple outputs, as illustrated in Fig. 1, for the signed distance function and its gradient. The authors of the paper argue that this approach enables more effective training and better adherence to the p-Poisson and curl-free constraints.

4.2 Loss Function

Real point clouds from range scanners often have incomplete data due to occlusions and concavities, resulting in holes. Estimating accurate closed surfaces becomes challenging, requiring a strategy to interpolate across gaps and reconstruct the surface cohesively.

PINC's approach is to minimize the surface area of the zero-level set, which is encapsulated in the final augmented loss function:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{PINC}} + \lambda_4 \int_{\Omega} \delta_\epsilon(u) \left\| \nabla_{\boldsymbol{x}} u \right\| dx, \tag{6}$$

where $\lambda_4 > 0$ is the weighting hyperparameter of the surface area minimization component in the total loss function, and $\delta_\epsilon(x) = 1 - \tanh^2\left(\frac{x}{\epsilon}\right)$ represents a smoothed Dirac delta function with a smoothing parameter $\epsilon > 0$. This addition aims to guide the surface reconstruction process by encouraging the minimization of the zero-level set area of u, thereby promoting a more coherent filling of the missing parts of the scanned point cloud.

To implement the loss function in code, we approximate all integrals using Monte Carlo integration. This involves simply replacing the integrals with sums over the data points of the specific type.

4.3 Distance Metrics and Evaluation

We quantify the separation between two sets of points, denoted as $\mathcal{X}$ and $\mathcal{Y}$, through the application of conventional one-sided and double-sided ℓ_2 Chamfer distances, denoted as $d_{\vec{C}}$, d_C, and Hausdorff distances, denoted as $d_{\vec{H}}$, d_H. The definitions for each are as follows:

$$d_{\vec{C}}(\mathcal{X}, \mathcal{Y}) = \frac{1}{|\mathcal{X}|} \sum_{\boldsymbol{x} \in \mathcal{X}} \min_{\mathbf{y} \in \mathcal{Y}} \|\boldsymbol{x} - \mathbf{y}\|_2, \qquad d_C(\mathcal{X}, \mathcal{Y}) = \frac{1}{2}\left(d_{\vec{C}}(\mathcal{X}, \mathcal{Y}) + d_{\vec{C}}(\mathcal{Y}, \mathcal{X})\right)$$

$$d_{\vec{H}}(\mathcal{X}, \mathcal{Y}) = \max_{\boldsymbol{x} \in \mathcal{X}} \min_{\mathbf{y} \in \mathcal{Y}} \|\boldsymbol{x} - \mathbf{y}\|_2 \qquad d_H(\mathcal{X}, \mathcal{Y}) = \max\left\{d_{\vec{H}}(\mathcal{X}, \mathcal{Y}) + d_{\vec{H}}(\mathcal{Y}, \mathcal{X})\right\}.$$

The estimation of the distance from the INR to the target point clouds is done identically as in Park et al. (2023) by first creating a mesh by extracting the zero level set of u using the marching cubes algorithm Lorensen & Cline (1987) on a $512 \times 512 \times 512$ uniform grid, then by sampling 10^7 points uniformly from the surface and finally measure the presented distances from the sampled points and the target points cloud.

Furthermore, to measure the accuracy of the trained gradient field, we evaluate Normal Consistency (NC) between the learned G and the surface normal as follows: from given an oriented point cloud $\mathcal{X}, \mathcal{N} = \{\boldsymbol{x}_i, \mathbf{n}_i\}_{i=1}^N$ comprising of sampled points $\boldsymbol{x}_i$ and the corresponding outward normal vectors $\mathbf{n}_i$, NC is defined by

$$NC(G, \mathcal{X}, \mathcal{N}) = \frac{1}{N} \sum_{i=1}^{N} \left| G\left(\boldsymbol{x}_i\right)^{\mathrm{T}} \mathbf{n}_i \right|,$$

the average of the absolute dot product of the trained G and the surface normals.

4.4 Datasets

The SRB dataset (Berger et al., 2013) is used to benchmark our implementation against the original results. It consists of 5 different benchmark figures which each have a scan point cloud and ground truth point could. The scan is used for training the reconstruction models while the ground truth is used to evaluate the trained model.

An evaluation of the Thingy10K Zhou & Jacobson (2016) dataset was also initially planned, but as the reference of which of the 10k object was not specified in detail by the authors, we were unable to evaluate our implementation on this dataset.

All scanned data points are utilized as boundary points in the loss function. To enhance consistency, neighboring points around these boundary points are sampled by adding a normally distributed variable with a variance equal to the distance to their 50th closest neighbor within the scanned dataset. One local

Table 1: Results table for the SRB dataset using different surface reconstruction methods. Our implementation is labeled as "[Re] PINC", their code running on our hardware is labeled as "PINC (rerun)", and their presented results are labeled as "PINC". The Ground Truth data is referred to as "GT", while the training data is referred to as "Scan".

	Anchor				Daratech				Dc				Gargoyle				Lord quas			
	GT		Scan		GT		Scan		GT		Scan		GT		Scan		GT		Scan	
	d_C	d_H	$d_{\overrightarrow{C}}$	$d_{\overrightarrow{H}}$	d_C	d_H	$d_{\overrightarrow{C}}$	$d_{\overrightarrow{H}}$	d_C	d_H	$d_{\overrightarrow{C}}$	$d_{\overrightarrow{H}}$	d_C	d_H	$d_{\overrightarrow{C}}$	$d_{\overrightarrow{H}}$	d_C	d_H	$d_{\overrightarrow{C}}$	$d_{\overrightarrow{H}}$
IGR	0.45	7.45	0.17	4.55	4.90	42.15	0.70	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	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	0.42	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	414.00	0.07	4.04
PHASE	**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
DiGS	**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
PINC	**0.29**	7.54	**0.09**	1.20	0.37	7.24	0.11	1.88	**0.14**	2.56	**0.04**	2.73	**0.16**	4.78	**0.05**	**0.80**	**0.10**	0.92	**0.04**	0.67
PINC (rerun) $\epsilon = 1$	0.32	7.63	0.10	1.26	5.97	55.25	0.64	4.38	0.15	2.62	0.05	2.80	**0.16**	4.77	0.06	0.81	0.11	0.79	**0.04**	0.74
PINC (rerun) $\epsilon = 0.1$	0.37	9.10	0.10	3.94	7.48	62.82	0.66	7.39	0.16	2.26	0.06	2.75	0.19	5.95	0.07	3.86	0.14	3.41	0.05	1.50
[Re] PINC $\epsilon = 1$	0.31	**6.01**	0.17	1.41	4.73	53.51	0.48	3.42	0.17	2.18	0.10	2.73	0.21	4.67	0.14	0.91	0.18	1.81	0.12	0.86
[Re] PINC $\epsilon = 0.1$	0.35	7.67	0.16	1.39	7.52	69.12	0.24	3.10	0.17	2.37	0.11	**2.72**	0.21	5.25	0.13	1.36	0.17	1.17	0.12	1.01

point is sampled per scanned point in the batch of scan points. In the published code there is probably a mistake in this calculation, that we outline in our README[2].

To ensure that the model has been trained on points representing the entire data distribution, 2048 points are uniformly sampled from the cubic space $[-1.1, 1.1]^3$.

4.5 Hyperparameters

Numerous hyperparameters can be adjusted, all of which were configured to match the experimental setup of the original paper. However, one notable exception is the epsilon parameter in the loss function. In the source code, it was suggested that using a value of $\epsilon = 0.1$ might yield better results than the original value of $\epsilon = 1$, leading us to try both values. The appendix figure 2 presents how the function $\delta_\epsilon(x) = 1 - \tanh^2\left(\frac{x}{\epsilon}\right)$ depends on ϵ. The loss weight hyperparameters of each of the loss terms were set to $\lambda_1 = 0.1, \lambda_2 = 10^{-4}, \lambda_3 = 5 \cdot 10^{-4}$ and $\lambda_4 = 0.1$, $p = \infty$ and $F = \frac{x}{3}$ as in the original paper. The Adam (Kingma & Ba, 2014) optimizer was used with a learning rate of 10^{-3} and a learning rate schedule that reduced the learning rate by a factor of 0.99 every 2000 step. The model was trained for 100 000 steps with a batch size of 16384. The network architecture consisted of 7 layers with 512 hidden dimensions and skip connections from the first layer to the fourth layer. The softplus activation function was used with a beta parameter of 100.0 which makes it very close to a ReLU activation function.

4.6 Experimental setup and code assets

The model has been implemented in JAX Bradbury et al. (2018) by first only following the directions presented in the original paper and then consulting the published code [3]. Similarly, as Park et al. (2023), all experiments were trained on a single NVIDIA® RTX 3090 GPU with 24GB of memory.

We also verified the correctness of our implementation by comparing the results of a forward pass using both our code and the original authors' code, which yielded identical numerical results and which is included in our published code [4].

5 Results

In Table 1, we present the published results of various comparable surface reconstruction methods applied to the SRB dataset Gropp et al. (2020); Sitzmann et al. (2020); Atzmon & Lipman (2020); Lipman (2021); Ben-Shabat et al. (2022); Park et al. (2023) compared to our obtained results. Notably, our replication of

[2] `https://anonymous.4open.science/r/pinc-B7CD/README.md`
[3] `https://github.com/yebbi/pinc`
[4] `https://anonymous.4open.science/r/pinc-B7CD/scripts/forward_pass_comparison.py`

the model exhibits superior performance compared to most other models for all benchmarks except Daratech which fails to reconstruct the surface. However, it falls short of achieving the performance levels reported in the original paper. Interestingly, upon training the model using the original code provided by the authors, we obtained results comparable to those of our replication. A significant observation is the consistent inadequate performance of both models in the case of Daratech, where convergence to a stable solution did not happen. This can also be visually confirmed in 3 and the observation holds for both our implementation and when utilizing the original authors' code. We also see that when training with a smaller ε we get a less smooth mesh which means $d_{\vec{C}}$ slightly improves while d_H becomes worse.

Table 2: Comparison of total training times for 100 000 steps.

	Total Training Time (hours)
Our Implementation	2.2
Their Implementation	13.5

We can also examine the training times of the original code and our replication in Table 2. The results indicate that our implementation requires approximately 5 less training time compared to the authors' code.

Presented in the appendix figure 3 is a visualization of all the final reconstructed surfaces, highlighting the failure on the Daratech example by all the models with the hyperparameters presented by Park et al. (2023)

6 Conclusion

In our research, our primary objective was to replicate and verify the findings of the implicit neural representation surface reconstruction method proposed by Park et al. (2023). The original study introduces a novel technique that integrates the p-Poisson equation into the model's loss function, enabling surface reconstruction from point clouds without relying on additional information such as surface normals. This purportedly leads to substantial improvements over comparable implicit neural representations. Our study aimed to corroborate these assertions, revealing that the overarching methodology of PINC generally outperforms most alternative methods. However, our attempt to replicate the model's performance did not align with the reported results in the original paper. Notably, even when utilizing the authors' provided code for training, we observed discrepancies in performance compared to the reported outcomes.

Additionally, our investigation revealed that our implementation of the method surpassed the training performance of the original code provided by the authors, resulting in a remarkable five-fold acceleration in training times.

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, 2020.

Yizhak Ben-Shabat, Chamin Hewa Koneputugodage, and Stephen Gould. DiGS: Divergence guided shape implicit neural representation for unoriented point clouds. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19301–19310, 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.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`.

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. ICML'20, pp. 11. JMLR.org, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Yaron Lipman. Phase transitions, distance functions, and implicit neural representations. In *International Conference on Machine Learning*, 2021.

William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH computer graphics*, 21(4):163–169, 1987.

Yesom Park, Taekyung Lee, Jooyoung Hahn, and Myungjoo Kang. p-poisson surface reconstruction in curl-free flow from point clouds. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, number 626 in NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

Qingnan Zhou and Alec Jacobson. Thingi10k: A dataset of 10,000 3D-printing models. *arXiv preprint arXiv:1605.04797*, 2016.

A Appendix

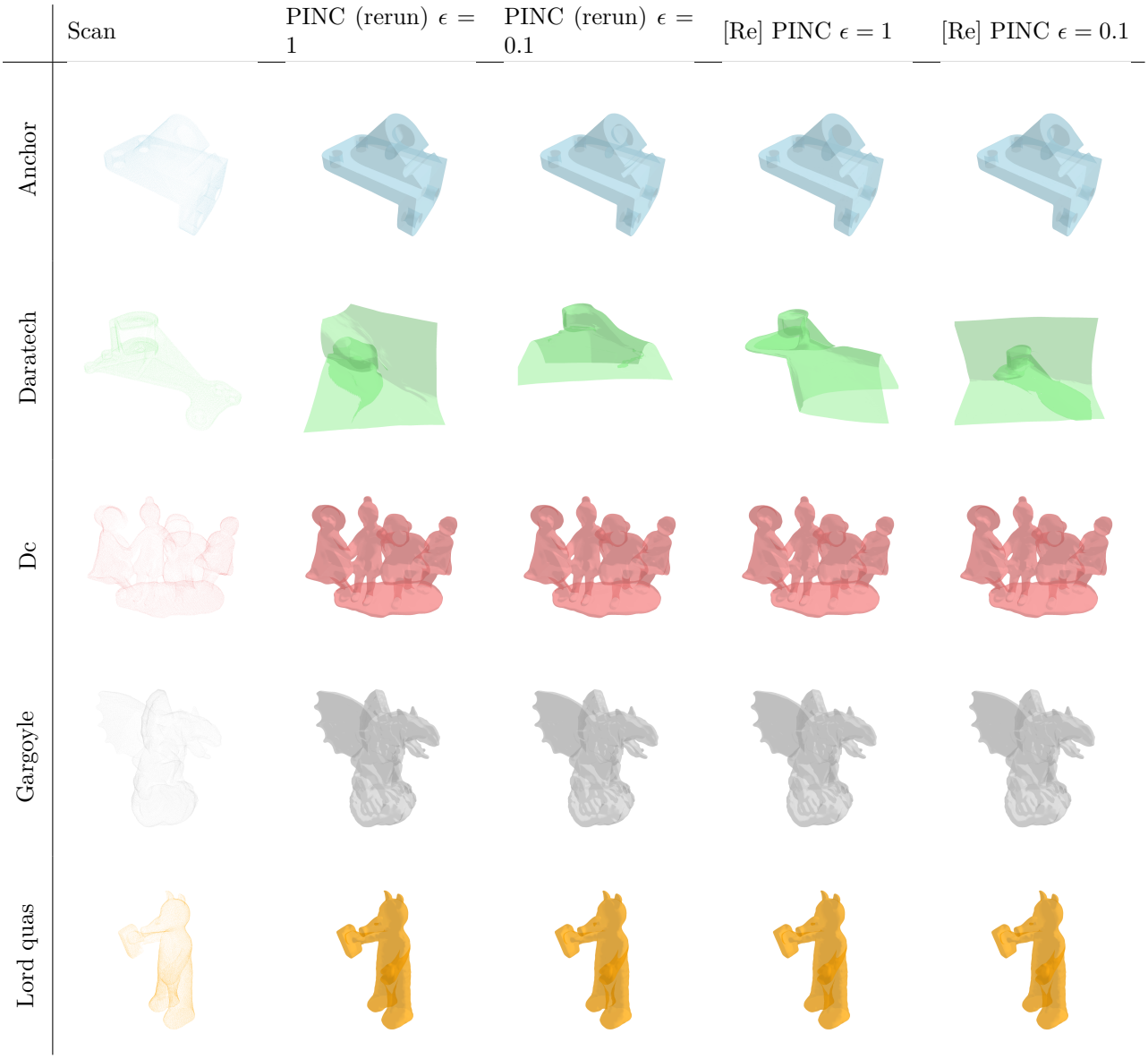

Table 3: 3D Reconstruction results for SRB Dataset.

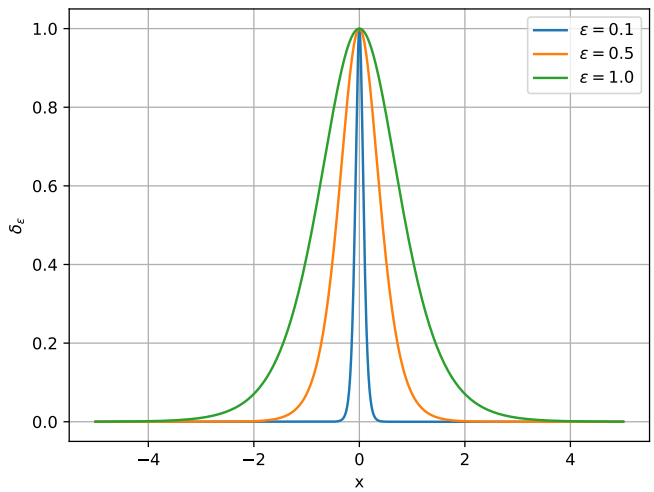

Figure 2: Plot of the function $\delta_\epsilon(x) = 1 - \tanh^2\left(\frac{x}{\epsilon}\right)$ for different values of ϵ