# Robust Neural Fields - Distance Field Approximation using MeshCNN

Julian Strietzel
Technical University of Munich
julian.strietzel@tum.de

Ata Jadid Ahari
Technical University of Munich
ata.ahari@tum.de

## Abstract

*Neural distance fields cope with different challenges including bad input shape representations and their missing ability of representing arbitrary surfaces. We root one issue with Signed Distance Field representations in the way that current SOTA-architectures extract the distance values from point-cloud representations. We suggest neural distance extractors as ground truth for the SDF overfitting process to become agnostic of problematic normals in input shape representations and suggest a modified version of MeshCNN as shape understanding regressor. Our contributions are three-fold: (1) We show issues in baseline SDF solutions that are SOTA and root those in their ground truth extraction, (2) provide some general insights on how to use a mesh understanding NN architecture as regression model, implementing it as distance extractor for SDF overfitting and (3) demonstrate how it can beat baselines in specific scenarios.*

## 1. Introduction

Recently neural fields were used successfully to represent 3d shapes. Advancements in signed and unsigned distance fields (SDF/UDF) showed significant improvements in representation learning. Still UDFs have issues when it comes to representing surfaces in their respective thickness and SDFs are restrictive in the type of possibly represented shapes and require quality inputs, including perfectly defined normals, that are not broadly available.

We rooted issues with 3d shape representation using Neural Fields in problematic normal representations in the point-clouds used as input for the overfitting. We evaluate several baseline experiments, showing that current methods are not working sufficiently and are heavily dependent on accurate normal representation that is unlikely. To get more agnostic of this, we suggest using a neural network (NN) structure as ground truth for the SDF overfitting. This, instead of inferring the distance sign directly using the normals, understands the overall shape structure and infers inside/outside information directly from the mesh representation.

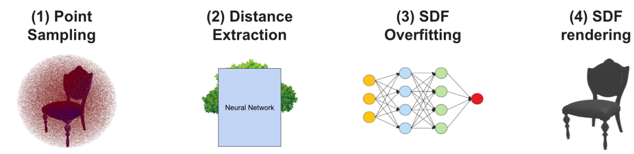


Figure 1. Architectural visualization

We suggest a for this purpose significantly modified version of MeshCNN, which is able to understand Mesh representations of shapes. We enhance input with a positional encoding for a requested point and train this model to regress distance field values as ground truth for a SDF overfitting process. Therefore, we are using well defined input meshes and expect it to generalize to worse defined ones, as it does not as rely on the normals as the baseline approach does.

Our ablation studies show how different modifications on MeshCNN and its training have significant impact on its performance as distance extractor. The promising results show how MeshCNN itself learns to represent an accurate distance field representation of the input mesh, being more agnostic to problematic and intentionally inverted normals than our baseline. Tests on generalization show room for improvement, where we suggest multiple steps to future research in this field.

## 2. Related works

In the past few years, a lot of advances have been made in 3D surface reconstruction with Neural Implicit Functions (NIFs). The NIF approaches use either occupancies [7] or signed distance field (SDF) functions [9] to represent 3d shapes and scenes and afterwards render it using marching cubes [5].

BACON [3] introduces a multiscale scene representation framework employing band-limited functions to facilitate efficient feature extraction. It employs windowed positional encoding to aid learning dynamic shapes via a coarse-to-fine training. Although it performs well on most shapes, it relies on normals to calculate the SDF and fails to represent

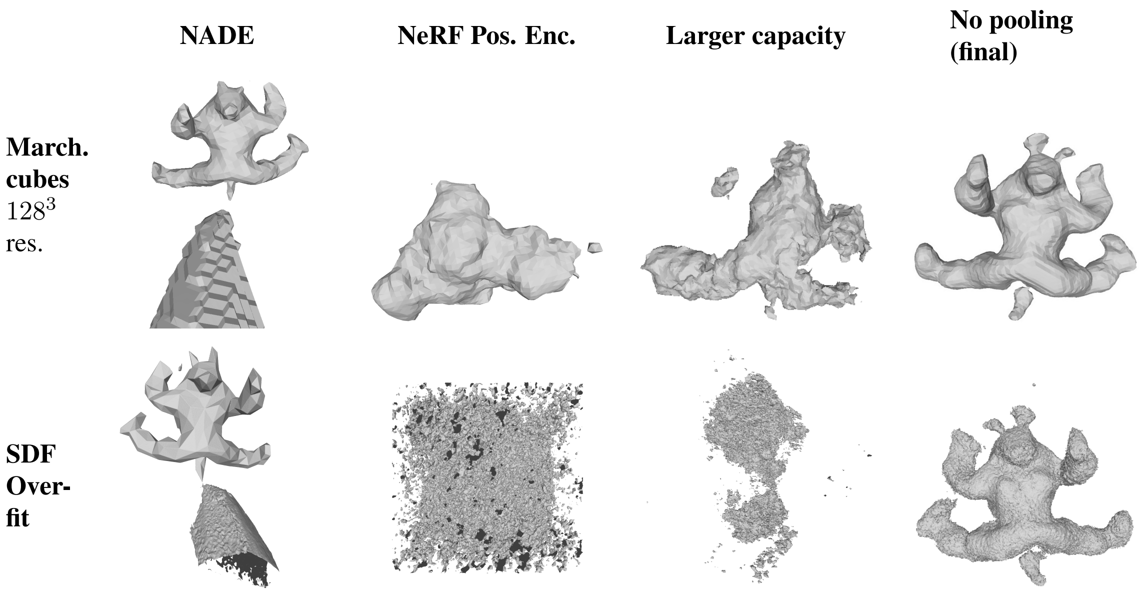


Table 1. Results and development over ablations

shapes with contradicting normals on edges robustly.

Learning Unsigned Distance Functions To model general shapes with open and multi-layer surfaces, NDF [4] learns unsigned distance functions to represent shapes by predicting the unsigned distance from a query location to the continuous surface. However, NDF requires time-consuming and non-trivial post-processing for mesh generation since methods such as marching cubes cannot be directly applied on them.

## 3. Methodology

**Architecture** We used a similar architecture which was used in BACON as our main architecture. It consists of three main steps. First, points from a 3d coordinate space near the surface of the input object are randomly sampled. Then, BACON uses the three closest neighbors on the surface to the selected points and calculates the SDF values using the coordinates and normals of these neighboring points. We call this part of the pipeline as Distance extraction and the method used in BACON as Normal Aware Distance Extraction (NADE), since it heavily relies on normals to calculate the SDF values. In the next step, the calculated SDF values are overfitted to an MLP. In the end, we can query this network on random points and construct a surface using marching cubes.

**MeshCNN** We used MeshCNN as our distance extractor instead of the NADE scheme. MeshCNN leverages a convolutional neural network (CNN) architecture to operate directly on mesh structures. It introduces a novel edge-aware convolution operation and a hierarchical mesh pooling strategy, allowing it to capture complex geometric features in 3D meshes, making it effective for tasks such as shape classification and segmentation. We modified the edge features and concatenated the sampled point coordinates, directly or after point encoding using [cite] to its already edge features. We also change the loss to Mean Squared Error in order to perform a regression task of predicting SDF values instead of mesh classification. MeshCNN will act as a mesh encoder, and this will help learn better SDF values as opposed to just sampling points and estimating SDF values from them. The ground truth SDF values were calculated using the NADE method.

## 4. Results

Our comprehensive evaluation demonstrates the effectiveness of leveraging MeshCNN as a neural distance extractor for enhancing SDF overfitting processes. This section delineates the outcomes of our experiments, underscoring the comparative analysis between our proposed methodology and SOTA baseline SDF solutions (NADE).

We use a Chamfer Distance (CD as defined in [6]) and the respective visual results qualitatively to analyze our methods. As a tool for debugging our implementation and gaining quick insights in the quality of the distance extractors, we render the outputs of our distance extraction methods directly using marching cubes on the input that we would otherwise feed to the SDF overfitting process as ground truth (see methodology section 3).

**Performance Comparison** Our direct and SDFs overfitted renderings on NADE using meticulously defined, wa-

| **Identifier** | ↓ **MAE** ($\times 10^{-3}$) | ↑ **Sign Acc. (%)** | ↓**Chamfer Dist.**($\times 10^{-2}$) |
|---|---|---|---|
| MLP Baseline | 63.00 | - | - |
| **MeshCNN - Base*** | 9.67 | - | 5.55 |
| **NeRF Posi. Encode.** | 8.48 | - | 3.29 |
| **No augment.**** | 6.79 | 73.80 | 2.96 |
| **Mesh Normaliz.** | 6.27 | 74.39 | 6.23 |
| **Deactivated Relu** | 6.03 | 74.81 | 5.23 |
| Combined Loss | 7.76 | 70.36 | 5.94 |
| Sign penalty loss | 35.14 | 71.11 | 7.74 |
| **6 frequency encoding** | 5.86 | 75.60 | 2.82 |
| **Long Train**** | 5.72 | 75.53 | 3.24 |
| Larger capacity | 5.59 | 74.47 | 3.09 |
| **LR x 16**** | 4.22 | 77.81 | 3.28 |
| **No pooling** | **4.13** | **78.33** | **2.76** |
| Generalization | 5.90 | 75.25 | 5.26 |

Table 2. Mean absolute error, sign accuracy and chamfer distance (of rendering using marching cubes) from ablation studies on architectural and training changes. Read from top to bottom: The fat names describe changes resulting in a new best-score from the previous best.
* Base refers to the default MeshCNN architecture using x, y, z coordinates attached as positional encoding.
** Changes related to training parameters.

tertight, and non-manifold test shapes from the ShapeNet dataset [1], showcase high-fidelity results. When compared to the MeshCNN results, it shows significantly better high- and low-level details, which appear sharper and more defined.

But our objective is not to enhance SDF accuracy per se but to increase its robustness against poorly defined normals. In this matter, we see huge undesired artifacts in both renderings using NADE, which we assume being caused by bad shape representations when using the normals of the point-cloud input representation (see figure 3). On the other hand, our MeshCNN distance extractor seems unaffected by such issues, even though its supervision is still based on NADE.

To illustrate this even further, we introduced artificial degradation to 5% and 30% of the normals in our test shape and employed NADE for distance extraction. The renderings (3) reveal that compromised normals significantly destroy output quality when using NADE, introducing artifacts and loss of detail. Conversely, MeshCNN's performance remains unaffected by such degradation, due to its inherent robustness to normal quality and underscores its reliance on mesh representation rather than normal input.

**Ablation Studies and Hyper-parameter Tuning** Our subsequent ablation studies (see table 2) illuminate the impact of various modifications to MeshCNN on its efficacy as a distance extractor. Beginning with a baseline implementation inspired by Hanocka [2], we appended the coordinates of the sample point to the edge feature inputs, markedly surpassing previous MLP baseline methods that we to feed with relative nearest neighbors. Incorporating sinusoidal

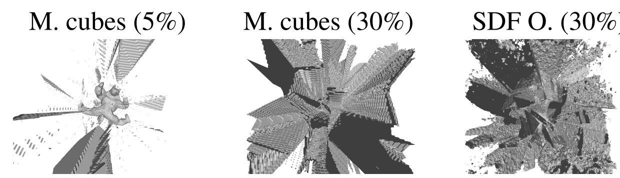


Table 3. NADE (direct marching cubes rendering and SDF overfitted) on different % of inverted normals.

positional encoding, as proposed in NeRF [8, 10], resulted in a 12% improvement in Mean Absolute Error (MAE), further enhanced by increasing encoding frequencies (-3%).

Other architectural adjustments to MeshCNN also yielded notable improvements. Aligning mesh normalization with the BACON pipeline expectations enhanced performance (-7%). Removing a terminal ReLU activation in each MeshConvolution benefited regression accuracy (-4%), as did minimizing pooling dimensional reductions to preserve shape resolution (-2%).

Furthermore, tuning parameters primarily related to the training process also proved beneficial. For instance, deactivating augmentations led to a significant accuracy boost (-20%). Although this may require reevaluation for broader input distributions during generalization expeiments, we strategically decided to isolate and understand thee model's learning capabilities, debug and validate implementations, and focus on overfitting to specific scenarios as part of the research process. Optimizing learning rate schedules combined with a doubled learning rate decay phase further refined our model's performance (-28%).

**Custom Loss** Ensuring accurate signs can be more relevant than minimizing absolute distance errors in certain applications, such as mesh reconstruction or surface rendering. We therefore experimented using a custom loss function (in combination with MSE loss and solo) to penalize a wrongly predicted sign. This loss also required extensive tuning and some other complications, so we lastly decided to keep the MSE loss for its superior performance and ease of use and leave this to future work.

$$\frac{1}{N}\sum_{i=1}^{N}\begin{cases}(output_i - target_i)^2 & \text{if } output_i \cdot target_i < 0\\ 0 & \text{otherwise}\end{cases} \tag{1}$$

### 4.1. Generalization Capability

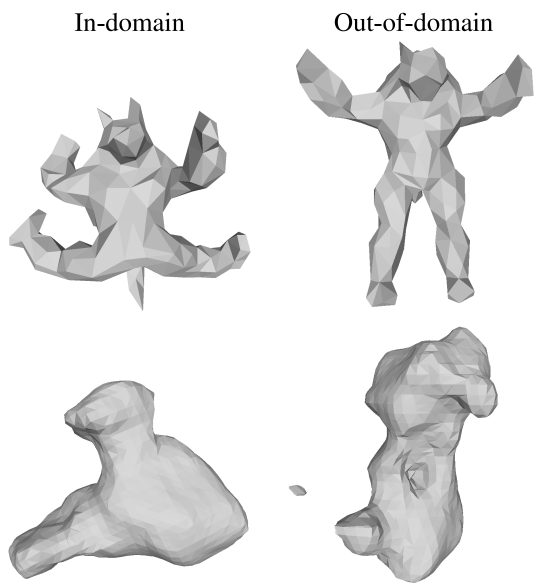


Table 4. MeshCNN: Reference and marching cubes renderings on seen and unseen data with reference gt objects.

Exploring the generalization capabilities of our model, we present results from tests conducted on a small set of input shapes to train our MeshCNN distance extractor and its performance on in-domain and out-of-domain shapes. Firstly, we observed a shallower learning curve and significantly longer training. We therefore could not extend our experiments to larger dataset sizes. Rendering in-domain shapes we saw that this translated to a lesser ability to understand the shapes (see figure 4) and the representational accuracy. Despite these challenges, our model maintained its ability to capture (very-)low-frequency details across different inputs (see figure 4), illustrating a foundational understanding of shape orientation and geometry.

These results suggest significant avenues for future research, particularly in exploring the scalability of neural distance extractors to larger datasets and more complex models. Our findings highlight the potential of neural distance extractors in 3D shape representation.

## 5. Conclusion

Our results offer profound insights into the limitations of current SDF solutions and propose a promising pathway to overcome these through the use of neural distance extractors. We demonstrate how basic distance extractors for SDFs struggle with seemingly easy shapes and destroyed normals and how alternative solutions significantly reduces the output quality. By decoupling the overfitting process from the quality of input normals, we enable more robust and accurate shape representations. This advancement not only broadens the applicability of neural field technologies but also sets a new baseline for future research in 3D shape representation. This implicates a new direction in the field of Neural Distance Fields, that is worth-full to dive into.

**Limitations & Future Work** In our research, we uncovered several promising avenues for further investigation. Our findings demonstrate that neural network-based distance extractors can overfit to a single shape, surpassing the performance of traditional baselines. However, our project was limited due to constraints on time and resources. To fully understand NN Distance Extractor's generalization potential, we recommend additional studies with larger datasets and networks with greater capacity.

Moreover, we suspect that the sign of the distance extracted is the most relevant feature for accurately rendering shapes. To improve the network's accuracy in determining the correct sign of distances, it would be beneficial to investigate diverse loss functions that specifically penalize inaccuracies in sign prediction. Furthermore, to cope with the networks potential limited accuracy, it could be explored how non-neural network methods of high-fidelity distance extraction from meshes could be integrated with our neural network to enhance its ability to understand details of a model's exterior and interior. This direction promises to refine the network's performance and broaden its applicability in rendering and understanding complex shapes.

Lastly, it's important to note that we are currently still utilizing the same method we aim to improve upon in the SDF overfitting process within our method. Specifically, the ground truth for MeshCNN training is generated using NADE. Up to this point, we have primarily trained MeshCNN on well-defined shapes, where this should not be an issue. However, even in the case of problematic shapes, MeshCNN has demonstrated remarkable abilities to abstract away from introduced artifacts. These capabilities, along with the performance on more challenging training data, require further evaluation.

## References


[1] Angel Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. 2015. 3

[2] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. MeshCNN: a network with an edge. *ACM Transactions on Graphics*, 38(4):90:1–90:12, 2019. 3

[3] David B. Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. BACON: Band-limited coordinate networks for multiscale scene representation. In *CVPR*, 2022. 1

[4] Xiaoxiao Long, Cheng Lin, Lingjie Liu, Yuan Liu, Peng Wang, Christian Theobalt, Taku Komura, and Wenping Wang. *NeuralUDF: Learning Unsigned Distance Fields for Multi-view Reconstruction of Surfaces with Arbitrary Topologies*. 2022. 2

[5] 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. 1

[6] Jinxiao Lu, Zhizhen Li, Jiquan Bai, and Qian Yu. Oriented and Directional Chamfer Distance Losses for 3D Object Reconstruction From a Single Image. *IEEE Access*, 10:61631–61638, 2022. 2

[7] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[8] 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, 2020. _eprint: 2003.08934. 3

[9] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. Conference Name: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) ISBN: 9781728132938 Place: Long Beach, CA, USA Publisher: IEEE. 1

[10] Lin Yen-Chen. NeRF-pytorch, 2020. Publication Title: GitHub repository. 3