

000
001
002
003
004
005
006
007
008
009
010
011
012
013
014
015054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
100
101
102
103
104
105
106
107

GridPull: Towards Scalability in Learning Implicit Representations from 3D Point Clouds

Anonymous ICCV submission

Paper ID 3404

Abstract

Learning implicit representations has been a widely used solution for surface reconstruction from 3D point clouds. The latest methods infer a distance or occupancy field by overfitting a neural network on a single point cloud. However, these methods suffer from a slow inference due to the slow convergence of neural networks and the extensive calculation of distances to surface points, which limits them to small scale points. To resolve the scalability issue in surface reconstruction, we propose GridPull to improve the efficiency of learning implicit representations from large scale point clouds. Our novelty lies in the fast inference of a discrete distance field defined on grids without using any neural components. To remedy the lack of continuousness brought by neural networks, we introduce a loss function to encourage continuous distances and consistent gradients in the field during pulling queries onto the surface in grids near to the surface. We use uniform grids for a fast grid search to localize sampled queries, and organize surface points in a tree structure to speed up the calculation of distances to the surface. We do not rely on learning priors or normal supervision during optimization, and achieve superiority over the latest methods in terms of complexity and accuracy. We evaluate our method on shape and scene benchmarks, and report numerical and visual comparisons with the latest methods to justify our effectiveness and superiority. The source code will be publicly available.

1. Introduction

It is vital to reconstruct surfaces from 3D point clouds for downstream applications. A widely used strategy is to learn implicit representations [45, 47, 22, 9, 70, 59, 41, 53] from 3D point clouds in a data-driven manner. With the learned implicit representations, we can reconstruct surfaces in meshes by running the marching cubes algorithm [35]. By learning priors from large scale datasets during training, previous methods [43, 19, 25, 34, 60, 57, 15] gen-

eralize the learned priors to either global implicit functions [68, 62, 7, 26, 6, 38] or local ones for unseen point clouds. However, the generalization ability limits their performances on large structure or geometry variations of training samples.

More recent methods [20, 2, 78, 3, 73, 4, 37, 12, 50] achieved better generalization by directly overfitting neural networks on single unseen point clouds. They rely on extensive calculations of distances between queries and surface points to probe the space, which supervises neural networks to converge to a distance or occupancy field [37, 20, 2, 78, 3, 10]. However, this inference procedure is time consuming due to the extensive distance calculations and the slow convergence of neural networks. This demerit makes these methods hard to scale up to large scale point clouds for surface reconstruction.

To address the scalability challenge, we propose GridPull to speed up the learning of implicit function from large scale point clouds. GridPull does not require learned priors or point normal, and directly infers a distance field from a point cloud without using any neural components. We infer the distance field on grids near the surface, which reduces the number of grids we need to infer. Moreover, we organize surface points in a tree structure to speed up the nearest neighbor search for the calculation of distances to the surface. Specifically, we infer the discrete distance field by pulling queries onto the surface in grids of interests. Our loss function encourages continuous distances and consistent gradients in the field, which makes up the lack of continuousness brought by neural networks. We justify our effectiveness and highlight our superiority by numerical and visual comparisons with the latest methods on the widely used benchmarks. Our contributions are listed below.

- i) We propose GridPull to reconstruct surfaces from large scale point clouds without using neural networks. GridPull speeds up the learning of implicit function, which addresses the scalability challenge in surface reconstruction.
- ii) We introduce a loss function to directly infer a discrete

108 distance field defined on grids but achieving continuous distances and consistent gradients in the field. 162
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161

162 on grids to speed up the learning of a radiance field, while they cared more about the quality of synthesized views than the underlying geometry. Without using neural components, some methods [52, 50] directly infer discrete implicit functions on grids. To make up the continuousness of neural networks, they employed different constraints, such as Possion equations or Viscosity priors, to pursue a continuous field from a single point cloud. However, these methods are slowly converged, which makes them hard to scale up to large scale point clouds. Although we also use grids to speed up the inference, we introduce a more efficient training strategy to decrease optimization space and more effective losses to increase inference efficiency. 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215

2. Related Work

Neural implicit representations have made a huge progress in various tasks [45, 47, 22, 9, 70, 59, 41, 53]. We can use different supervision including 3D supervision [44, 49, 48, 42, 11], multi-view [56, 33, 27, 77, 32, 69, 46, 30, 72, 71, 17, 65, 76, 67, 63, 64], and point clouds [68, 31, 43, 19] to learn neural implicit representations. In the following, we focus on reviewing works on surface reconstruction by learning implicit representations point clouds below.

Learning with Priors. With neural networks, one intuitive strategy is to use neural networks to learn priors from a training set and then generalize the learned priors to unseen samples. We can learn either global priors to map a shape level point cloud into a global implicit function [43, 19, 25, 34, 60, 57, 15] or local priors for local implicit functions to represent parts or patches [68, 62, 7, 26, 6, 38] which are further used to approximate a global implicit function.

These methods require large scale datasets to learn priors. However, the learned priors may not generalize well to unseen point clouds that have large geometric variations compared to training samples. Our method does not require priors, and falls into the following category.

Learning with Overfitting. To improve the generalization, we can learn implicit functions by overfitting neural networks on single point clouds. Methods using this strategy introduce novel constraints [20, 2, 78, 3, 73, 4], ways of leveraging gradients [37, 12], differentiable poisson solver [50] or specially designed priors [38, 39] to learn signed [37, 20, 2, 78, 3, 10] or unsigned distance functions [12, 79].

These methods rely on neural networks to infer implicit functions without learning priors. However, the slow convergence of neural networks is a big limit for them to scale up to large scale point clouds. Our method addresses the scalability challenge by directly inferring a discrete distance field without using neural networks. Our novel loss leads to less complexity and higher accuracy than the overfitting based methods.

Learning with Grids. Learning implicit functions with grids has been used in prior-based methods [59, 41, 51, 61, 34, 27, 66]. Using neural networks, these methods learn features at vertices of grids and further map the interpolated features into a distance or occupancy field. Some methods [54, 58, 74, 29] inferred a discrete radiance field defined

on grids to speed up the learning of a radiance field, while they cared more about the quality of synthesized views than the underlying geometry. Without using neural components, some methods [52, 50] directly infer discrete implicit functions on grids. To make up the continuousness of neural networks, they employed different constraints, such as Possion equations or Viscosity priors, to pursue a continuous field from a single point cloud. However, these methods are slowly converged, which makes them hard to scale up to large scale point clouds. Although we also use grids to speed up the inference, we introduce a more efficient training strategy to decrease optimization space and more effective losses to increase inference efficiency.

3. Method

Overview. GridPull aims to achieve a fast inference of a distance field, such as a signed distance field, from a 3D point cloud $S = \{s_j | j \in [1, J]\}$ without using any neural components. As shown in Fig. 1 (b), we represent the distance field by a signed distance function (SDF) f defined on a set of discrete grids G with a resolution of R^3 , where the vertices $V = \{v_i | i \in [1, I], I = (R + 1)^3\}$ shared among these grids hold learnable signed distances $D = \{d_i\}$. For an arbitrary location q in Fig. 1 (e), such as a randomly sampled query or a surface point, we use trilinear interpolation to approximate the signed distance $f(q)$ from the eight nearest grid vertices below,

$$f(q) = triInter(q, D). \quad (1)$$

Our goal is to conduct direct optimization in Fig. 1 (c) to infer the signed distances D on grid vertices by minimizing our loss function L on queries q below,

$$\min_D L(f(q)). \quad (2)$$

With the learned function f , we run the marching cubes algorithm [35] to reconstruct a mesh as the surface as shown in Fig. 1 (d).

Discrete Grids. We split the space occupied by the point cloud S into discrete grids G with a resolution of R in Fig. 1 (b). GridPull can work with high resolutions, such as $R = 512$, because of our efficient and effective loss function and training strategy. Within each grid, we interpolate the signed distance $f(q)$ of query q using the trilinear interpolation in Eq. 1. We organize surface points in a tree structure to speed up the search of the nearest surface point for queries q . Since we need extensively conduct trilinear interpolation on signed distances for queries q , we use uniform grids for fast nearest grid search for queries. We did not use an Octree to organize grids, due to its inefficient and inaccurate grid search for queries.

Geometric Initialization. We initialize the signed distances d_i on grid vertices which represent a sphere in the

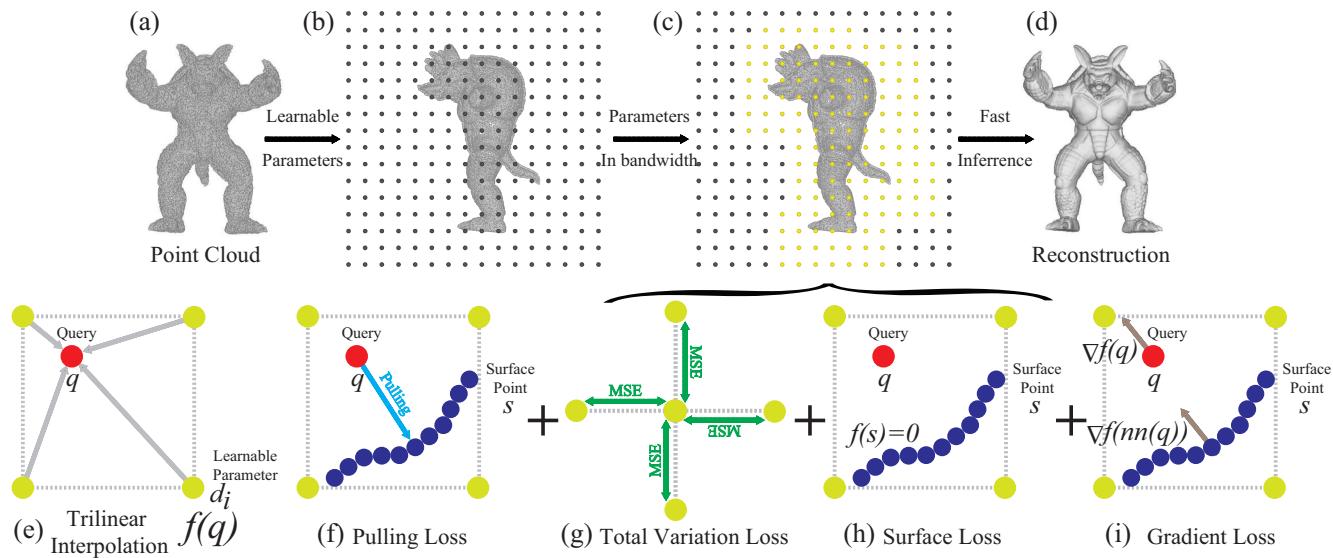


Figure 1. We proposed GridPull for fast inference of SDF defined on discrete grids using effective losses.

field. The initialization can speed up the convergence during optimization. Our preliminary results show that a small sphere with a radius of one or two grids works well. Since we can calculate the continuous distance loss over smaller regions if we enlarge a small sphere to the target shape than shrinking a large sphere.

Signed Distance Inference. For a query q , we infer its signed distance $f(q)$ through pulling it onto its nearest surface point in the point cloud S . As shown in Fig. 1 (e), we first obtain $f(q)$ as the interpolation of the signed distances d_i on the nearest eight vertices of q . We implement the trilinear interpolation by solving $\mathbf{a} = [a_0, \dots, a_7]$ in a linear regression system on the eight vertices v_i below,

$$f(q) = a_0 + a_1x + a_2y + a_3z + a_4xy + a_5xz + a_6yz + a_7xyz, \quad (3)$$

where $[x, y, z]$ is the coordinate of q and the eight constraints are $\{f(v_i) = d_i\}$.

With the predicted signed distance $f(q)$ and the gradient $\nabla f(q) = \partial f / \partial q$, we pull a query q onto the surface and obtain a pulled query p by $p = q - f(q) \times \nabla f(q) / \| \nabla f(q) \|_2$ in Fig. 1 (f).

Our goal is to minimize the distance between the pulled query p and the nearest surface point $NN(q) \in S$ of query q below,

$$L_{Pull} = \| p - NN(q) \|_2. \quad (4)$$

We adopt this pulling loss from [36]. Instead, we do not rely on neural network in the pulling procedure. We use the trilinear interpolation on signed distances d_i on grid vertices to obtain signed distances and gradients according to Eq. 3.

We sample queries q around surface points in each epoch. For each randomly sampled surface point, we sample queries using a gaussian distribution with the surface point as the center and a two-grid length variance. We build a KD-tree on surface points so that we can speed up the search-

ing of the nearest neighbor from a large number of surface points for each query.

To speed up the inference, we only calculate the pulling loss on grids near the surface. We regard the grids containing surface points as the base and find the union of their M_1 nearest neighboring grids. These grids form a M_1 -bandwidth near the surface in Fig. 1 (c).

Continuous Distance Fields. Although the pulling loss can infer correct signed distances at grid vertices, it fails to learn continuous signed distances across the field, especially near the surface. Fig. 2 (a) shows severe artifacts on the surface and gradients near the surface are messy and inconsistent, which are caused by the discontinuous signed distances across neighboring regions.

Neural network based methods [36, 40] do not have this issue, since the continuous character of neural networks makes the sudden change of signs on the same shape side hard. While our method uses signed distances d_i on grid vertices to represent a discrete field, this makes that one d_i can change separately without considering the change of neighboring d_i , which leads to a poor continuousness in the field.

To resolve this issue, we impose a total variation (TV) loss as a continuous constraint on our signed distances d_i on grid vertices. Our key idea is to make the change of one d_i affect the change of its neighboring $NN(d_i)$, which smoothes the local distance field and propagates local updates to the rest grids. Hence, as shown in a 2D case in Fig. 1 (g), we approximate one d_i to each one of its six neighbors by minimizing the difference between each two below,

$$L_{TV} = \sqrt{\sum_{a=\{x,y,z\}} e_{a+1}(d_i)^2 + e_{a-1}(d_i)^2}, \quad (5)$$

where $e_{a+1}(d_i)$ is the difference between d_i and the signed

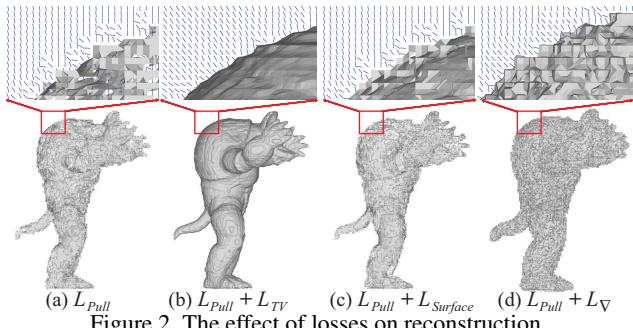


Figure 2. The effect of losses on reconstruction.

distance along the a axis, i.e., x, y, z axis. Fig. 2 (b) shows that the continuous constraints can significantly improve the signed distance field in terms of continuousness and gradients.

One remaining issue is its computational complexity. Since we are targeting grids at a high resolution, such as $R = 512$, it is very time consuming to impose this constraint on all d_i across the field.

To resolve this issue, we focus on grids near the surface. Similar to imposing the pulling loss on grids in the M_1 -bandwidth near the surface, we impose our continuous constraints on the grids in the M_2 -bandwidth near the surface, which significantly reduces the number of learnable parameters to optimize.

Signed Distance Supervision. For signed distances, one supervision we can directly use is the input point cloud S . It is a discrete surface indicating the zero level-set of the signed distance function. Hence, as introduced in Fig. 1 (h), we can constrain the signed distances at the points s on the surface below,

$$L_{Surface} = \|f(s)\|_2. \quad (6)$$

Although Fig. 2 (c) indicates that $L_{Surface}$ slightly improves the signed distance field and leads to a little bit more continuous surface, it is helpful to determine the zero level set with other losses. For noises on S , our continuous constraint can relieve the impact of noises on this supervision. We will show this in experiments.

Consistent Gradients. Fig. 2 (a) indicates that the pulling loss produces gradients that are messy and non-orthogonal to the surface. This leads to inaccurate signed distance inference during the pulling procedure.

We additionally introduce a constraint on gradients to improve the field near the surface. As shown in Fig. 1 (i), we aim to encourage the gradient at a query q to point to the same direction of the gradient at its nearest surface point $NN(q)$. We use a cosine distance below to achieve more consistent gradients,

$$L_\nabla = 1 - \cos(\nabla f(q), \nabla f(NN(q))). \quad (7)$$

L_∇ constrains the gradients in the field and corrects inaccurate gradients. Fig. 2 (d) shows that more consistent gradients can improve the surface to be more continuous us-

ing small planes on the surface which however the surface not smooth at all.

Loss Function. We learn signed distances on grid vertices D by minimizing the loss function below,

$$L = L_{Pull} + \alpha L_{TV} + \beta L_{Surface} + \gamma L_\nabla. \quad (8)$$

where α , β , and γ are balance weights to make each term contribute equally. Fig. 3 shows the signed distance field on a slide of D during optimization. Our inference starts from an initial sphere and only optimizes variables in the bandwidth. We skip the grids that are not optimized after the initialization during the marching cubes for surface reconstruction. Please watch our video for more details.

4. Experiments and Analysis

We evaluate GridPull by numerical and visual comparisons with the latest methods on synthetic and real datasets in surface reconstruction.

Datasets and Metrics. We use benchmarks for shapes and scenes in evaluations. For shapes, we conduct evaluations on five datasets including a subset of ShapeNet [1], FAMOUS [15], Thingi10k [80], Surface Reconstruction Benchmark (SRB) [68] and D-FAUST [5]. For scenes, we also report comparisons on five datasets including 3D-Scene [81], SceneNet [23], 3DFRONT [16], Matterport [8] and KITTI [18].

We use L2 Chamfer distance (CD_{L2}), CD_{L1} and Hausdorff distance(HD) to measure the error between the reconstructed surface and the ground truth. Moreover, we use normal consistency (NC) and F-score to evaluate the accuracy of normal on the reconstructed surface. We also report our time and storage complexity to highlight our advantage towards scalability. We also use the intersection over union (IoU) to measure the reconstruction for fair comparison with the latest methods.

Details. We use $R = 256$ to infer a distance field. We calculate the pulling loss on grids in a bandwidth with $M_1 = 3$. We impose the continuous constraint on grids in a bandwidth with $M_2 = 14$ for shapes and $M_2 = 25$ for scenes, since scenes contain open surfaces with larger empty spaces. We run the marching cubes for surface reconstruction at a resolution of 256.

We use the Adam optimizer with an initial learning rate of 1.0. We decrease the learning rate with a decay rate of 0.3 every 400 iterations. We use 50,000 queries in each iteration, and run 1600 iterations for each point cloud during optimization. We set the balance weights $\alpha = 1$, $\beta = 1$, and $\gamma = 0.005$ for equal contribution from the three terms.

4.1. Surface Reconstruction for Shapes

ShapeNet. Following predictive context prior (PCP) [40] and NeuralPull (NP) [36], we report our results on a subset

324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431

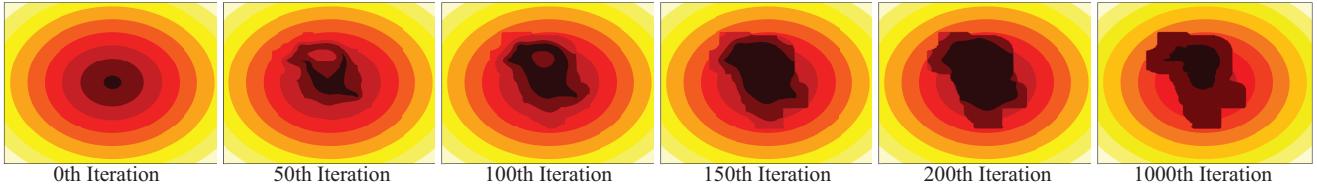


Figure 3. Visualization of signed distances in optimization.

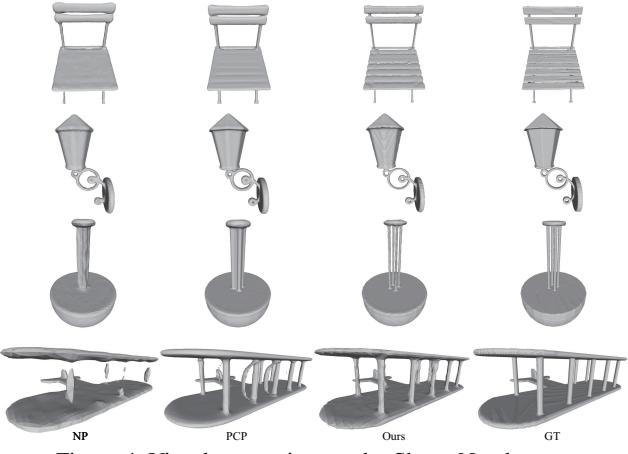


Figure 4. Visual comparison under Shapenet dataset.



Figure 5. Visual comparison under FAMOUS dataset.

in ShapeNet. Tab. 1 shows our superiority over PCP and NP in numerical comparisons over all classes, even PCP employs a local shape prior learned from a large scale dataset. The visual comparison in Fig. 4 shows that we can reveal more accurate and detailed geometry.

Method	$CD_{L2} \times 100$	NC	F-score(μ)	F-score(2μ)
NP [36]	0.038	0.939	0.961	0.976
PCP [40]	0.0136	0.9590	0.9871	0.9899
Ours	0.0086	0.9723	0.9896	0.9923

Table 1. Reconstruction accuracy under ShapeNet in terms of $CD_{L2} \times 100$, NC, F-score with $\mu = 0.002$. More numerical results in each class can be found in our supplementary materials.

FAMOUS. We follow the experimental setting in PCP [40] and NP [36] to evaluate GridPull on FAMOUS. We compare GridPull with the latest methods with priors including PCP, GenSDF, FGC and no priors including NP and IGR. GridPull outperforms these methods in terms of CD_{L2} and running time. Although FGC is fast, it does not reconstruct

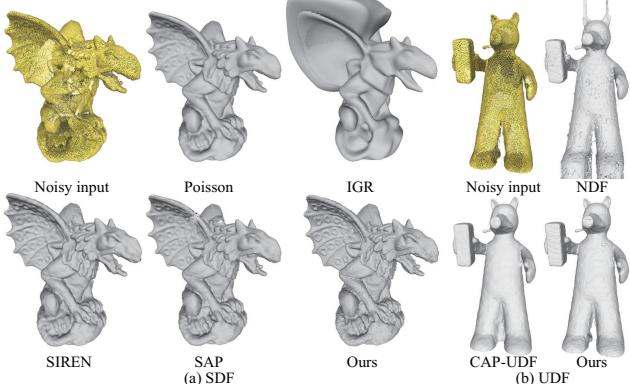


Figure 6. Visual comparison under SRB dataset.

a watertight mesh but just a polygon soup with no normal, which contains almost no geometry details. What was worse, it fails to reconstruct some point clouds in more than eight hours, hence we remove these shapes from its results. We highlight our superiority in visual comparison in Fig. 5. We can reveal details like hair and expressions more clearly.

Method	$CD_{L2} \times 100$	Min time	Mean time
IGR [21]	1.650	402s	407s
GenSDF [13]	0.668	230s	232s
NP [36]	0.220	593s	605s
FGC [75]	0.055	10s	Timeout
PCP [40]	0.044	3416s	3534s
Ours	0.040	198s	201s

Table 2. Reconstruction accuracy under FAMOUS.

SRB. We follow the experimental setting in VisCo [52] to evaluate GridPull on real scans in SRB. We report the comparison with the latest overfitting based methods in terms of CD_{L1} in Tab. 3. Meanwhile, we also report their one-sided distances between the reconstructed mesh and the input noisy point cloud. The numerical comparison shows that our method significantly outperforms the latest. Moreover, we also report our results of learning unsigned distances in Tab. 4. We replace our pulling loss with the pulling loss introduced for unsigned distances in CAP-UDF [79], and remove the gradient loss due to the nondifferentiable character of UDF on surface. Visual comparison in Fig. 6 shows that our method can reconstruct surfaces that are more compact than the methods learning SDF like SIREN or more smooth than the methods learning UDF like CAP-UDF.

	Poisson [28]	IGR [21]	SIREN [55]	VisCo [52]	SAP [50]	Ours
CD_{L1}	0.354	0.184	0.204	0.182	0.076	0.066
HD	7.328	3.126	4.026	2.948	1.378	1.051
CD_{L1}^{\rightarrow}	0.354	0.084	0.090	0.112	0.050	0.043
HD^{\rightarrow}	7.328	1.462	1.726	1.438	0.612	0.472
Time	7s	1848s	779s	1800s	326s	199s

Table 3. Reconstruction accuracy under SRB. More numerical results in each scene can be found in our supplementary materials.

	NDF [12]	CAP-UDF [79]	Ours(UDF)
CD_{L1}	0.238	0.073	0.070
F-Score	68.6	84.5	85.1

Table 4. Accuracy of reconstruction with UDF under SRB.

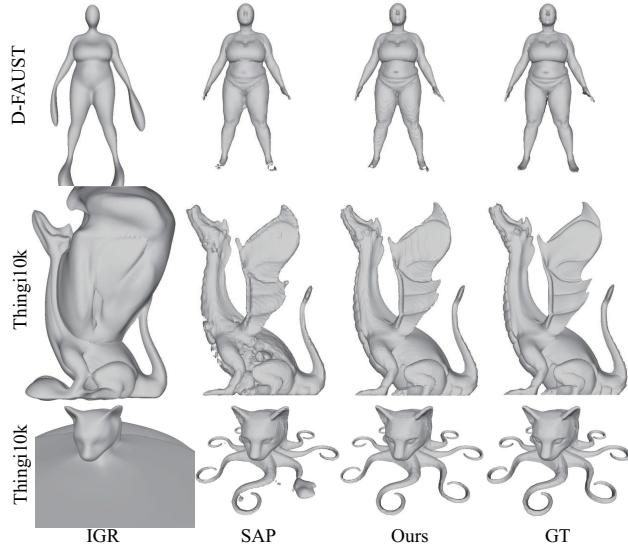


Figure 7. Visual comparison under Thing10k and D-FAUST.

Thing10K. We follow the experimental setting in SAP [50] to evaluate GridPull in Thing10K. Tab. 5 indicates our superiority over the overfitting based methods in the numerical comparison. Visual comparisons in Fig. 7 show our more accurate surfaces with complex details.

		IGR [21]	SAP [50]	Ours
Thing10k	CD_{L1}	0.440	0.054	0.051
	F-Score	0.505	0.940	0.948
	NC	0.692	0.947	0.965
DFAUST	CD_{L1}	0.235	0.043	0.015
	F-Score	0.805	0.966	0.975
	NC	0.911	0.959	0.978

Table 5. Reconstruction accuracy under Thing10k and D-FAUST in terms of CD_{L1} and F-Score with a threshold of 0.01.

D-FAUST. We also follow the experimental setting in S-AP [50] to evaluate GridPull in D-FAUST. Tab. 5 shows that we outperform the overfitting based methods with more surface details. As shown in Fig. 7, we recover more accurate human bodies and more detailed expressions on faces.

4.2. Surface Reconstruction for Scenes

3DScape. We follow PCP [40] to report CD_{L1} , CD_{L2} and Normal Consistency(NC) for evaluation. We report the comparisons with the latest methods in Tab. 6. We outperform both kinds of methods with learned priors such as ConvOcc [57] and PCP [40] and overfitting based NP [36] in all scenes. The visual comparisons 8 shows that our method can work well with real scans on scenes and reveal more geometry details on surfaces.

	ConvOcc [57]	NP [36]	PCP [40]	Ours
$CD_{L2} \times 100$	13.328	0.4466	0.1130	0.1042
CD_{L1}	0.0492	0.0194	0.0071	0.0064
NC	0.8720	0.8514	0.8864	0.9088

Table 6. Accuracy comparison under 3DScape. More numerical results can be found in our supplementary materials.

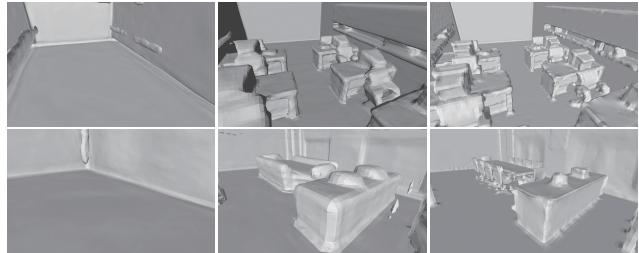


Figure 9. Visual comparison under SceneNet dataset.

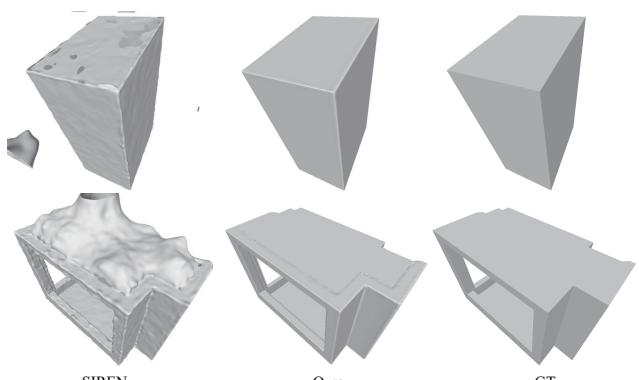


Figure 10. Visual comparison under 3Dfront dataset.

SceneNet. Following the experimental setting in PCP [40], we report our CD_{L1} , NC, and F-score in Tab. 7. The comparison indicates that we outperform these methods by producing more accurate and smooth surfaces. This is also verified by our visual comparison in Fig. 9, where the latest methods reveal either no or inaccurate indoor geometry.

	NP [36]	PCP [40]	Ours
CD_{L1}	0.054	0.017	0.0156
NC	0.895	0.938	0.9416
F-Score	0.818	0.965	0.9674

Table 7. Reconstruction accuracy under SceneNet. F-Score is with a threshold of 0.025. More numerical results in each scene can be found in our supplementary materials.

3D-FRONT. We follow the experimental setting in NeuralPoisson [14] to report CD_{L1} , IoU and L2 distance over the voxelization of meshes. IoU and L2 distances are calculated as the error between the reconstructed surfaces and the ground truth over the voxelization at a resolution of 128. The comparison in Tab. 8 shows that our results are more accurate than the latest method. Our visual comparison with SIREN in Fig. 10 shows that GridPull can infer an accurate zero level set which leads to very sharp corners.

Method	CD	IOU	L2
SIREN [55]	0.0183	0.524	2.373
NeuralPoisson [14]	0.0161	0.545	2.073
Ours	0.0141	0.562	1.984

Table 8. Reconstruction accuracy under 3DFRONT.

Matterport. We follow the experimental setting in NGS [24], and report the comparison in terms of CD_{L1} , NC, and F-score in Tab. 9. We achieve the best performance in terms of accuracy. NGS [24] has a pretty fast inference but it costs about four to five days to learn priors from the training set in Matterport. Visual comparison in Fig. 11 shows that we can reconstruct more accurate sur-

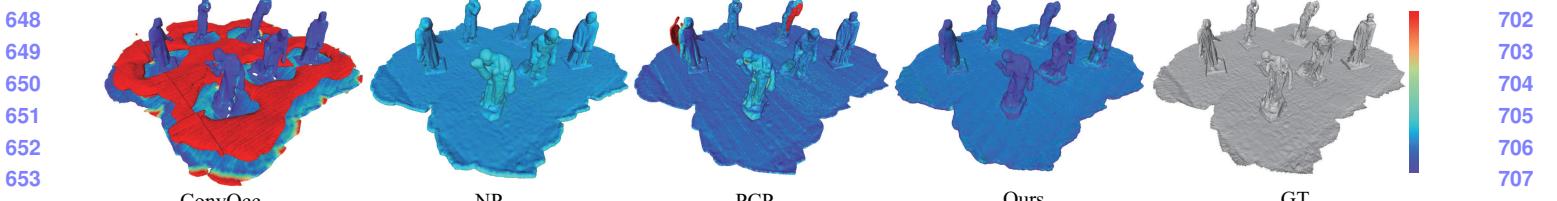


Figure 8. Visual comparison under 3DScene dataset.

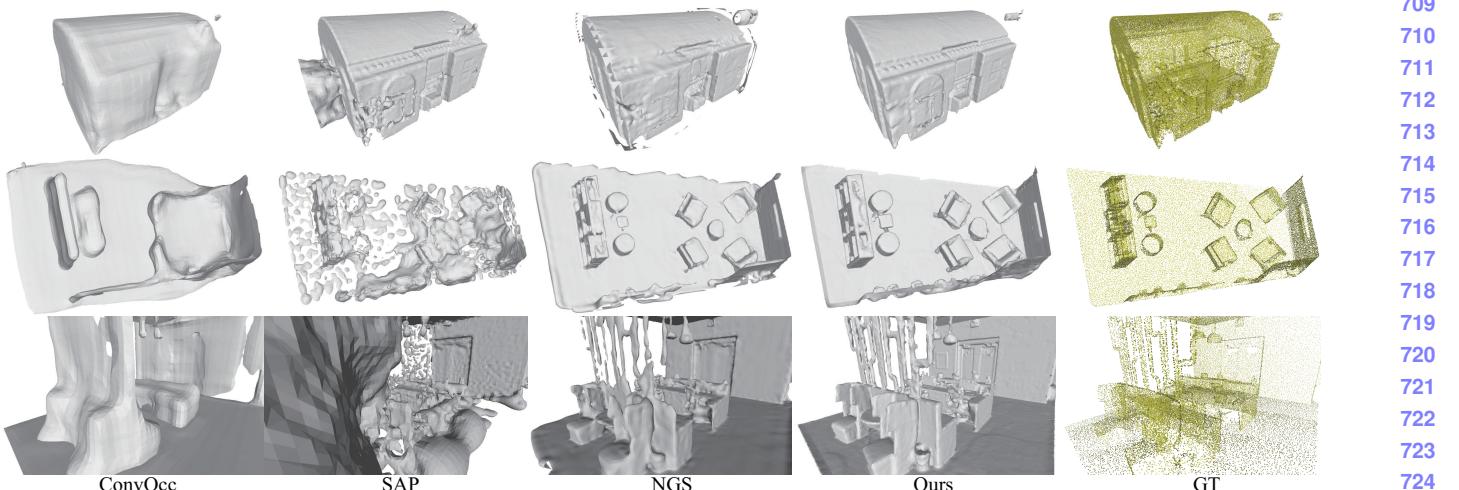


Figure 11. Visual comparison under Matterport3D dataset.

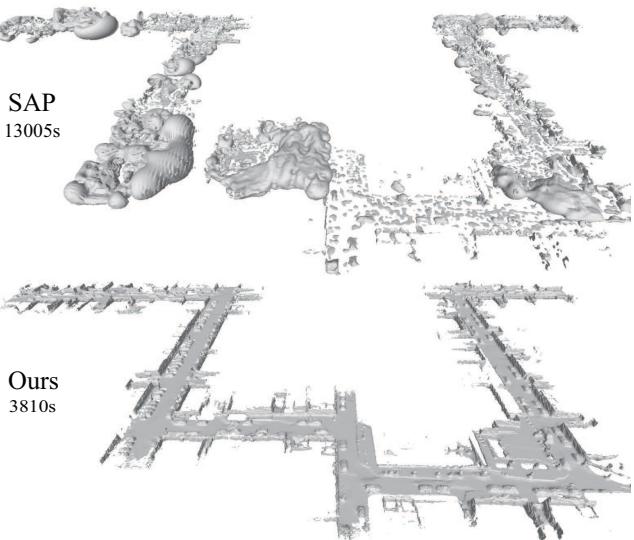


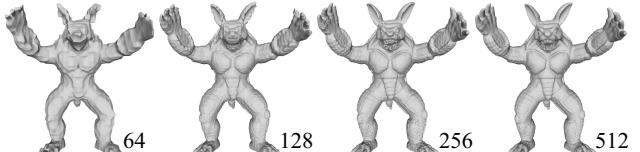
Figure 12. Visual comparison under Kitti dataset.
 faces while NGS is poorly generalized to unseen scenes.
KITTI. We evaluate GridPull on a large scale real scan from KITTI odometry (Sequence00, frame 3000 to 4000) which contains about 13.8 million points. The comparison in Fig. 12 shows that we can reconstruct a more accurate surface in much less time (3810s) than the overfitting based method SAP (13005s) [50].

4.3. Ablation Studies

We conduct ablation studies on FAMOUS [15] to justify the effectiveness of the modules in GridPull.

Method	$CD_{L1} \times 1000$	F-Score(0.002)	NC	Time
ConvOcc [57]	6.21	88.9	92.0	34s
SAP [50]	4.17	91.2	91.4	643s
NGS [24]	3.04	97.3	95.7	2.1s
Ours	2.93	97.7	96.3	223s

Table 9. Reconstruction accuracy under Matterport3D.

Figure 13. Effect of Resolution R .

Loss. We justify the effectiveness of our loss in terms of CD_{L2} in Tab. 10. For the pulling loss, it is able to estimate a coarse shape but with discontinuous meshes and artifacts. The gradient loss can improve the continuousness on the surface. Using the TV loss, we obtained a more continuous field, which removes most of artifacts and significantly improve the accuracy. The surface loss encourages a more accurate zero level-set, which also improves the accuracy.

Loss	$CD_{L2} \times 100$
L_{Pull}	0.184
$L_{Pull} + L_\nabla$	0.126
$L_{Pull} + L_\nabla + L_{TV}$	0.056
$L_{Pull} + L_\nabla + L_{TV} + L_{Surface}$	0.040

Table 10. Effect of losses.

Resolution R . We compare the effect of resolution R on the reconstructed surfaces. We try several candidates $\{32, 64, 128, 256, 512\}$ to learn the discrete distance field. We use the same bandwidth for the calculation of the pulling

756 loss and the TV loss, and reconstruct meshes by running the
 757 marching cubes at the same resolution. The numerical and
 758 visual comparisons show that higher resolutions can infer
 759 more geometry details but also cost more inference time.
 760

	32	64	128	256	512
$CD_{L2} \times 100$	0.186	0.097	0.048	0.040	0.035
Time	50s	67s	84s	201s	645s

Table 11. Effect of Resolution R .

Point Number. We compare our performance on different numbers of points. We use the same set of shapes but with different numbers of points including $\{10K, 100K, 1000K\}$. The numerical comparison in Tab. 12 shows that our method can estimate a discrete distance field from different numbers of points well, which leads to more accurate reconstructed surfaces than the latest overfitting based methods. In addition, our time complexity is not affected by the number of points a lot, while the latest methods require more time to converge on more points.

Number	Method	$CD_{L2} \times 100$	Time	Memory
10K	NP	0.351	548s	4.5G
	SAP	0.275	147s	3.5G
	Ours	0.122	82s	3.3G
100K	NP	0.224	615s	4.6G
	SAP	0.076	354s	3.6G
	Ours	0.040	84s	3.3G
1000K	NP	0.175	1304s	4.8G
	SAP	0.063	841s	3.7G
	Ours	0.039	90s	3.4G

Table 12. Effect of point numbers.

Initialization. We highlight our geometric initialization of distances on grid vertices by comparing it with random initialization. The comparison in Tab. 13 shows that random initialization makes the optimization converge hard, and cannot work well with the TV loss, especially with a bandwidth, which produces lots of artifacts in empty spaces. Moreover, we try an Eikonal constraint on gradients based on the geometric initialization, but did not get improvement.

	Random	Geometric	$\ \nabla\ = 1$
$CD_{L2} \times 100$	0.155	0.040	0.054

Table 13. Effect of Initialization.

Bandwidth. We report the effect of bandwidth M_1 and M_2 in the pulling loss and the TV loss in Tab. 14. For both losses, the bandwidth controls the scope of query sampling and continuous constraints. A larger bandwidth would cover more parameters which require more time and make the optimization converge slowly. For instance, if we sample queries and impose the pulling in all R^3 grids, the speed is even slower than NeuralPull which uses neural network for the pulling, since neural networks can generalize signed distances nearby and do not need to infer in all grids. Although a small bandwidth would save time, a too small bandwidth would cover too few grids around the surface, which leads to discontinuousness and artifacts. While if the bandwidth is large enough, we may not see surface improvement but just cost more time. Moreover, even with the same loss,

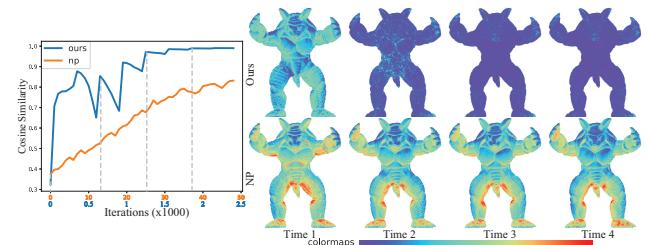


Figure 14. Gradients comparison with neural networks.

NeuralPull does not produce better results than ours. Comparing with methods using neural networks, our discrete SDF can have faster convergence, as indicated by the angular error of gradients in Fig. 14.

Bandwidth	$CD_{L2} \times 100$	Time
$L_{Pull}(M_1 = 3)$	0.040	84s
$L_{Pull}(\text{All Grids})$	0.043	628s
$L_{TV}(M_2 = 2)$	0.079	83s
$L_{TV}(M_2 = 14)$	0.040	84s
$L_{TV}(M_2 = 30)$	0.041	89s
$L_{TV}(M_2 = 50)$	0.040	96s
$L_{TV}(\text{All Grids})$	0.040	101s
NeuralPull	0.220	552s
$\text{NeuralPull} + L_{\nabla} + L_{Surface}$	0.203	554s

Table 14. Effect of Bandwidth.

Noises. We report our performance on point clouds with noise. We follow the experimental setting in PCP, and report our results with middle and large level noise in Tab. 15. To handle noise, we weight more on the continuous constraint L_{TV} using a larger $\alpha = 2$ to decrease the impact of noise. The comparison shows that our method can handle noise better than the neural network based methods. Since the continuousness of neural networks make gradients near the surface over smooth to overfit all noise, this makes it hard to infer accurate zero level-set.

Noise level	NP [36]	PCP [40]	Ours
F-med-noise	0.280	0.071	0.044
F-max-noise	0.310	0.298	0.060

Table 15. Effect of noises.

5. Conclusion

We introduce GridPull for the fast inference of distance fields from large scale point clouds without using neural components. We infer a distance field using learnable parameters defined on discrete grids, which we can directly optimize in a more efficient way than neural networks. Besides the complexity advantage, our loss function manages to achieve more continuous distance fields with more consistent gradients, which leads to higher accuracy in surface reconstruction. We evaluate GridPull and justify its effectiveness on benchmarks. Numerical and visual comparisons show that GridPull outperforms the latest methods in terms of accuracy and complexity, even without using priors, normal, and neural networks.

864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917

References

- [1] ShapeNet. <https://www.shapenet.org/>. 4
- [2] Matan Atzmon and Yaron Lipman. SAL: Sign agnostic learning of shapes from raw data. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2
- [3] Matan Atzmon and yaron Lipman. SALD: sign agnostic learning with derivatives. In *International Conference on Learning Representations*, 2021. 1, 2
- [4] Yizhak Ben-Shabat, Chamin Hewa Koneputugodage, and Stephen Gould. DiGS : Divergence guided shape implicit neural representation for unoriented point clouds. *CoRR*, abs/2106.10811, 2021. 1, 2
- [5] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *IEEE Computer Vision and Pattern Recognition*, 2017. 4
- [6] Alexandre Boulch and Renaud Marlet. POCO: Point convolution for surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6302–6314, 2022. 1, 2
- [7] Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard A. Newcombe. Deep local shapes: Learning local SDF priors for detailed 3D reconstruction. In *European Conference on Computer Vision*, volume 12374, pages 608–625, 2020. 1, 2
- [8] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgbd data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 4
- [9] Chao Chen, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Unsupervised learning of fine structure generation for 3D point clouds by 2D projections matching. In *IEEE International Conference on Computer Vision*, 2021. 1, 2
- [10] Chao Chen, Yu-Shen Liu, and Zhizhong Han. Latent partition implicit with surface codes for 3d representation. In *European Conference on Computer Vision*, 2022. 1, 2
- [11] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2
- [12] Julian Chibane, Aymen Mir, and Gerard Pons-Moll. Neural unsigned distance fields for implicit function learning. *arXiv*, 2010.13938, 2020. 1, 2, 6
- [13] Gene Chou, Ilya Chugunov, and Felix Heide. GenSDF: Two-stage learning of generalizable signed distance functions. In *Proc. of Neural Information Processing Systems (NeurIPS)*, 2022. 5
- [14] Angela Dai and Matthias Nießner. Neural Poisson: Indicator functions for neural fields. *arXiv preprint arXiv:2211.14249*, 2022. 6
- [15] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J. Mitra, and Michael Wimmer. Points2Surf: Learning implicit surfaces from point clouds. In *European Conference on Computer Vision*, 2020. 1, 2, 4, 7
- [16] Huan Fu, Bowen Cai, Lin Gao, Lingxiao Zhang, Cao Li, Zengqi Xun, Chengyue Sun, Yiyun Fei, Yu Zheng, Ying Li, Yi Liu, Peng Liu, Lin Ma, Le Weng, Xiaohang Hu, Xin Ma, Qian Qian, Rongfei Jia, Binqiang Zhao, and Hao Zhang. 3d-front: 3d furnished rooms with layouts and semantics. *CoRR*, abs/2011.09127, 2020. 4
- [17] Qiancheng Fu, Qingshan Xu, Yew-Soon Ong, and Wenbing Tao. Geo-Neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. 2022. 2
- [18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition*, 2012. 4
- [19] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T. Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *International Conference on Computer Vision*, 2019. 1, 2
- [20] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3789–3799, 2020. 1, 2
- [21] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv*, 2002.10099, 2020. 5, 6
- [22] Zhizhong Han, Chao Chen, Yu-Shen Liu, and Matthias Zwicker. DRWR: A differentiable renderer without rendering for unsupervised 3D structure learning from silhouette images. In *International Conference on Machine Learning*, 2020. 1, 2
- [23] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla. Understanding realworld indoor scenes with synthetic data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4077–4085, 2016. 4
- [24] Jiahui Huang, Hao-Xiang Chen, and Shi-Min Hu. A neural galerkin solver for accurate surface reconstruction. *ACM Trans. Graph.*, 41(6), 2022. 6, 7
- [25] Meng Jia and Matthew Kyan. Learning occupancy function from point clouds for surface reconstruction. *arXiv*, 2010.11378, 2020. 1, 2
- [26] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3D scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2
- [27] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. SDFDiff: Differentiable rendering of signed distance fields for 3D shape optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2
- [28] Michael M. Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions Graphics*, 32(3):29:1–29:13, 2013. 5
- [29] Hai Li, Xingrui Yang, Hongjia Zhai, Yuqian Liu, Hujun Bao, and Guofeng Zhang. Vox-Surf: Voxel-based implicit surface representation. *CoRR*, abs/2208.10925, 2022. 2
- [30] Chen-Hsuan Lin, Chaoyang Wang, and Simon Lucey. SDF-SRN: Learning signed distance 3D object reconstruction from static images. In *Advances in Neural Information Processing Systems*, 2020. 2

- 972 [31] Minghua Liu, Xiaoshuai Zhang, and Hao Su. Meshing point
973 clouds with predicted intrinsic-extrinsic ratio guidance. In
974 *European Conference on Computer vision*, 2020. 2
975 [32] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li.
976 Learning to infer implicit surfaces without 3D supervision.
977 In *Advances in Neural Information Processing Systems*,
978 2019. 2
979 [33] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc
980 Pollefeys, and Zhaopeng Cui. DIST: Rendering deep implicit
981 signed distance function with differentiable sphere tracing.
982 In *IEEE Conference on Computer Vision and Pattern Recog-
983 nition*, 2020. 2
984 [34] Shi-Lin Liu, Hao-Xiang Guo, Hao Pan, Pengshuai Wang, X-
985 in Tong, and Yang Liu. Deep implicit moving least-squares
986 functions for 3D reconstruction. In *IEEE Conference on
987 Computer Vision and Pattern Recognition*, 2021. 1, 2
988 [35] William E. Lorensen and Harvey E. Cline. Marching cubes:
989 A high resolution 3D surface construction algorithm. *Com-
990 puter Graphics*, 21(4):163–169, 1987. 1, 2
991 [36] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias
992 Zwicker. Neural-Pull: Learning signed distance functions
993 from point clouds by learning to pull space onto surfaces.
994 *CoRR*, abs/2011.13495, 2020. 3, 4, 5, 6, 8
995 [37] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias
996 Zwicker. Neural-pull: Learning signed distance functions
997 from point clouds by learning to pull space onto surfaces. In
998 *International Conference on Machine Learning*, 2021. 1, 2
999 [38] Baorui Ma, Yu-Shen Liu, and Zhizhong Han. Reconstruct-
1000 ing surfaces for sparse point clouds with on-surface priors.
1001 In *IEEE Conference on Computer Vision and Pattern Recog-
1002 nition*, pages 6305–6315, 2022. 1, 2
1003 [39] Baorui Ma, Yu-Shen Liu, Matthias Zwicker, and Zhizhong
1004 Han. Surface reconstruction from point clouds by learning
1005 predictive context priors. In *IEEE Conference on Computer
1006 Vision and Pattern Recognition*, pages 6316–6327, 2022. 2
1007 [40] Baorui Ma, Yu-Shen Liu, Matthias Zwicker, and Zhizhong
1008 Han. Surface reconstruction from point clouds by learning
1009 predictive context priors. In *IEEE Conference on Computer
1010 Vision and Pattern Recognition*, 2022. 3, 4, 5, 6, 8
1011 [41] Julien N. P. Martel, David B. Lindell, Connor Z. Lin, Eric R.
1012 Chan, Marco Monteiro, and Gordon Wetzstein. ACORN:
1013 adaptive coordinate networks for neural scene representa-
1014 tion. *CoRR*, abs/2105.02788, 2021. 1, 2
1015 [42] Lars Mescheder, Michael Oechsle, Michael Niemeyer,
1016 Sebastian Nowozin, and Andreas Geiger. Occupancy networks:
1017 Learning 3D reconstruction in function space. In *IEEE Con-
1018 ference on Computer Vision and Pattern Recognition*, 2019.
1019 2
1020 [43] Zhenxing Mi, Yiming Luo, and Wenbing Tao. SSRNet: Scal-
1021 able 3D surface reconstruction network. In *IEEE Conference
1022 on Computer Vision and Pattern Recognition*, 2020. 1, 2
1023 [44] Mateusz Michalkiewicz, Jhony K. Pontes, Dominic Jack,
1024 Mahsa Baktashmotagh, and Anders P. Eriksson. Deep level
1025 sets: Implicit surface representations for 3D shape inference.
CoRR, abs/1901.06802, 2019. 2
[45] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik,
Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF:
Representing scenes as neural radiance fields for view syn-
thesis. In *European Conference on Computer Vision*, 2020.
1, 2
[46] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and
Andreas Geiger. Differentiable volumetric rendering: Learn-
ing implicit 3D representations without 3D supervision. In
*IEEE Conference on Computer Vision and Pattern Recog-
nition*, 2020. 2
[47] Michael Oechsle, Songyou Peng, and Andreas Geiger. U-
NISURF: Unifying neural implicit surfaces and radiance
fields for multi-view reconstruction. In *International Con-
ference on Computer Vision*, 2021. 1, 2
[48] Amine Ouasfi and Adnane Boukhayma. Few ‘zero level set’-
shot learning of shape signed distance functions in feature
space. In *European Conference on Computer Vision*, 2022.
2
[49] Jeong Joon Park, Peter Florence, Julian Straub, Richard
Newcombe, and Steven Lovegrove. DeepSDF: Learning
continuous signed distance functions for shape representa-
tion. In *IEEE Conference on Computer Vision and Pattern
Recognition*, 2019. 2
[50] Songyou Peng, Chiyu “Max” Jiang, Yiyi Liao, Michael
Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as
points: A differentiable poisson solver. In *Advances in Neu-
ral Information Processing Systems*, 2021. 1, 2, 5, 6, 7
[51] Songyou Peng, Michael Niemeyer, Lars M. Mescheder, Marc
Pollefeys, and Andreas Geiger. Convolutional occupancy
networks. In *European Conference on Computer Vision*, vol-
ume 12348, pages 523–540, 2020. 2
[52] Albert Pumarola, Artyom Sanakoyev, Lior Yariv, Ali Tha-
bet, and Yaron Lipman. Visco grids: Surface reconstruction
with viscosity and coarea grids. In *Advances in Neural In-
formation Processing Systems*, 2022. 2, 5
[53] Konstantinos Rematas, Ricardo Martin-Brualla, and Vittorio
Ferrari. Sharf: Shape-conditioned radiance fields from a sin-
gle view. In *International Conference on Machine Learning*,
2021. 1, 2
[54] Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinrong
Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels:
Radiance fields without neural networks. In *IEEE Con-
ference on Computer Vision and Pattern Recognition*, 2022. 2
[55] Vincent Sitzmann, Julien N.P. Martel, Alexander W.
Bergman, David B. Lindell, and Gordon Wetzstein. Implicit
neural representations with periodic activation functions. In
Advances in Neural Information Processing Systems, 2020.
5, 6
[56] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wet-
zstein. Scene representation networks: Continuous 3D-
structure-aware neural scene representations. In *Advances
in Neural Information Processing Systems*, 2019. 2
[57] Lars Mescheder, Marc Pollefeys, Andreas Geiger, Songy-
ou Peng, Michael Niemeyer. Convolutional occupancy net-
works. In *European Conference on Computer Vision*, 2020.
1, 2, 6, 7
[58] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel
grid optimization: Super-fast convergence for radiance fields
reconstruction. In *IEEE Conference on Computer Vision and
Pattern Recognition*, 2022. 2

- 1080 [59] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3D shapes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2
- 1081 [60] Jiapeng Tang, Jiabao Lei, Dan Xu, Feiying Ma, Kui Jia, and Lei Zhang. SA-ConvONet: Sign-agnostic optimization of convolutional occupancy networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 1, 2
- 1082 [61] Jia-Heng Tang, Weikai Chen, Jie Yang, Bo Wang, Songrun Liu, Bo Yang, and Lin Gao. Octfield: Hierarchical implicit functions for 3d modeling. In *The Annual Conference on Neural Information Processing Systems*, 2021. 2
- 1083 [62] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Carsten Stoll, and Christian Theobalt. PatchNets: Patch-Based Generalizable Deep Implicit 3D Shape Representations. *European Conference on Computer Vision*, 2020. 1, 2
- 1084 [63] Delio Vicini, Sbastien Speirer, and Wenzel Jakob. Differentiable signed distance function rendering. *ACM Transactions on Graphics*, 41(4):125:1–125:18, 2022. 2
- 1085 [64] Jiepeng Wang, Peng Wang, Xiaoxiao Long, Christian Theobalt, Taku Komura, Lingjie Liu, and Wenping Wang. NeuRIS: Neural reconstruction of indoor scenes using normal priors. In *European Conference on Computer Vision*, 2022. 2
- 1086 [65] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems*, pages 27171–27183, 2021. 2
- 1087 [66] Peng-Shuai Wang, Yang Liu, and Xin Tong. Dual octree graph networks for learning adaptive volumetric shape representations. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 2
- 1088 [67] Yiqun Wang, Ivan Skorokhodov, and Peter Wonka. HF-NeuS: Improved surface reconstruction using high-frequency details. 2022. 2
- 1089 [68] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 4
- 1090 [69] Yunjie Wu and Zhengxing Sun. DFR: differentiable function rendering for learning 3D generation from images. *Computer Graphics Forum*, 39(5):241–252, 2020. 2
- 1091 [70] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. SnowflakeNet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *IEEE International Conference on Computer Vision*, 2021. 1, 2
- 1092 [71] Lior Yariv, Jitao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Advances in Neural Information Processing Systems*, 2021. 2
- 1093 [72] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neu-
- 1094 ral surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33, 2020. 2
- 1095 [73] Wang Yifan, Shihao Wu, Cengiz Oztireli, and Olga Sorkine-Hornung. Iso-Points: Optimizing neural implicit surfaces with hybrid representations. *CoRR*, abs/2012.06434, 2020. 1, 2
- 1096 [74] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *IEEE International Conference on Computer Vision*, 2021. 2
- 1097 [75] Mulin Yu and Florent Lafarge. Finding good configurations of planar primitives in unorganized point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6367–6376, June 2022. 5
- 1098 [76] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Satler, and Andreas Geiger. MonoSDF: Exploring monocular geometric cues for neural implicit surface reconstruction. *ArXiv*, abs/2022.00665, 2022. 2
- 1099 [77] Sergey Zakharov, Wadim Kehl, Arjun Bhargava, and Adrien Gaidon. Autolabeling 3D objects with differentiable rendering of sdf shape priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2
- 1100 [78] Wenbin Zhao, Jiabao Lei, Yuxin Wen, Jianguo Zhang, and Kui Jia. Sign-agnostic implicit learning of surface self-similarities for shape modeling and reconstruction from raw point clouds. *CoRR*, abs/2012.07498, 2020. 1, 2
- 1101 [79] Junsheng Zhou, Baorui Ma, Yu-Shen Liu, Yi Fang, and Zhizhong Han. Learning consistency-aware unsigned distance functions progressively from raw point clouds. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 5, 6
- 1102 [80] Qingnan Zhou and Alec Jacobson. Thingi10k: A dataset of 10, 000 3d-printing models. *CoRR*, abs/1605.04797, 2016. 4
- 1103 [81] Qian-Yi Zhou and Vladlen Koltun. Dense scene reconstruction with points of interest. *ACM Transactions on Graphics*, 32(4):112:1–112:8, 2013. 4
- 1104 [82] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1105 [83] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1106 [84] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1107 [85] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1108 [86] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1109 [87] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1110 [88] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1111 [89] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1112 [90] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1113 [91] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1114 [92] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1115 [93] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1116 [94] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1117 [95] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1118 [96] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1119 [97] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1120 [98] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1121 [99] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1122 [100] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1123 [101] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1124 [102] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1125 [103] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1126 [104] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1127 [105] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1128 [106] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1129 [107] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1130 [108] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1131 [109] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1132 [110] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2
- 1133 [111] Mingchuan Zhu, Mingming Jiang, and Ming Tang. Multi-scale implicit volumes for 3D reconstruction. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 1, 2