

# Learning Continuous Implicit Field with Local Distance Indicator for Arbitrary-Scale Point Cloud Upsampling

Shujuan Li<sup>1\*</sup>, Junsheng Zhou<sup>1\*</sup>, Baorui Ma<sup>1,2</sup>, Yu-Shen Liu<sup>1†</sup>, Zhizhong Han<sup>3</sup>

<sup>1</sup>School of Software, Tsinghua University, Beijing, China,

<sup>2</sup>Beijing Academy of Artificial Intelligence

<sup>3</sup>Department of Computer Science, Wayne State University, Detroit, USA

lisj22, zhoujs21@mails.tsinghua.edu.cn, brma@baai.ac.cn, liuyushen@tsinghua.edu.cn, h312h@wayne.edu

## Abstract

Point cloud upsampling aims to generate dense and uniformly distributed point sets from a sparse point cloud, which plays a critical role in 3D computer vision. Previous methods typically split a sparse point cloud into several local patches, upsample patch points, and merge all upsampled patches. However, these methods often produce holes, outliers or non-uniformity due to the splitting and merging process which does not maintain consistency among local patches. To address these issues, we propose a novel approach that learns an unsigned distance field guided by local priors for point cloud upsampling. Specifically, we train a local distance indicator (LDI) that predicts the unsigned distance from a query point to a local implicit surface. Utilizing the learned LDI, we learn an unsigned distance field to represent the sparse point cloud with patch consistency. At inference time, we randomly sample queries around the sparse point cloud, and project these query points onto the zero-level set of the learned implicit field to generate a dense point cloud. We justify that the implicit field is naturally continuous, which inherently enables the application of arbitrary-scale upsampling without necessarily retraining for various scales. We conduct comprehensive experiments on both synthetic data and real scans, and report state-of-the-art results under widely used benchmarks. Project page: <https://lisj575.github.io/APU-LDI>

## Introduction

Point cloud, as a lightweight representation of 3D objects and scenes, has gained significant attention. It can be easily obtained through depth cameras or other 3D scanning devices. However, the raw point clouds obtained from these devices are often sparse and non-uniform, which negatively impacts various downstream applications like semantic understanding (Zhang et al. 2019), surface reconstruction (Chen, Liu, and Han 2023b), and point cloud rendering (Dai et al. 2020). Hence, it is crucial to generate dense and uniformly distributed points from the raw point cloud data.

Recently, the learning-based methods on point cloud upsampling have achieved promising results. The mainstream methods typically split a sparse point cloud into several

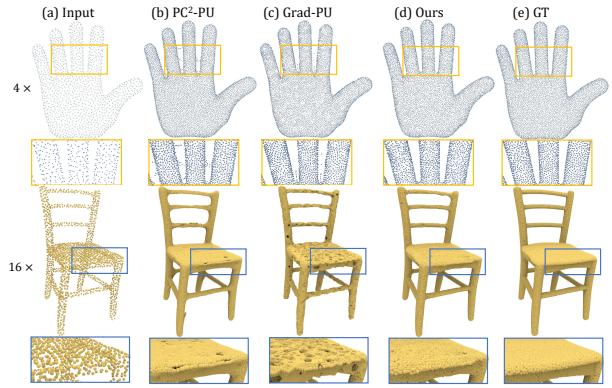


Figure 1: Comparison with PC<sup>2</sup>-PU (Long et al. 2022) and Grad-PU (He et al. 2023). The input point cloud has 2048 points, and ground truth has 8192 points for the palm and 32768 points for the chair. Previous methods upsample the point cloud at the local level, which results in outliers, holes and non-uniform when combining local patches. We represent the entire point cloud as a continuous implicit field and generate globally consistent results.

patches, in which a neural network is trained for predicting surface signals, such as surface points (Yu et al. 2018; Li et al. 2019; Qian et al. 2021a; Feng et al. 2022), offsets (Li et al. 2021), interpolation weights (Luo et al. 2021; Qian et al. 2021b), and the point-to-surface distances (He et al. 2023; Zhao et al. 2023). With the help of learned signals, points on each patch become denser. Finally, the upsampled patches are combined into a complete point cloud as the final prediction. However, the inconsistency among local patches causes two issues as shown in Figure 1. First, the geometry of the patch boundary is hard to determine without the global shape information, resulting in outliers on the final dense point cloud. Second, combining inconsistent patches leads to holes or non-uniformity on the final point cloud.

To solve these issues, we propose to learn an unsigned distance field to represent a shape with the guidance from a pre-trained local distance indicator (LDI) learned from dense point cloud patches at local level. With the continuous implicit field and local distance prior, we produce accurate upsampling results which capture rich geometric details and

\*These authors contributed equally.

†Corresponding author: Yu-Shen Liu

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

also retain uniformity due to the global consistency of the entire shape. Specifically, we first train a LDI that predicts the distance from a query location to the underlying surface, which aims to learn a sparse-to-dense prior on local patches.

To predict the distance more accurately and robustly, we further design an attention-based module to learn attention weights between the query point and its neighbouring patch points at the feature space. Given the learned LDI, we then utilize it as the supervision to guide the learning of the global implicit function which represents the underlying surface of the entire sparse point cloud. Finally, for randomly sampled query points, we use the implicit function to obtain the distances and the gradients, and then project queries onto a surface by pulling them along the direction of gradients at a stride of distances to form the dense point cloud. By introducing the continuous implicit function, we achieve an accurate and uniform point cloud upsampling with consistency among patches. We further justify that the continuity of the implicit surface can control the density of the dense point cloud by changing the number of sampled query points, which enables our method to achieve arbitrary-scale upsampling without retraining for various scales.

Our contributions are summarized as follows.

- We propose to learn an unsigned distance field from a sparse point cloud with the guidance of a pre-trained distance indicator at local level. By pulling 3D queries into the zero level set of the field, we can achieve point cloud upsampling at arbitrary-scales.
- We design an attention-based module to learn a local distance indicator that estimates point-to-surface distances, which brings rich local priors to the global implicit field.
- We achieve state-of-the-art results in arbitrary-scale point cloud upsampling for synthetic data and real scans on widely used benchmarks.

## Related Work

### Optimization-based Point Cloud Upsampling

Optimization-based methods generate dense point clouds by employing well-designed shape priors. As the pioneering work in point cloud upsampling, Alexa et al. 2003 propose to add points at the vertices of Voronoi diagrams. To preserve more geometrical structures, Huang et al. 2009 introduce weighted locally optimal projection operator and robust normal estimation process for point cloud upsampling. Later, Edge-Aware Resampling (EAR) (Huang et al. 2013) achieves fine detail preservation by resampling points away from edges and progressively upsampling them to approach the edge singularities. In addition, several methods are proposed to deal with different conditions such as noisy data (Lipman et al. 2007) and missing regions (Wu et al. 2015). Overall, optimization-based methods heavily rely on surface smoothness assumptions, which often leads to struggles between geometric detail and robustness.

### Learning-based Point Cloud Upsampling

Recently, with the development of deep learning, learning-based methods (Yu et al. 2018; Qian et al. 2020; Luo et al.

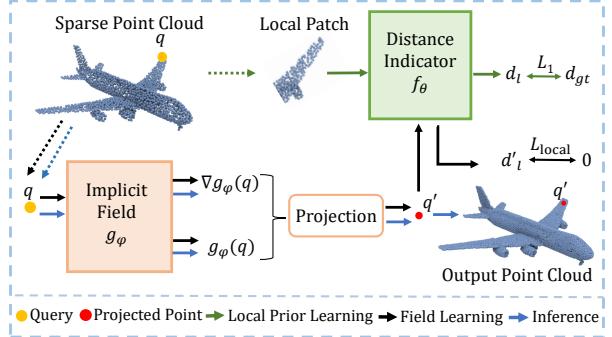


Figure 2: Overview of the proposed method. The key idea of our method is to learn an unsigned distance field to represent the sparse point cloud at global level with the guidance from a pre-trained LDI learned from dense local patches.

2021; Liu et al. 2022; Feng et al. 2022) are proposed for point cloud upsampling. Yu et al. (Yu et al. 2018) propose the first learning-based point cloud upsampling method PU-Net. PU-GAN (Li et al. 2019) introduces generative adversarial network (Goodfellow et al. 2014) to generate uniform point clouds. PU-GCN (Qian et al. 2021a) uses a Graph Convolutional Network to learn better local features. Considering the patch connection, PC<sup>2</sup>-PU (Long et al. 2022) introduces patch-to-patch and point-to-point modules to help the combination of different patches. Meta-PU (Ye et al. 2021) supports arbitrary scale factor upsampling using meta-network to adjust the weights of residual graph convolution blocks dynamically. PU-SMG (Dell’Eva et al. 2022) maps the sparse point cloud to the Spherical Mixture of Gaussians. More recently, Grad-PU (He et al. 2023) samples more accurate point clouds by predicting point-to-point distances. PU-SSAS (Zhao et al. 2023) uses implicit neural functions to estimate projection distance and direction with unsigned distance and normal supervision. Existing methods focus on the geometrical structure at the local level and use networks to learn rich patch information, but the ambiguous boundaries at local patches always lead to issues like holes, outliers and non-uniformity. Although PC<sup>2</sup>-PU tries to alleviate the connection problem between patches with patch correlation module, they still suffer from outliers for the lack of global consistency of the global shape. We propose to learn an unsigned distance field of the entire shape with local priors, which introduces globally consistent information.

### Implicit Neural Representation

Implicit neural representation has shown great advantages in continuous surface representation (Zhang et al. 2023; Ma et al. 2023). Usually, the implicit function is learned by a neural network, the input of the network is a query point, and the output is occupancy probability (Mescheder et al. 2019; Chen and Zhang 2019) or signed/unsigned distance (Park et al. 2019; Michalkiewicz et al. 2019). Moreover, Neural Radiance Fields (NeRFs) (Mildenhall et al. 2021) use implicit representations to encode the geometry and appearance information. Recently, some researches (Ma et al.

2022; Zhou et al. 2023; Jin, Wu, and Zhou 2023; Chen, Liu, and Han 2023a) propose to learn implicit continuous surfaces from 3D point clouds. Among them, OnSurfacePrior (Ma et al. 2022) proposes to learn surface priors to guide the learning of signed distance functions (SDF). However, directly transferring this method to the point cloud upsampling task leads to oversmoothed surfaces due to the MLP-based prior network and the lack of sufficient global constraints. In our method, the designed attention-based distance indicator guides the neural network to learn a global unsigned distance field that handles more complex shapes, and the additional global constraints help to preserve more details. More recently, some methods (Zhao et al. 2022, 2023) have achieved encouraging results by applying implicit representations to point cloud upsampling tasks, while these methods often require other shape information, such as normals and unsigned distances. Our method only requires dense point clouds as supervision.

## Methodology

### Overview

Given the input sparse point cloud  $S = \{s_i\}_{i=1}^N$ , point cloud upsampling aims to generate a dense point cloud  $T = \{t_j\}_{j=1}^M$ , where  $M = rN$  and  $r$  is the scale factor. The generated point cloud should meet the requirement that each point of the dense point cloud is located at the underlying surface represented by the sparse point cloud and the dense point cloud is uniformly distributed. The overview of our proposed method is shown in Figure 2. We learn an unsigned distance field to represent the continuous surface described by the sparse point cloud  $S$ , which is guided by the local distance indicator that predicts the distance from the query point to the local patch. At inference time, we project the query points to the zero-level set of the implicit field to generate the dense point cloud. In this Section, we will begin by introducing the specifics of the local distance indicator. Subsequently, we will delve into the learning process of the global implicit field, followed by the technique of arbitrary-scale upsampling on the continuous implicit field.

### Local Distance Indicator (LDI)

To learn more accurate geometrical details, we first train a neural network as a local distance indicator to learn upsampling priors from grouped patches in local level. For a sparse point cloud  $S$ , we first split it into different patches like previous methods (Long et al. 2022; He et al. 2023). Then, we sample queries around each patch point with normal distribution. The distance indicator takes the query point  $q$  and its corresponding local patch  $P$  as the input and predicts the distance  $d_l$  from the query point to the local patch in a local coordinate system as follows:

$$f_\theta(q, P) = d_l, \quad (1)$$

where  $f_\theta$  denotes the network parameterized by  $\theta$ .

To predict the distance more accurately and robustly, we further design an attention-based module to learn attention weights between the query point and its neighboring points at the feature space. The structure is shown in

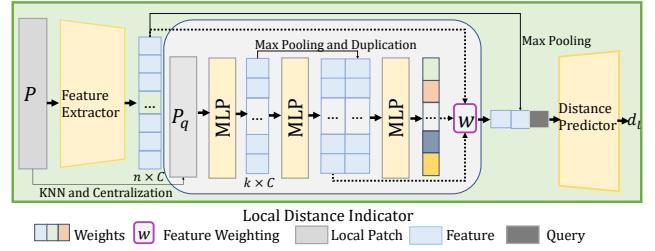


Figure 3: Structure of the local distance indicator. The attention-based distance indicator predicts the distance between the query point and the patch surface.

Figure 3. Specifically, for patch  $P = \{p_h\}_{h=1}^n$ , we use the feature extractor to obtain the point-wise patch features  $F = \{f_h\}_{h=1}^n$  for patch points. Then we find a smaller region  $P_q = \{p_l\}_{l=1}^k$  where  $k < n$  with k-nearest neighbor algorithm (KNN) for query point  $q$ . We further normalize  $P_q$  by centering it to the query point  $q$  followed by MLPs to extract the relative features  $F' = \{f'_d\}_{d=1}^k$ . To add the global information of the patch to the features, we use max-pooling on the relative features to generate the global feature. Finally, we concatenate the relative features and global features together and predict the relative weights  $w_d$  between the query point and the neighboring points. With the relative weights, we find the most important features on the local patch. We also use the relative features of  $q$  and the patch points to add relative distance perception for more accurate prediction. The final feature  $f_q$  for  $q$  is computed by weighting patch features and relative features:

$$f_q = \sum_{d=1}^k w_d \cdot f'_d + (1 - w_d) \cdot f_d. \quad (2)$$

Then, we concatenate  $f_q$ , the global feature of patch features and the query point coordinate together and use a distance predictor to map the final vector to point-to-surface distance  $d_l$ . We train the network by minimizing the L1 loss between the predicted distance and the distance  $d_{gt}$  between the query point and its nearest point in the dense point cloud:

$$L_1 = |d_l - d_{gt}|. \quad (3)$$

We visualize the weights learned by the attention module in Figure 4. For a query point, we select its four neighbors to generate its final feature. The color map indicates the values of the weights, where the darker the color, the greater the weight. The results show that the closer points have relatively larger weights, which proves that the learned weights are reasonable. In addition, the value of the weights are not strictly positively correlated with the distances, which proves that the attention module is adaptable.

### Learning the Continuous Field

With the learned LDI, we can learn the global implicit field to represent the entire point cloud. As shown in Figure 2, for the query point  $q$ , we use the neural network  $g_\phi$  that is parameterized by  $\phi$  to predict its distance  $d_g$  to the surface

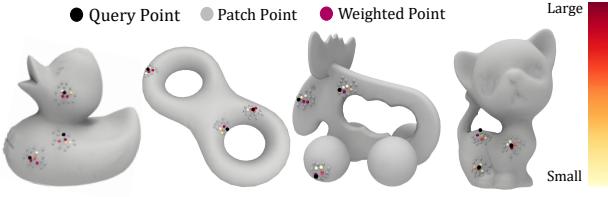


Figure 4: The visualization of learned weights.

in the global coordinate system. Here, we use the pulling process introduced by (Ma et al. 2021) to project the query point  $q$  to the surface point  $q'$ , formulated as,

$$q' = q - g_\phi(q)\nabla g_\phi(q)/\|g_\phi(q)\|_2, \quad (4)$$

where  $\nabla g_\phi(q)$  is the gradient at  $q$  within the network. We aim to learn to project  $q_i$  to the underlying surface represented by the sparse point cloud  $S$ . However, the real position of  $q'$  is unknown due to the sparsity of  $S$ , where we leverage our learned LDI as supervision for guiding the implicit field learning process.

Specifically, we use KNN to find the local patch  $P_{q'}$  that  $q'$  is centered at and use the fixed indicator to predict the distance  $f_\theta(q', P_{q'})$  from  $q'$  to  $P_{q'}$  at the local level. We justify that the indicated distance  $f_\theta(q', P_{q'})$  of  $q'$  increases as  $q'$  gets projected to locations far from the underlying surface and decreases when  $q'$  is projected close to the underlying surface. Additionally,  $f_\theta(q', P_{q'}) = 0$  means that  $q'$  is projected to the correct position. Therefore, we minimize the L1 loss between the predicted distance and 0:

$$L_{\text{local}} = |f_\theta(q', P_{q'})|. \quad (5)$$

To take full advantage of the information of the global point cloud, we also add some additional constraints.

At the early stage of training, the global network  $g_\phi$  projects the query point to the location away from the surface, leading to large  $f_\theta(q', P_{q'})$ , which makes the optimization difficult. Therefore, we introduce the surface point constraint that encourages  $q'$  to get projected closer to its nearest point  $q_s$  on the sparse point cloud:

$$L_{\text{np}} = \|q' - q_s\|. \quad (6)$$

To make the implicit field faithfully describe the underlying surface described by the sparse point cloud, we add a term to encourage the zero-level set of the implicit field to pass through the input points  $\{s_i\}_{i=1}^N$ . The loss term is:

$$L_{\text{surf}} = |g_\phi(s_i)|. \quad (7)$$

Due to the approximation error of the point-to-surface distance, the gradient of the implicit field is not the exact normal direction, leading to errors when  $q'$  is very close to the underlying surface. To learn a more accurate distance and direction of the query point  $q$ , we add a shortest path constraint introduced by (Ma et al. 2022):

$$L_{\text{sp}} = |g_\phi(q)|. \quad (8)$$

The total global loss is given as:

$$L_{\text{global}} = \alpha \cdot L_{\text{np}} + \beta \cdot L_{\text{surf}} + \gamma \cdot L_{\text{sp}}, \quad (9)$$

where  $\alpha$ ,  $\beta$  and  $\gamma$  are the weights for different loss terms.

Therefore, the total loss is  $L = L_{\text{local}} + L_{\text{global}}$ . After training, the global implicit field can be leveraged to predict the projection direction and distance for an arbitrary query point to reach the underlying surface. The advantage of the implicit function is that the neural network learns the relation between the spatial point and the underlying surface, so it can represent the shape with arbitrary precision theoretically, which leads to a continuous surface representation. Next, we show the details of arbitrary-scale point cloud upsampling with the implicit field.

## Arbitrary-Scale Point Cloud Upsampling

With the learned global implicit field, each query point can be projected to the continuous underlying surface. We change the number of query points to achieve arbitrary-scale upsampling without retraining. For the surface point  $s \in S$ , we sample offsets from a uniform distribution that is bounded by zero and the nearest distance of  $s$  and other points in  $S$ . Then, we add the offsets to  $s$  to form the queries. We use the implicit field to predict the gradient and distance for each query point, and project it to the surface by pulling  $q$  along gradient  $\nabla g_\phi(q)$  at a stride of the predicted distance  $g_\phi(q)$  with Eq. 4. To make the upsampled point cloud more uniform, we generate more query points than the target and employ farthest points sampling (FPS) (Eldar et al. 1997) algorithm to downsample it to the target number.

## Implementation Details

For the local distance indicator, the input patches are normalized and interpolated using the same strategies in (He et al. 2023). During the learning of global implicit field, we adopt the similar strategy as (Zhou et al. 2022) to sample about 240 queries around each point  $s_i$ . Specifically, we add Gaussian noises to the sparse point cloud to generate query points where the mean is 0 and the standard variance is 20% of the distance between  $s_i$  and its 50-th nearest point on the point cloud. We also add some query points randomly sampled on the whole space.

We employ network structures of encoder and decoder similar to (He et al. 2023) for the feature extractor and distance predictor individually. The encoder mainly consists of several dense blocks (Huang et al. 2017) and the decoder contains several MLP layers with ReLU activation. Besides, we adopt a neural network similar to DeepSDF (Park et al. 2019) to represent the global implicit field, which is composed of 8 fully connected layers with a residual connection.

## Experimental Results

### Experiment Setup

**Dataset.** We adopt two public synthetic datasets to evaluate our method. First, we make a comparison with the baseline methods on the PU-GAN dataset provided by (Li et al. 2019). This dataset contains 147 objects with various geometries, and 120 objects are used for training and the rest are used for testing. Second, we adopt a more complex dataset PU1K provided by (Qian et al. 2021a), which contains 1,147 3D models collected from different categories,

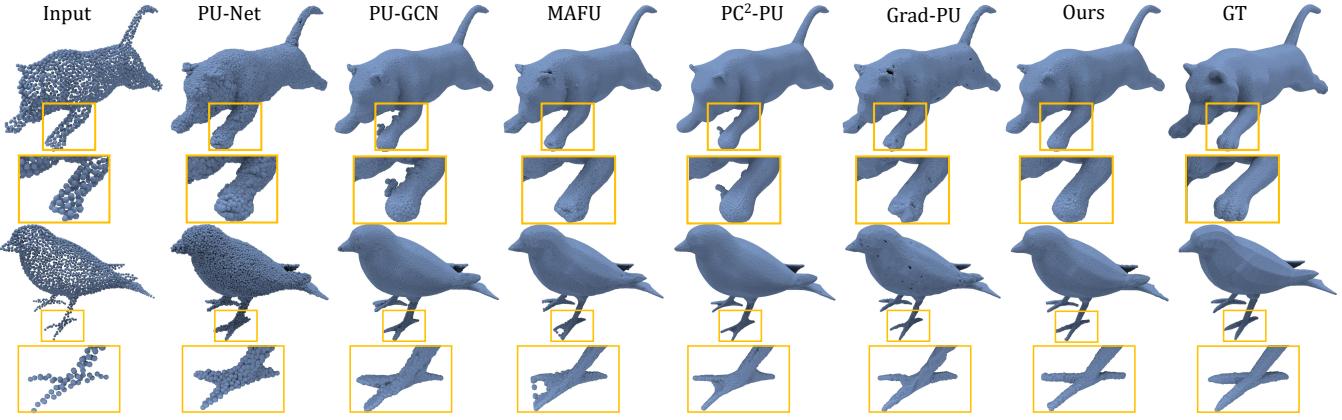


Figure 5: Qualitative comparisons of  $16\times$  upsampling results on the PU-GAN dataset. Our method generates uniform dense point clouds with fewer holes and outliers than state-of-the-art methods.

Factor	$4\times$ (r=4)			$16\times$ (r=16)		
	CD $\downarrow$ $10^{-3}$	HD $\downarrow$ $10^{-3}$	P2F $\downarrow$ $10^{-3}$	CD $\downarrow$ $10^{-3}$	HD $\downarrow$ $10^{-3}$	P2F $\downarrow$ $10^{-3}$
Methods						
PU-Net	0.507	4.312	4.694	0.596	6.929	6.014
MPU	0.289	4.771	2.830	0.175	5.898	3.111
PU-GAN	0.284	3.301	2.660	0.202	5.045	2.996
Dis-PU	0.278	3.447	2.343	0.180	4.888	2.560
PU-GCN	0.274	2.974	2.535	0.144	3.713	2.731
NePs	0.385	5.615	1.642	0.147	8.851	1.925
PU-SMG	0.296	2.404	2.492	0.155	2.749	2.573
PC <sup>2</sup> -PU	0.255	2.504	2.119	0.111	2.806	2.351
MAFU	0.280	2.315	1.854	0.156	3.408	1.996
Grad-PU	0.260	2.462	1.949	0.132	2.421	2.190
Ours	<b>0.232</b>	<b>1.675</b>	<b>1.338</b>	<b>0.092</b>	<b>1.504</b>	<b>1.544</b>

Table 1: Quantitative comparison between our method and the state-of-the-art methods on the PU-GAN dataset.

and 1020 objects are used for training. We use the patch to form the training data provided by the authors, which contains the pairs of sparse patches with 256 points and dense patches with 1024 points. We use the sample data augmentation strategy like previous methods (He et al. 2023) to make data diverse, including normalizing, rotation and random scale. For testing, each input point cloud generated from the watertight meshes contains 2048 points and ground truth contains 8096 points. We use the training sets of the two datasets to learn the local priors, and then learn the implicit field on the sparse point cloud of the test. All experiments are conducted on a single GeForce RTX 3090 gpu. We set  $\alpha = 1.0$ ,  $\gamma = 0.1$ , and  $\beta$  decays linearly from 0.5 to 0.

**Evaluation Metrics.** To evaluate the performance of point cloud upsampling, we use Chamfer distance (CD), Hausdorff distance (HD), and point-to-surface distance (P2F) as evaluation metrics following (He et al. 2023).

**Arbitrary-Scale Point Cloud Upsampling.** We compare our method with previous arbitrary-scale upsampling meth-

Methods	CD $\downarrow$ $10^{-3}$	HD $\downarrow$ $10^{-3}$	P2F $\downarrow$ $10^{-3}$
PU-Net	1.155	15.170	4.834
MPU	0.935	13.327	3.511
PU-GCN	0.585	7.577	2.499
Dis-PU	0.485	6.145	1.802
PU-SMG	0.527	4.396	1.647
PU-Transformer	0.451	3.843	1.277
Grad-PU	0.404	3.732	1.474
Ours	<b>0.371</b>	<b>3.197</b>	<b>1.111</b>

Table 2: Comparison with the state-of-the-art methods on PU1K dataset.

ods including MAFU and Grad-PU on the PU-GAN dataset in Table 3. All methods are trained with scale factor  $4\times$  and are trained for one time. The results show that our method achieves the best performance in various scale factors.

**Baselines.** We compare the proposed method with several state-of-the-art methods and divide them into fix-scale methods and arbitrary-scale methods according to the scale factor. The fix-scale methods include PU-Net (Yu et al. 2018), MPU (Yifan et al. 2019), PU-GAN (Li et al. 2019), Dis-PU (Li et al. 2021), PU-GCN (Qian et al. 2021a), PC<sup>2</sup>-PU (Long et al. 2022) and PU-Transformer (Qiu et al. 2022), and the arbitrary-scale methods include MAFU (Qian et al. 2021b), NePs (Feng et al. 2022), PU-SMG (Dell’Eva et al. 2022), and Grad-PU (He et al. 2023). For a fair comparison, we used the official pre-trained models when available and re-trained the others with the published code with default parameter settings.

### Evaluation on Synthetic Dataset

**Results on PU-GAN Dataset.** As shown in Table 1, our method significantly outperforms the baseline methods including PU-Net (Yu et al. 2018), MPU(Yifan et al. 2019), PU-GAN (Li et al. 2019), Dis-PU (Li et al. 2021), PU-GCN (Qian et al. 2021a), NePs (Feng et al. 2022), PU-SMG

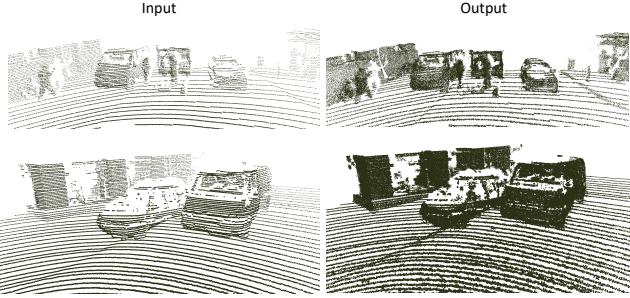


Figure 6: The results on KITTI dataset. Our method generates visual-appealing results.

Methods	3 × (r=3)			5 × (r=5)		
	CD↓ $10^{-3}$	HD↓ $10^{-3}$	P2F↓ $10^{-3}$	CD↓ $10^{-3}$	HD↓ $10^{-3}$	P2F↓ $10^{-3}$
MAFU	0.555	2.902	1.758	0.252	2.308	1.949
Grad-PU	0.524	3.050	1.778	0.244	2.447	2.459
Ours	<b>0.448</b>	<b>2.447</b>	<b>1.333</b>	<b>0.193</b>	<b>1.706</b>	<b>1.445</b>
Methods	7 × (r=7)			13 × (r=13)		
	CD↓ $10^{-3}$	HD↓ $10^{-3}$	P2F↓ $10^{-3}$	CD↓ $10^{-3}$	HD↓ $10^{-3}$	P2F↓ $10^{-3}$
MAFU	0.207	2.593	2.013	0.177	3.212	2.014
Grad-PU	0.223	2.654	3.325	0.242	4.144	5.703
Ours	<b>0.148</b>	<b>1.534</b>	<b>1.468</b>	<b>0.134</b>	<b>1.878</b>	<b>1.524</b>

Table 3: Quantitative comparison between our method and the state-of-the-art methods with various scale factors.

(Dell’Eva et al. 2022), PC<sup>2</sup>-PU (Long et al. 2022), MAFU (Qian et al. 2021b), and Grad-PU (He et al. 2023), indicating that learning continuous field helps to upsample points accurately on the surface. We also perform a visual comparison in Figure 5, which shows that our method can produce uniform point cloud with fewer holes and outliers. The qualitative and quantitative results for the scale factor 16× show that our method gains larger improvements. The reason behind is that the errors that produced by the local prediction accumulate with the scale factor increasing. Our method will not be affected by these due to the continuous field learning.

**Results on PU1K Dataset.** We evaluate our method on the more challenging PU1K dataset. We set the scale factor to 4× and report the comparison results with (Yu et al. 2018; Yifan et al. 2019; Qian et al. 2021a; Li et al. 2021; Qiu et al. 2022; Dell’Eva et al. 2022; He et al. 2023) in Table 2, which shows that our method outperforms others on all metrics.

### Evaluation on Real Scan Dataset

We validate our method on the real scanned datasets ScanObjectNN (Uy et al. 2019) and KITTI (Geiger et al. 2013) with the models trained on the PU-GAN dataset. We only show the qualitative results since the ground truth dense point clouds are unavailable. We utilize BallPivoting (Bernardini et al. 1999) to reconstruct the meshes for the

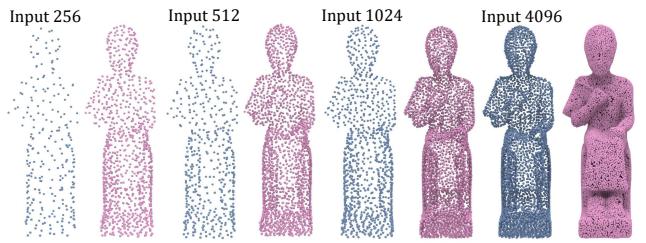


Figure 7: 4× results with various input sizes. The blue point clouds are inputs and the red point clouds are corresponding outputs. Our method generates consistent point clouds with various input sizes.

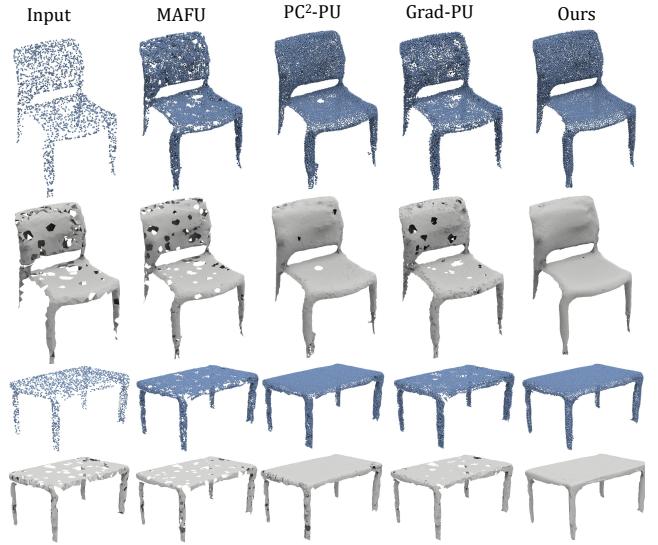


Figure 8: The 4× results on ScanObjectNN dataset. The meshes are reconstructed with the BallPivoting algorithm. The results show that our method generates more complete and smooth dense point clouds and meshes.

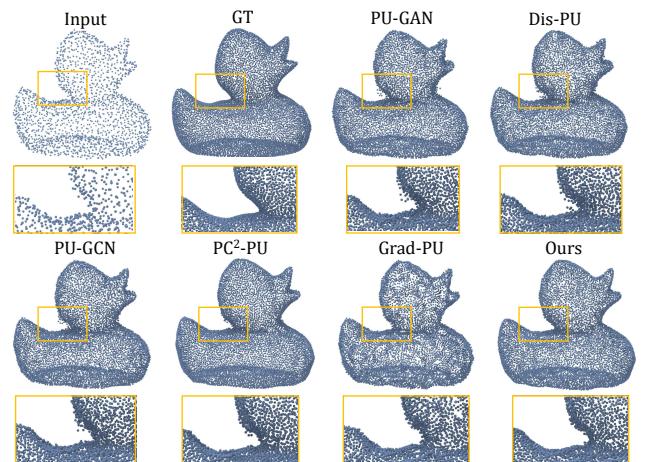


Figure 9: 4× results with noise level  $\tau = 1\%$ . The point cloud sampled by our method is cleaner with fewer outliers.

Noise Level	$\tau = 1\%$			$\tau = 2\%$		
	CD ↓ $10^{-3}$	HD ↓ $10^{-3}$	P2F ↓ $10^{-3}$	CD ↓ $10^{-3}$	HD ↓ $10^{-3}$	P2F ↓ $10^{-3}$
Methods						
PU-Net	0.588	6.182	9.842	1.057	9.954	16.282
PU-GAN	0.435	7.848	7.300	0.815	9.450	14.246
Dis-PU	0.430	6.580	6.954	0.776	8.861	13.934
PU-GCN	0.411	5.001	6.963	0.781	8.926	13.730
PC <sup>2</sup> -PU	0.369	4.390	5.646	0.733	7.921	12.610
Grad-PU	0.423	4.307	6.403	0.730	6.993	11.481
Ours	<b>0.339</b>	<b>3.089</b>	<b>5.167</b>	<b>0.622</b>	<b>6.485</b>	<b>10.984</b>

Table 4: Quantitative comparison on the PU-GAN dataset with different noise level  $\tau$ .

upsampled point clouds on ScanObjectNN and report the results in Figure 8 which show that our method can generate more complete and smooth point clouds and meshes. The testing results on KITTI in Figure 6 show that we achieve visual-appealing upsampling results. The comparison with other methods is provided in the supplemental materials.

### Robustness Test

**Various Input Size.** The input sizes of test point cloud in PU-GAN and PU1K are fixed to 2048. To prove that our method is robust to the input size variation, we conduct  $4 \times$  upsampling experiment on PU-GAN dataset. As shown in Figure 7, our method is not affected by the input sizes and generates consistent results in different point density.

**Additive Noise.** Following previous methods (Qian et al. 2021a; Long et al. 2022; He et al. 2023), we conduct  $r = 4 \times$  experiments on PU-GAN dataset to test the robustness of the proposed method. We add Gaussian noise to clean point clouds to generate noisy point clouds at two noise levels (1% and 2%). We compare our method with PU-Net, PU-GAN, Dis-PU, PU-GCN, PC<sup>2</sup>-PU and Grad-PU, and all of these methods are trained with the same augmentation strategy of Gaussian noise perturbation. The reported results in Table 4 show that our method achieves the best performances at both noise levels. Figure 9 shows that our method can produce point cloud with cleaner surface with fewer outliers, which proves the robustness of our method to noisy input.

### Ablation Studies

We conduct ablation studies on the subset of the PU1K dataset to show how each module influences the results. We randomly choose about 15% shapes from the testing set as the subset and the results are reported in Table 5.

**Module Design.** We first demonstrate the effectiveness of the attention-based module. We adopt the same interpolation operation as (He et al. 2023) to replace the attention weighting process with the inverse Euclidean distances with neighboring points. At local level, we use the same gradient descent (GD) strategy as Grad-PU. As shown by “Interpolation (GD)” and “Attention (GD)” in the label, the attention module improves the performance in all metrics. At global level, we use the interpolation based local distance

Ablation Settings	CD↓	HD↓	P2F↓
Interpolation (GD)	0.759	5.206	1.753
Attention (GD)	0.678	<u>4.931</u>	1.456
Interpolation (Global)	0.714	5.452	1.503
OnSurfacePrior	0.719	9.533	2.934
w/o $L_{local}$	0.874	6.523	1.819
w/o $L_{np}$	<b>0.555</b>	5.720	1.794
w/o $L_{sp}$	0.746	5.089	<b>1.344</b>
w/o $L_{surf}$	0.676	5.040	1.541
full	0.637	<b>4.864</b>	1.401

Table 5: Results of the ablation study for different experiment settings, with metrics CD( $\times 10^{-3}$ ), HD( $\times 10^{-3}$ ) and P2F( $\times 10^{-3}$ ). Bold and underlined numbers indicate the first and second best performance, respectively.

indicator to guide the learning of continuous field (“Interpolation (Global)”). Comparing with our attention-based indicator (“full”), we find a drop in performance, which shows that the attention-based module plays a positive role in the learning of the global implicit field.

As shown by “Interpolation (GD)” vs “Interpolation (Global)” and “Attention (GD)” vs “full”, the global implicit field strategy performances better than the gradient descent strategy in almost all metrics, which shows the global implicit field helps to generate point clouds with higher quality.

We conduct the experiment to compare our method with OnSurfacePrior (Ma et al. 2022) that also has the combination of local indicator and global learning as shown by “On-SurfacePrior” in Table 5. The results demonstrates that our designs are more adaptive to the point cloud upsampling.

**Loss Function.** We remove each loss term from the full setting to show how it influences the final result. As shown by “w/o  $L_{local}$ ”, the performance degenerates dramatically without the local distance indicator. As shown by “w/o  $L_{np}$ ”, the nearest point loss increases the CD, but it improves the other metrics significantly. The reason for the metric increase is that the nearest point on the sparse point is not exact the nearest point on the underlying surface, which leads to approximation errors. The third row “w/o  $L_{sp}$ ” shows that the shortest path loss improves the CD and HD, but leads to an increase on P2F for the introduced noises. The surface loss can improve the performance on all metrics.

### Conclusion

In this paper, we present a novel method to learn an unsigned distance field with a local distance indicator for arbitrary-scale point cloud sampling. Instead of upsampling the point sets at the local level as existing methods, we propose to represent the entire point cloud in the global implicit field to keep the consistency among local patches. We first design an attention-based local distance indicator that predicts the distance from a query location to the surface of the local patch, and then utilize it to guide the learning of implicit field for the point cloud. Extensive quantitative and qualitative comparisons on synthetic data and real scans demonstrate that our method outperforms prior state-of-the-art methods.

## Acknowledgments

This work was supported by National Key R&D Program of China (2022YFC3800600), the National Natural Science Foundation of China (62272263, 62072268), and in part by Tsinghua-Kuaishou Institute of Future Media Data.

## References

- Alexa, M.; Behr, J.; Cohen-Or, D.; Fleishman, S.; Levin, D.; and Silva, C. T. 2003. Computing and rendering point set surfaces. *IEEE Transactions on visualization and computer graphics*, 9(1): 3–15.
- Bernardini, F.; Mittleman, J.; Rushmeier, H.; Silva, C.; and Taubin, G. 1999. The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics*, 5(4): 349–359.
- Chen, C.; Liu, Y.-S.; and Han, Z. 2023a. Gridpull: Towards scalability in learning implicit representations from 3d point clouds. In *Proceedings of the ieee/cvf international conference on computer vision*, 18322–18334.
- Chen, C.; Liu, Y.-S.; and Han, Z. 2023b. Unsupervised inference of signed distance functions from single sparse point clouds without learning priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17712–17723.
- Chen, Z.; and Zhang, H. 2019. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5939–5948.
- Dai, P.; Zhang, Y.; Li, Z.; Liu, S.; and Zeng, B. 2020. Neural point cloud rendering via multi-plane projection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7830–7839.
- Dell’Eva, A.; et al. 2022. Arbitrary point cloud upsampling with spherical mixture of gaussians. In *2022 International Conference on 3D Vision (3DV)*, 465–474. IEEE.
- Eldar, Y.; Lindenbaum, M.; Porat, M.; and Zeevi, Y. Y. 1997. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing*, 6(9): 1305–1315.
- Feng, W.; Li, J.; Cai, H.; Luo, X.; and Zhang, J. 2022. Neural points: Point cloud representation with neural fields for arbitrary upsampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18633–18642.
- Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237.
- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.
- He, Y.; Tang, D.; Zhang, Y.; Xue, X.; and Fu, Y. 2023. Grad-PU: Arbitrary-Scale Point Cloud Upsampling via Gradient Descent with Learned Distance Functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5354–5363.
- Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.
- Huang, H.; Li, D.; Zhang, H.; Ascher, U.; and Cohen-Or, D. 2009. Consolidation of unorganized point clouds for surface reconstruction. *ACM transactions on graphics (TOG)*, 28(5): 1–7.
- Huang, H.; Wu, S.; Gong, M.; Cohen-Or, D.; Ascher, U.; and Zhang, H. 2013. Edge-aware point set resampling. *ACM transactions on graphics (TOG)*, 32(1): 1–12.
- Jin, C.; Wu, T.; and Zhou, J. 2023. Multi-grid representation with field regularization for self-supervised surface reconstruction from point clouds. *Computers & Graphics*.
- Li, R.; Li, X.; Fu, C.-W.; Cohen-Or, D.; and Heng, P.-A. 2019. Pu-gan: a point cloud upsampling adversarial network. In *Proceedings of the IEEE/CVF international conference on computer vision*, 7203–7212.
- Li, R.; Li, X.; Heng, P.-A.; and Fu, C.-W. 2021. Point cloud upsampling via disentangled refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 344–353.
- Lipman, Y.; Cohen-Or, D.; Levin, D.; and Tal-Ezer, H. 2007. Parameterization-free projection for geometry reconstruction. *ACM Transactions on Graphics (TOG)*, 26(3): 22–es.
- Liu, X.; Liu, X.; Liu, Y.-S.; and Han, Z. 2022. Spu-net: Self-supervised point cloud upsampling by coarse-to-fine reconstruction with self-projection optimization. *IEEE Transactions on Image Processing*, 31: 4213–4226.
- Long, C.; Zhang, W.; Li, R.; Wang, H.; Dong, Z.; and Yang, B. 2022. Pc2-pu: Patch correlation and point correlation for effective point cloud upsampling. In *Proceedings of the 30th ACM International Conference on Multimedia*, 2191–2201.
- Luo, L.; Tang, L.; Zhou, W.; Wang, S.; and Yang, Z.-X. 2021. Pu-eva: An edge-vector based approximation solution for flexible-scale point cloud upsampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16208–16217.
- Ma, B.; Han, Z.; Liu, Y.-S.; and Zwicker, M. 2021. Neural-Pull: Learning Signed Distance Function from Point clouds by Learning to Pull Space onto Surface. In *International Conference on Machine Learning*, 7246–7257. PMLR.
- Ma, B.; Zhou, J.; Liu, Y.-S.; and Han, Z. 2023. Towards better gradient consistency for neural signed distance functions via level set alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17724–17734.
- Ma, B.; et al. 2022. Reconstructing surfaces for sparse point clouds with on-surface priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6315–6325.
- Mescheder, L.; Oechsle, M.; Niemeyer, M.; Nowozin, S.; and Geiger, A. 2019. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4460–4470.

- Michalkiewicz, M.; Pontes, J. K.; Jack, D.; Baktashmotagh, M.; and Eriksson, A. 2019. Implicit surface representations as layers in neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4743–4752.
- Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.
- Park, J. J.; Florence, P.; Straub, J.; Newcombe, R.; and Lovegrove, S. 2019. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 165–174.
- Qian, G.; Abualshour, A.; Li, G.; Thabet, A.; and Ghanem, B. 2021a. Pu-gcn: Point cloud upsampling using graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11683–11692.
- Qian, Y.; Hou, J.; Kwong, S.; and He, Y. 2020. PUGeo-Net: A geometry-centric network for 3D point cloud upsampling. In *European conference on computer vision*, 752–769. Springer.
- Qian, Y.; Hou, J.; Kwong, S.; and He, Y. 2021b. Deep magnification-flexible upsampling over 3d point clouds. *IEEE Transactions on Image Processing*, 30: 8354–8367.
- Qiu, S.; et al. 2022. Pu-transformer: Point cloud upsampling transformer. In *Proceedings of the Asian Conference on Computer Vision*, 2475–2493.
- Uy, M. A.; Pham, Q.-H.; Hua, B.-S.; Nguyen, T.; and Yeung, S.-K. 2019. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1588–1597.
- Wu, S.; Huang, H.; Gong, M.; Zwicker, M.; and Cohen-Or, D. 2015. Deep points consolidation. *ACM Transactions on Graphics (ToG)*, 34(6): 1–13.
- Ye, S.; Chen, D.; Han, S.; Wan, Z.; and Liao, J. 2021. Meta-PU: An arbitrary-scale upsampling network for point cloud. *IEEE transactions on visualization and computer graphics*, 28(9): 3206–3218.
- Yifan, W.; Wu, S.; Huang, H.; Cohen-Or, D.; and Sorkine-Hornung, O. 2019. Patch-based progressive 3d point set upsampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5958–5967.
- Yu, L.; Li, X.; Fu, C.-W.; Cohen-Or, D.; and Heng, P.-A. 2018. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2790–2799.
- Zhang, J.; Zhao, X.; Chen, Z.; and Lu, Z. 2019. A review of deep learning-based semantic segmentation for point cloud. *IEEE access*, 7: 179118–179133.
- Zhang, W.; Xing, R.; Zeng, Y.; Liu, Y.-S.; Shi, K.; and Han, Z. 2023. Fast learning radiance fields by shooting much fewer rays. *IEEE Transactions on Image Processing*.
- Zhao, W.; Liu, X.; Zhai, D.; Jiang, J.; and Ji, X. 2023. Self-Supervised Arbitrary-Scale Implicit Point Clouds Upsampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Zhao, W.; Liu, X.; Zhong, Z.; Jiang, J.; Gao, W.; Li, G.; and Ji, X. 2022. Self-supervised arbitrary-scale point clouds upsampling via implicit neural representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1999–2007.
- Zhou, J.; Ma, B.; Li, S.; Liu, Y.-S.; and Han, Z. 2023. Learning a More Continuous Zero Level Set in Unsigned Distance Fields through Level Set Projection. In *Proceedings of the IEEE/CVF international conference on computer vision*.
- Zhou, J.; Ma, B.; Liu, Y.-S.; Fang, Y.; and Han, Z. 2022. Learning consistency-aware unsigned distance functions progressively from raw point clouds. *Advances in Neural Information Processing Systems*, 35: 16481–16494.