GridFormer: Point-Grid Transformer for Surface Reconstruction

Shengtao Li[1,2], Ge Gao[1,2*], Yudong Liu[1,2], Yu-Shen Liu[2], Ming Gu[1,2]

[1]Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, China
[2]School of Software, Tsinghua University, Beijing, China
list21@mails.tsinghua.edu.cn, gaoge@tsinghua.edu.cn,
liuyd23@mails.tsinghua.edu.cn, liuyushen@tsinghua.edu.cn, guming@tsinghua.edu.cn

Abstract

Implicit neural networks have emerged as a crucial technology in 3D surface reconstruction. To reconstruct continuous surfaces from discrete point clouds, encoding the input points into regular grid features (plane or volume) has been commonly employed in existing approaches. However, these methods typically use the grid as an index for uniformly scattering point features. Compared with the irregular point features, the regular grid features may sacrifice some reconstruction details but improve efficiency. To take full advantage of these two types of features, we introduce a novel and high-efficiency attention mechanism between the grid and point features named Point-Grid Transformer (GridFormer). This mechanism treats the grid as a transfer point connecting the space and point cloud. Our method maximizes the spatial expressiveness of grid features and maintains computational efficiency. Furthermore, optimizing predictions over the entire space could potentially result in blurred boundaries. To address this issue, we further propose a boundary optimization strategy incorporating margin binary cross-entropy loss and boundary sampling. This approach enables us to achieve a more precise representation of the object structure. Our experiments validate that our method is effective and outperforms the state-of-the-art approaches under widely used benchmarks by producing more precise geometry reconstructions. The code is available at https://github.com/list17/GridFormer.

Introduction

Perceiving and modeling the surrounding world are essential tasks in 3D computer vision. Point clouds obtained from various sensors allow us to capture discrete spatial information about 3D surfaces directly. Surface reconstruction plays a vital role in converting this discrete representation into a continuous one. Recently, learning-based approaches have gained significant popularity in reconstructing point clouds. These implicit methods employ a conversion process that transforms the input point cloud into a global feature to represent the spatial continuous field of a 3D shape. However, bridging the gap between the continuous space and the discrete point cloud poses challenges, resulting in varying reconstruction outcomes across different representations.

*Corresponding author: Ge Gao

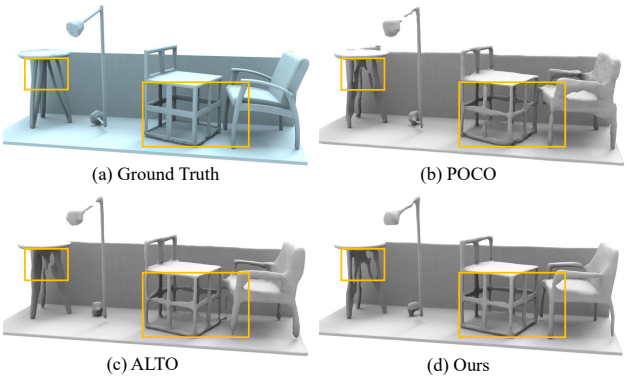


Figure 1: Visualization of the complex scene reconstruction results on the Synthetic Rooms dataset (Peng et al. 2020). Our method can produce high-fidelity reconstructions compared with the point-based method POCO (Boulch and Marlet 2022) and the grid-based method ALTO (Wang et al. 2023).

Typically, these methods encode the point cloud using either a single global feature or regular local grid features. The regular grid features are learned by uniformly distributing the point-wise features across each grid. While regular grid features capture information averagely from every point of the space, they may overlook the shape details in the point cloud. Some other methods will attach the features to the input points (Boulch and Marlet 2022; Zhang, Nießner, and Wonka 2022) or move the regular grid features close to the surface (Li et al. 2022). These irregular features can represent the 3D shape more accurately. However, connecting the irregular features with the space can be difficult. Also, the target occupancy function is not differentiable or even not continuous at the zero level. This intrinsic property increases the error bound and makes training more difficult.

To address these challenges, we propose a novel and highly efficient attention mechanism that bridges the space and point cloud by treating the grid as a transfer point. Figure 2 shows the difference between our approach and the previous techniques based on point or grid structures. The point-based method can effectively obtain the shape information from the point cloud, but the irregular structure of

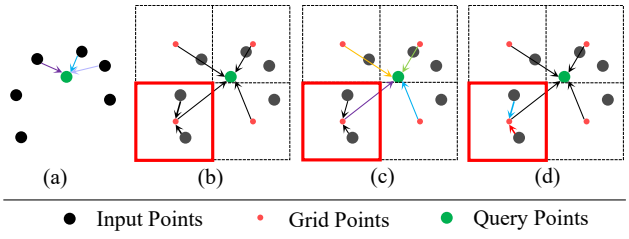


Figure 2: Comparisons between our GridFormer and other methods. The colorized arrows in (a), (c), and (d) represent learnable weights for scattering point or grid features. (a) The point-based approach expresses the query point feature by aggregating the nearby point features with learnable weights. (b) The grid-based approach learns the grid features by uniformly scattering the point features. The decoder aggregates the grid features by the weights calculated by bilinear or trilinear interpolation. (c) The attention-based decoder in ALTO (Wang et al. 2023) makes the weights between the query and grid points learnable. (d) Our point-grid transformer learns the weights between the input and grid features. This enables our method to approximate (a) through grid points while maintaining high efficiency.

the points significantly decreases efficiency. Our approach leverages the concept of point-grid attention to model the grid feature. This enables our network to learn the relationship between the input and grid features, which can implicitly bridge the space and the points. And the visual reconstruction differences between these methods can be found in Figure 1.

In particular, apart from employing uniform sampling, we have devised a two-stage training strategy. It incorporates margin binary cross-entropy loss and boundary sampling to narrow down the error bound caused by the discontinuity property, ensuring a more precise reconstruction result.

Our contributions can be listed as follows:

- We introduce the Point-Grid Transformer (GridFormer) for surface reconstruction. Our method significantly improves the spatial expressiveness of grid features for learning implicit neural fields.
- We design a two-stage training strategy incorporating margin binary cross-entropy loss and boundary sampling. This strategy enhances our model's predictive capability by yielding a more precise occupancy field near the surface.
- Both object-level and scene-level reconstruction experiments validate our method, demonstrating its effectiveness and ability to produce accurate reconstruction results.

Related Work

3D Representations

Explicit Representations. Voxels are amongst the most widely used shape representations (Maturana and Scherer 2015; Choy et al. 2016). However, as the resolution increases, the memory consumed by voxels increases dramatically. Different from voxels, point clouds represent a 3D shape as a set of discrete points (Qi et al. 2016, 2017). These points are irregularly distributed in space and lack continuous topological relationships. Hence post-processing steps (Kazhdan and Hoppe 2013) are needed to extract continuous surface. Meshes (Gkioxari, Malik, and Johnson 2019; Pan et al. 2019) avoid the complexity brought by voxels and can better represent the topological structure. However, generating mesh directly from the neural network is also more complicated. Most meshed-based methods require deforming geometric primitives (Groueix et al. 2018a) or templates of fixed topology (Groueix et al. 2018b).

Neural Implicit Representations. Implicit representation characterizes the whole space by predicting each point as inside, outside, or on the surface. It relies on a neural network acting as the function to model the binary occupancy field (Chen and Zhang 2019; Mescheder et al. 2019; Sitzmann, Zollhoefer, and Wetzstein 2019) or distance field (Gropp et al. 2020; Park et al. 2019; Atzmon and Lipman 2020; Takikawa et al. 2021) and then uses the marching cubes (Lorensen and Cline 1987) algorithm to extract the mesh. The implicit function typically receives a query point with a feature and outputs the corresponding occupancy or distance value. A single global feature has been applied to represent different shapes initially but it cannot capture local details. To resolve this, several works explored capturing local features both in the 2D image field (Saito et al. 2019, 2020; Xu et al. 2019) and in the 3D point cloud field (Chibane, Alldieck, and Pons-Moll 2020; Peng et al. 2020; Chen, Liu, and Han 2022; Baorui et al. 2022).

To obtain more faithful geometric features, some subsequent works also explored multi-resolution (Takikawa et al. 2021; Chen et al. 2021; Huang et al. 2023), irregular or dynamic feature representations (Li et al. 2022; Boulch and Marlet 2022; Zhang, Nießner, and Wonka 2022). Indeed, compared to the regular gird features (Peng et al. 2020; Tang et al. 2021; Lionar et al. 2021), the irregular feature representation is more compact and better suited to capture the details. Some other works (Baorui et al. 2021; Ben-Shabat, Hewa Koneputugodage, and Gould 2022; Ma et al. 2023) use the gradient or divergence to constrain the implicit fields for better reconstruction. Due to the intrinsic properties of the occupancy field, we adopt the margin to optimize our estimated occupancy function.

Transformers for Point Cloud

Transformer was first applied in NLP tasks (Vaswani et al. 2017) and has made a great success. It relies on a self-attention mechanism to capture the relationships between different words. This practical approach has also brought about innovations in other fields. Recently several works, like Point Transformer (Zhao et al. 2021) and Point Cloud Transformer (Guo et al. 2021), have explored the application of transformers in point cloud processing. However, due to the discrete distribution of point clouds, many of these approaches rely on the k-nearest neighbor (kNN) search to find

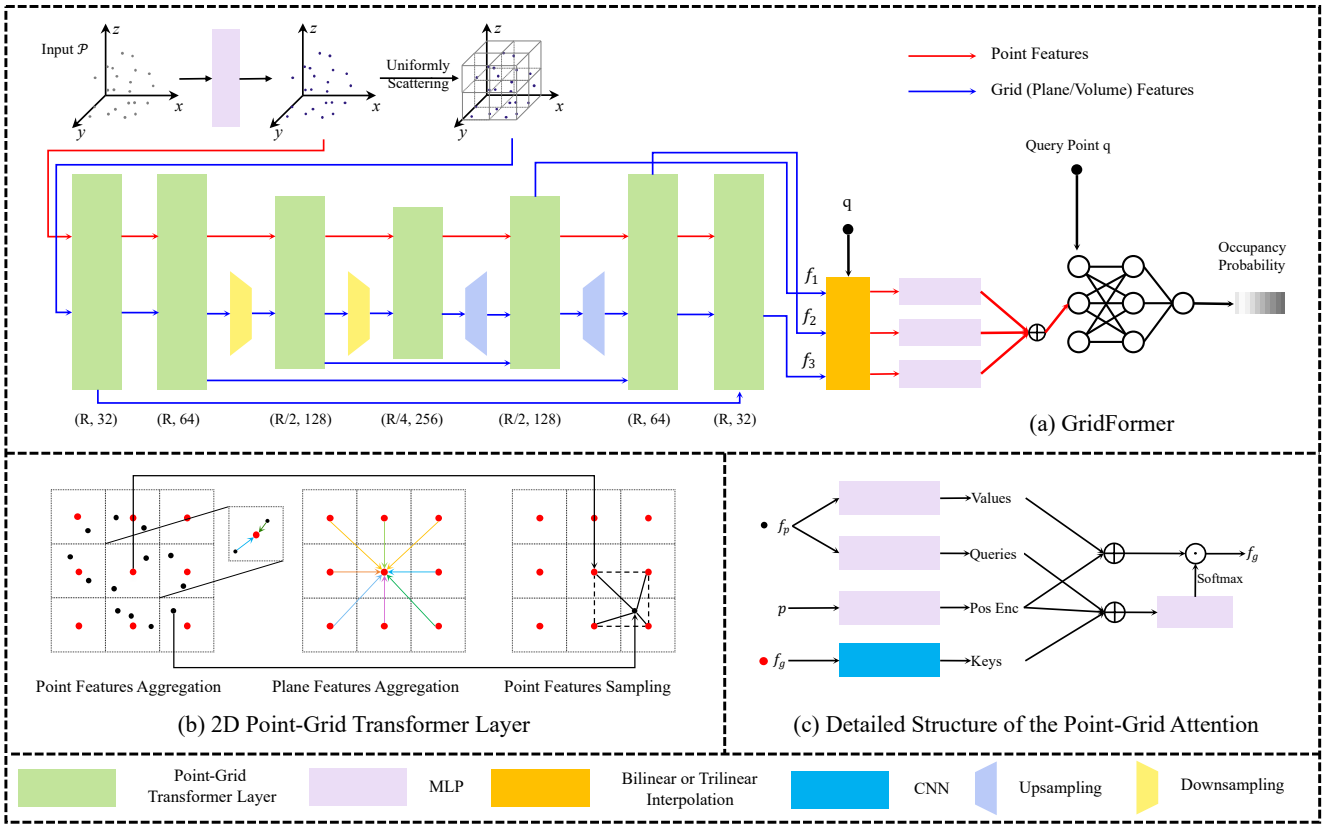


Figure 3: Overview of our method. (a) The architecture of GridFormer. (b) The 2D plane point-grid transformer layer in which the colorized arrows represent learnable weights. (c) The detailed structure of the point-grid attention mechanism for point features aggregation. 'Pos Enc' denotes position encoding.

the nearest neighboring points. As the size of the point cloud increases, the kNN search becomes more complex and computationally expensive. Fast Point Transformer (Park et al. 2022) designs a lightweight self-attention layer that uses the voxel hashing-based architecture to boost computational efficiency. Our method also utilizes grids for the aggregation of point features. But our innovation lies in utilizing the fixed grid not only for acceleration but also to connect the space and the point cloud.

Method

Our method aims to establish an efficient attention mechanism for connecting the space and the point cloud. Figure 3(a) shows an overview of our network structure. Based on our point-grid transformer layer, the model constructs a continuous occupancy function $o : \mathbf{R}^3 \rightarrow [0, 1]$. For a point cloud $\mathcal{P} = \{p_i \mid p_i \in \mathbf{R}^3\}$, we first learn the per-point features by applying a small point-wise multi-layer perception (MLP). Then the grid (plane or volume) features are initialized by scattering the point features uniformly. The U-Net-like network is constructed based on the point-grid transformer layer. Each layer takes in point and grid features. In the following sections, we will elaborate on the point-grid transformer layer, the multi-resolution decoder, and our subsequent optimization strategy.

Point-Grid Transformer Layer

Given the point cloud $\mathcal{P}$, we define the f_p, f_g, f_q as the features of input, grid, and query points, and the p, g, q represent the corresponding points. ϕ and ψ represent the MLP and convolutional neural network (CNN), respectively. The steps of the point-grid transformer layer, as illustrated in Figure 3(b) and (c), will be described in detail below.

Position Encoding. We convert the point clouds from the global coordinate system to the local coordinate system of the grid where the point is situated. Then we use an MLP ϕ_{pos} with two linear layers and one ReLU nonlinearity function. It takes the localized points as input and outputs the position encoding, as denoted by

$$f_{pos} = \phi_{pos}(p - \lfloor (p \times r) \rfloor / r), \tag{1}$$

where r represents the plane or volume resolution.

Point Features Aggregation. This section will describe the point-grid attention mechanism used to aggregate the point features. We leverage the point transformer mechanism in (Zhao et al. 2021) to learn the weights between the point and grid features. The detailed structure is shown in Figure 3(c). A small CNN ψ_k is applied to the grid features to learn the keys, and two MLP networks ϕ_q and ϕ_v

are utilized to learn the queries and values of the point features, respectively. We also add position encoding in both the point-grid attention generation branch and the feature transformation branch following (Zhao et al. 2021). The point-grid weights can be represented as follows:

$$\omega_{ij} = \phi_w(\psi_k(f_{g_i}) - \phi_q(f_{p_j}) + f_{pos}). \quad (2)$$

Here the point p_j is situated in the same grid $\mathcal{N}_{g_i}$ as the grid point g_i. The softmax function is applied to normalize the weights in the same grid. Then we aggregate the point features by our learned point-grid weights:

$$f_{g_i} = \sum_{p_j \in \mathcal{N}_{g_i}} \omega_{ij}(\phi_v(f_{p_j}) + f_{pos}). \quad (3)$$

Grid Features Aggregation. To increase the receptive field of each grid, we further aggregate the neighbor grid features by a small CNN. Considering our motivation and the lightweight network design, we replace the CNN with the depth-wise convolutional network (Chollet 2017) in the last three point-grid transformer layers directly connected to the decoder. The depth-wise convolution convolves each input channel with a different kernel which acts as the shared weights between the grid features.

Point Features Sampling. Merely updating the point features from f_{p_j} to $\phi_v(f_{p_j}) + f_{pos}$ will disregard the point features within the neighborhood. To address this issue, we opt to sample the point features from the hybrid grid features using bilinear or trilinear interpolation denoted as $\mathcal{S}$. Simultaneously, considering that the grid features inherently contain position encoding, we omit this component during this stage. The revised point features can be mathematically expressed as

$$f_{p_i} = \phi_v(f_{p_j}) + \mathcal{S}(f_g). \quad (4)$$

Throughout this entire stage, the grid features only serve the purpose of computing the keys. Consequently, we implement a skip connection for both the point and grid features within one attention layer.

Multi-resolution Decoder

For any given query point q, we can sample the query feature by bilinear or trilinear interpolation, leveraging our learned grid features as shown in Figure 2(d). The weights in the interpolation are fixed for a certain query point since they are computed by the relative distance. An improved method is learning the weights also through the attention mechanism between the query and grid points like the Figure 2(c). However, In the feature transfer chain from the input point cloud to the grid points, then to the query points, we already make the weights of the front half part learnable. At the same time, the number of query points will increase rapidly when we need a high-resolution reconstruction result. An attention-based decoder will consume more time to decode the occupancy value for each query point.

Based on the aforementioned factors, we still keep the interpolation method to sample the query features. We opt for grid features $f = (f_1, f_2, f_3)$ from two different resolutions as illustrated in Figure 3(a). We scale the different feature dimensions to 32 through a shallow MLP to keep the same settings as other methods. We use the same network o_θ for the decoder as (Peng et al. 2020), which takes in the accumulated feature $f_q \in \mathcal{F}$ and the query point $q \in \mathbf{R}^3$ and outputs the occupancy:

$$o_\theta : \mathbf{R}^3 \times \mathcal{F} \rightarrow [0, 1]. \quad (5)$$

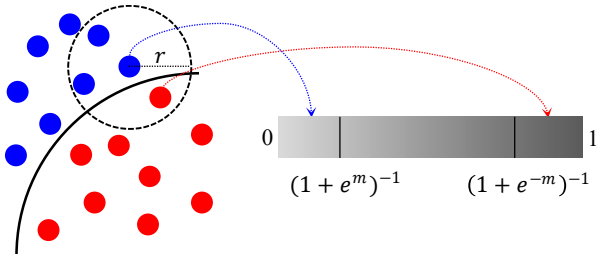


Figure 4: Illustration of boundary optimization.

Boundary Optimization

In this section, we propose boundary optimization for the occupancy function. According to the definition of the occupancy function $o : \mathbf{R}^3 \rightarrow [0, 1]$, it is not continuous and not differential on the surface. Based on these intrinsic features, we adopt the margin binary cross-entropy loss to finetune our model with the boundary sampling.

Boundary Sampling. The uniform sampling strategy has been proven to be the most suitable training strategy in (Mescheder et al. 2019) as other alternative sampling strategies tend to introduce bias to the model. But for a precise reconstruction, it is required to have accurate predictions near the surface. Hence, we divide the whole training procedure into two stages. At the first stage, we use uniform sampling for training until the model converges. At the second stage, we switch to boundary sampling. To ensure fairness, we extract boundary regions from the original training data rather than resampling the query points.

Since both the noise levels and densities of the point clouds are different, extracting the region based on the input point cloud is insufficient. We extract the boundary points based on the ground-truth occupancy labels the same as (Tang et al. 2022). A point is a boundary point only if at least one of its neighboring points lies on the opposite side of the surface. A fixed radius r is set to search for the opposite points shown in Figure 4.

Margin Binary Cross-entropy Loss. At the first stage, we minimize the binary cross-entropy loss between the predicted $\hat{o}_q$ and the ground-truth occupancy values o_q with uniformly sampled points $q \in \mathbf{R}^3$:

$$\mathcal{L}(\hat{o}_q, o_q) = -[o_q \cdot log(\hat{o}_q) + (1 - o_q) \cdot log(1 - \hat{o}_q)] \quad (6)$$

where o_q is calculated by applying a sigmoid layer to the output of the network $o(q)$:

$$o_q = \frac{1}{1 + e^{-o(q)}}. \quad (7)$$

Methods	3000 Pts, σ = 0.005				1000 Pts, σ = 0.005				300 Pts, σ = 0.005			
	IoU ↑	CD ↓	NC ↑	FS ↑	IoU ↑	CD ↓	NC ↑	FS ↑	IoU ↑	CD ↓	NC ↑	FS ↑
ONet (Mescheder et al. 2019)	0.761	0.87	0.891	0.785	0.772	0.81	0.894	0.801	0.778	0.80	0.895	0.806
ConvONet (Peng et al. 2020)	0.884	0.44	0.938	0.942	0.859	0.50	0.929	0.918	0.821	0.59	0.907	0.883
POCO (Boulch and Marlet 2022)	0.926	0.30	0.950	0.984	0.884	0.40	0.928	0.950	0.808	0.61	0.892	0.869
ALTO (Wang et al. 2023)	0.930	0.30	0.952	0.980	0.905	0.35	0.940	0.964	0.863	0.47	0.922	0.924
Ours	**0.936**	**0.28**	**0.956**	**0.985**	**0.913**	**0.33**	**0.946**	**0.970**	**0.866**	**0.46**	**0.925**	**0.926**

Table 1: Comparison on the ShapeNet dataset with different point density levels. 'Pts' denotes input points and σ is the standard deviation of the Gaussian noise.

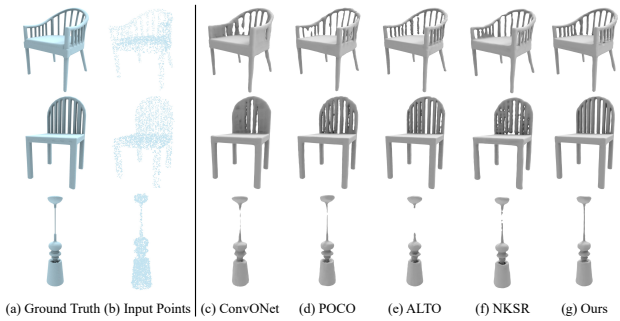


Figure 5: Object-level reconstruction results on the ShapeNet dataset. All the methods are trained and tested on 3000 noisy points.

A margin m is added directly to the output according to the ground-truth label $l \in [0, 1]$. Consequently, the new output is defined as:

$$o_q = \frac{1}{1 + e^{-(o(q) - m \times (l \times 2 - 1))}}. \tag{8}$$

As shown in Figure 4, the margin binary cross-entropy loss can make the predicted occupancy values as close to 0 or 1 as possible.

Implementation Details

We implement our model in Pytorch (Paszke et al. 2019) and use the Adam optimizer (Kingma and Ba 2014). The learning rate is 10^{-4} at the first stage, and 10^{-6} at the finetune stage. The depth of our U-Net-like encoder is 4, and we do not downsample or upsample the grid features in the two top levels the same as (Wang et al. 2023). The radius r to search for the opposite points is set to 0.08 and the margin m is set to 2.0. At reference time, we apply Multiresolution IsoSurface Extraction (MISE) (Mescheder et al. 2019) to obtain the mesh.

Experiments

Datasets, Metrics, and Baselines

ShapeNet. We use ShapeNet (Chang et al. 2015) for object-level reconstruction evaluation. ShapeNet preprocessed by ONet (Mescheder et al. 2019) contains watertight meshes of shapes in 13 classes, with train/val splits and 8500 objects for testing. To comprehensively evaluate our method, we leverage two different settings for a fair comparison. Following ALTO (Wang et al. 2023), we sample different densities of points and add Gaussian noise with zero mean and standard deviation of 0.005. To evaluate the effect of noise, we follow NKSR (Huang et al. 2023) to sample points with different noise levels.

Scene-Level Datasets. For our scene-level reconstruction, we use the Synthetic Rooms dataset (Peng et al. 2020) and ScanNet-v2 (Dai et al. 2017). The Synthetic Rooms dataset comprises 5000 synthetic room scenes containing randomly placed walls, floors, and ShapeNet objects. We utilize the same train/validation/test division as previously established.

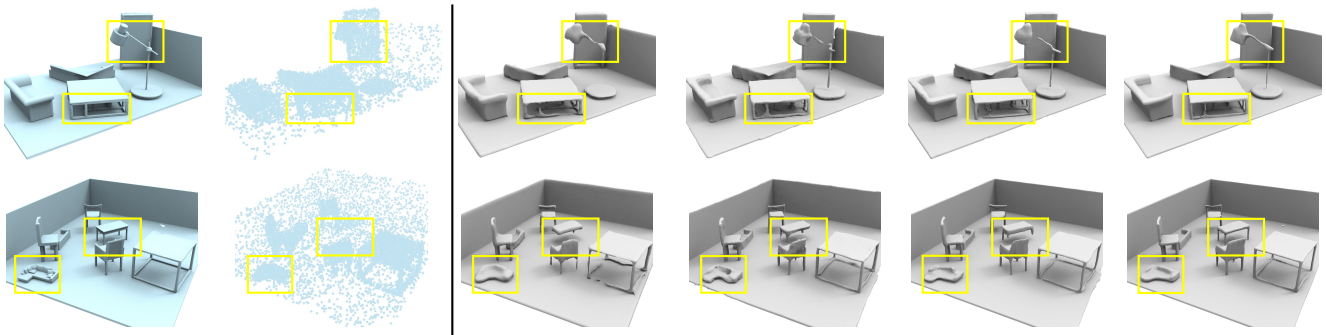
The ScanNet-v2 dataset includes 1513 scans of real-world environments featuring a diverse selection of room types. The meshes provided in ScanNet-v2 are not watertight, so models are trained using the Synthetic Rooms dataset and then tested on ScanNet-v2. This enables the evaluation of the generalization performance of our method.

Evaluation Metrics. Following ConvONet (Peng et al. 2020), we use the volumetric IoU, Chamfer-L_1 distance $\times 10^2$ (CD), normal consistency (NC), and F-Score (Tatarchenko* et al. 2019) with threshold value 1% (FS) metrics for our evaluation. Other used metrics also include the Chamfer-L_2 distance (L2-CD). Please refer to the supplementary of (Peng et al. 2020) for the mathematical details.

Baselines. To evaluate the validity of our attention mechanism, the baselines used for comparison include ONet (Mescheder et al. 2019), ConvONet (Peng et al. 2020), POCO (Boulch and Marlet 2022), and ALTO (Wang et al. 2023). In addition to these, we also include SPSR (Kazhdan and Hoppe 2013), SAP (Peng et al. 2021), and NKSR (Huang et al. 2023). Please note that NKSR utilized point normals in most of their experiments. To ensure fairness in our comparison, we only evaluate their results from training without point normals.

Object-level Reconstruction

We first evaluate our method on the task of single-object reconstruction. The quantitative results of different density levels are shown in Table 1. Our method performs better than the point-based and other grid-based methods. We also find that when the points are too sparse (300 points), the improvement will become smaller because learning the weight



(a) Ground Truth (b) Input Points (c) ConvONet (d) POCO (e) ALTO (f) Ours

Figure 6: Scene-level comparisons on the Synthetic Rooms dataset. Our method preserves most of the details of the furniture.

Methods	1000Pts $\sigma = 0.0$		3000 Pts $\sigma = 0.005$		3000 Pts $\sigma = 0.025$	
	IoU ↑	CD ↓	IoU ↑	CD ↓	IoU ↑	CD ↓
ConvONet (Peng et al. 2020)	0.823	0.61	0.880	0.44	0.787	0.73
SAP (Peng et al. 2021)	0.908	0.34	0.911	0.33	0.829	0.53
POCO (Boulch and Marlet 2022)	0.927	0.30	0.926	0.30	0.817	0.58
ALTO (Wang et al. 2023)	0.940	0.29	0.931	0.30	0.839	0.51
NKSR (Huang et al. 2023)	0.934	**0.26**	0.926	**0.27**	0.829	**0.50**
Ours	**0.946**	0.28	**0.936**	0.28	**0.844**	**0.50**

Table 2: Comparison on the ShapeNet dataset with different noise levels.

for a single point is meaningless. This also verifies the effectiveness of our attention mechanism. We also evaluate the effect of noise following the setting of NKSR (Huang et al. 2023) in Table 2. Qualitative comparisons are provided in Figure 5. Compared with other baselines, our method can capture more details, and the overall topology of the shape is more unified and consistent.

Methods	IoU ↑	CD ↓	NC ↑	FS ↑
ONet (Mescheder et al. 2019)	0.475	2.03	0.783	0.541
SPSR (Kazhdan and Hoppe 2013)	-	2.23	0.866	0.810
ConvONet (Peng et al. 2020)	0.849	0.42	0.915	0.964
DP-ConvONet (Lionar et al. 2021)	0.800	0.42	0.912	0.960
POCO (Boulch and Marlet 2022)	0.884	0.36	0.919	0.980
ALTO (Wang et al. 2023)	0.914	0.35	0.921	0.981
Ours	**0.918**	**0.34**	**0.926**	**0.983**

Table 3: Comparison on the Synthetic Rooms dataset.

Scene-level Reconstruction

We report numerical comparisons in Table 3 with previous point-based and grid-based methods. The comparison verifies the validity of our proposed mechanism on the scenes. We further present the visual comparison in Figure 6, which shows that our method achieves more detailed reconstruction results, particularly for finer structures.

Real-world Generalization

We also explore the generalization performance of our method under the ScanNet-v2 dataset in Table 4. Figure 7 illustrates that our method can reconstruct a smoother and more complete surface than the other methods. At the same time, we also find that since the Synthetic Rooms dataset only contains regularly generated walls and floors, which are different from the real-scanned rooms as shown in Figure 6 and Figure 7. This issue causes all methods to try to complete the irregular walls and floors to some extent, like the second group of reconstruction results in Figure 7. The completed parts will greatly influence the metrics. For a more accurate comparison, we remove the floors following the method used in ConvONet (Peng et al. 2020) but keep the walls since they are randomly seated throughout the scene.

Methods	CD ↓	NC ↑	FS ↑
ConvONet (Peng et al. 2020)	0.80	0.816	0.810
DP-ConvONet (Lionar et al. 2021)	1.35	0.769	0.682
POCO (Boulch and Marlet 2022)	0.74	0.813	0.816
ALTO (Wang et al. 2023)	0.79	0.802	0.809
Ours	**0.71**	**0.822**	**0.846**

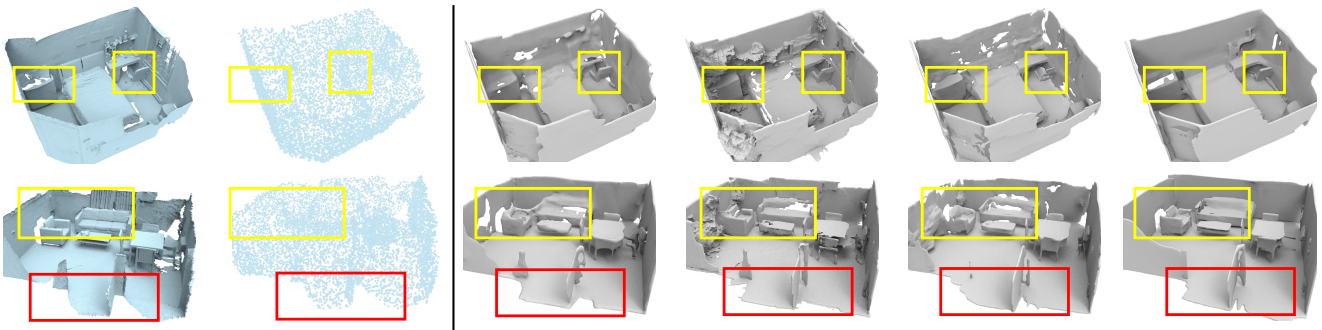
Table 4: Comparison of the generalization performance on the ScanNet dataset. All methods are trained on the Synthetic Rooms dataset and tested on ScanNet-v2 without floors.

Ablation Study

Grid Representation. We report the effect of our point-grid attention for different representations (triplane and volume) in table 5 on the Synthetic Rooms dataset. All methods use the same decoder without attention.

Method	IoU ↑	CD ↓	NC ↑	FS ↑
ConvONet ($3 * 128^2$) (Peng et al. 2020)	0.805	0.44	0.903	0.948
ConvONet (64^3) (Peng et al. 2020)	0.849	0.42	0.915	0.964
ALTO ($3 * 128^2$) (Wang et al. 2023)	0.834	0.43	0.906	0.960
ALTO (64^3) (Wang et al. 2023)	0.903	0.36	0.920	0.981
Ours ($3 * 128^2$)	0.835	0.40	0.900	0.949
Ours (64^3)	**0.918**	**0.34**	**0.926**	**0.983**

Table 5: Ablation on the network framework.



(a) Ground Truth (b) Input Points (d) ConvONet (d) POCO (d) ALTO (d) Ours

Figure 7: Generalization performance on the ScanNet dataset. These methods are trained on the Synthetic Rooms dataset and tested on the ScanNet-v2 dataset. The red squares show all the methods will complete the floor to some extent.

Grid Downsampling. We experiment without downsampling in our U-Net-like encoder and the results are shown in Table 6. It takes less time to encode the input points but the reconstruction results are significantly inferior. Thus we consider it worthy to consume a little bit more time for a better result.

Method	IOU ↑	CD ↓	NC ↑	Encode Time (s) ↓
Ours (w/o downsampling)	0.897	0.38	0.949	**0.39**
Ours (w/ downsampling)	**0.942**	**0.30**	**0.962**	0.41

Table 6: Ablation on the gird downsampling.

Boundary Optimization. We further explore the effect of our boundary optimization. Table 7 shows that the optimization works under different noise and density levels. We also visualize the distance from reconstruction results to the ground-truth meshes in Figure 8. The proposed boundary optimization can effectively help reduce the error bound.

	3000 Pts $\sigma = 0.005$	3000 Pts $\sigma = 0.025$	1000 Pts $\sigma = 0$	1000 Pts $\sigma = 0.005$
w/o optimzation	0.211	0.881	0.253	0.371
w/ optimization	**0.201**	**0.718**	**0.247**	**0.328**

Table 7: Ablation study on boundary optimization in terms of L2-CD ($\times 10^4$) .

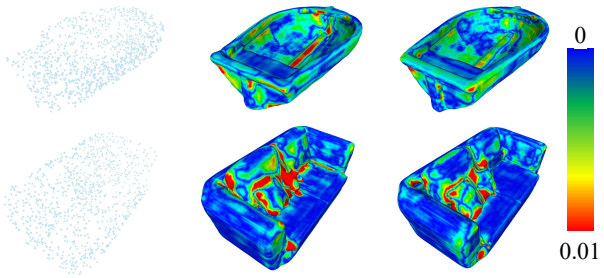
Method	GPU Memory (MiB)	Inference Time (s)
ConvONet (Peng et al. 2020)	1957	0.43
POCO (Boulch and Marlet 2022)	6540	9.30
ALTO (Wang et al. 2023)	3257	7.96
Ours	1915	5.61

Table 8: GPU memory and runtime comparisons on ShapeNet chairs.

Conclusion

We proposed the Point-Grid Transformer (GridFormer) using a novel point-grid attention mechanism between the point and grid features. It is valid both for object-level and scene-level reconstruction and reconstructs a smoother surface on the unseen dataset. Compared with the attention-based decoder used in the point-based and other grid-based methods, our attention-based encoder costs less time and GPU memory (Table 8) and achieves comparable or even better results. Our introduced boundary optimization strategy can reduce the error between the estimated and ground-truth occupancy functions and help extract a more accurate surface.

Finally, the experiments show that both the density of the input points and the grid size impact our method's effect. In future work, exploring how to dynamically divide the grid to achieve the attention mechanism between different resolutions may apply this mechanism to more scenarios.



(a) Input Points (b) w/o optimization (c) w/ optimization

Figure 8: The visual effect of boundary optimization.

Acknowledgments

The corresponding author is Ge Gao. This work was supported by the National Key Research and Development Program of China (2021YFB1600303).

References

Atzmon, M.; and Lipman, Y. 2020. SAL: Sign Agnostic Learning of Shapes From Raw Data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Baorui, M.; Yu-Shen, L.; Matthias, Z.; and Zhizhong, H. 2022. Surface Reconstruction from Point Clouds by Learning Predictive Context Priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Baorui, M.; Zhizhong, H.; Yu-Shen, L.; and Matthias, Z. 2021. Neural-Pull: Learning Signed Distance Functions from Point Clouds by Learning to Pull Space onto Surfaces. In *International Conference on Machine Learning (ICML)*.

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

Boulch, A.; and Marlet, R. 2022. POCO: Point Convolution for Surface Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6302–6314.

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; Xiao, J.; Yi, L.; and Yu, F. 2015. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago.

Chen, C.; Liu, Y.-S.; and Han, Z. 2022. Latent Partition Implicit with Surface Codes for 3D Representation. In *European Conference on Computer Vision (ECCV)*.

Chen, Z.; and Zhang, H. 2019. Learning Implicit Fields for Generative Shape Modeling. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chen, Z.; Zhang, Y.; Genova, K.; Funkhouse, T.; Fanello, S.; Bouaziz, S.; Haene, C.; Du, R.; Keskin, C.; and Tang, D. 2021. Multiresolution Deep Implicit Functions for 3D Shape Representation. In *2021 IEEE/CVF International Conference on Computer Vision*, ICCV. IEEE.

Chibane, J.; Alldieck, T.; and Pons-Moll, G. 2020. Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Chollet, F. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1251–1258.

Choy, C. B.; Xu, D.; Gwak, J.; Chen, K.; and Savarese, S. 2016. 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5828–5839.

Gkioxari, G.; Malik, J.; and Johnson, J. 2019. Mesh r-cnn. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9785–9795.

Gropp, A.; Yariv, L.; Haim, N.; Atzmon, M.; and Lipman, Y. 2020. Implicit Geometric Regularization for Learning Shapes. In *Proceedings of Machine Learning and Systems 2020*, 3569–3579.

Groueix, T.; Fisher, M.; Kim, V. G.; Russell, B.; and Aubry, M. 2018a. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Groueix, T.; Fisher, M.; Kim, V. G.; Russell, B. C.; and Aubry, M. 2018b. 3D-CODED: 3D Correspondences by Deep Deformation. In *European Conference on Computer Vision*.

Guo, M.-H.; Cai, J.-X.; Liu, Z.-N.; Mu, T.-J.; Martin, R. R.; and Hu, S.-M. 2021. PCT: Point cloud transformer. *Computational Visual Media*, 7(2): 187–199.

Huang, J.; Gojcic, Z.; Atzmon, M.; Litany, O.; Fidler, S.; and Williams, F. 2023. Neural Kernel Surface Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4369–4379.

Kazhdan, M.; and Hoppe, H. 2013. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3): 1–13.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, T.; Wen, X.; Liu, Y.-S.; Su, H.; and Han, Z. 2022. Learning Deep Implicit Functions for 3D Shapes with Dynamic Code Clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Lionar, S.; Emtsev, D.; Svilarkovic, D.; and Peng, S. 2021. Dynamic Plane Convolutional Occupancy Networks. In *Winter Conference on Applications of Computer Vision (WACV)*.

Lorensen, W. E.; and Cline, H. E. 1987. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, 163–169. New York, NY, USA: Association for Computing Machinery. ISBN 0897912276.

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 *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Maturana, D.; and Scherer, S. 2015. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Mescheder, L.; Oechsle, M.; Niemeyer, M.; Nowozin, S.; and Geiger, A. 2019. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pan, J.; Han, X.; Chen, W.; Tang, J.; and Jia, K. 2019. Deep Mesh Reconstruction from Single RGB Images via Topology Modification Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 9964–9973.

Park, C.; Jeong, Y.; Cho, M.; and Park, J. 2022. Fast Point Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16949–16958.

Park, J. J.; Florence, P.; Straub, J.; Newcombe, R.; and Lovegrove, S. 2019. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Peng, S.; Jiang, C. M.; Liao, Y.; Niemeyer, M.; Pollefeys, M.; and Geiger, A. 2021. Shape As Points: A Differentiable Poisson Solver. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Peng, S.; Niemeyer, M.; Mescheder, L.; Pollefeys, M.; and Geiger, A. 2020. Convolutional Occupancy Networks. In *European Conference on Computer Vision (ECCV)*.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2016. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *arXiv preprint arXiv:1612.00593*.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *arXiv preprint arXiv:1706.02413*.

Saito, S.; ; Huang, Z.; Natsume, R.; Morishima, S.; Kanazawa, A.; and Li, H. 2019. PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization. *arXiv preprint arXiv:1905.05172*.

Saito, S.; Simon, T.; Saragih, J.; and Joo, H. 2020. PIFuHD: Multi-Level Pixel-Aligned Implicit Function for High-Resolution 3D Human Digitization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Sitzmann, V.; Zollhoefer, M.; and Wetzstein, G. 2019. Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Takikawa, T.; Litalien, J.; Yin, K.; Kreis, K.; Loop, C.; Nowrouzezahrai, D.; Jacobson, A.; McGuire, M.; and Fidler, S. 2021. Neural Geometric Level of Detail: Real-time Rendering with Implicit 3D Shapes.

Tang, J.; Lei, J.; Xu, D.; Ma, F.; Jia, K.; and Zhang, L. 2021. SA-ConvONet: Sign-Agnostic Optimization of Convolutional Occupancy Networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Tang, L.; Zhan, Y.; Chen, Z.; Yu, B.; and Tao, D. 2022. Contrastive Boundary Learning for Point Cloud Segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8479–8489.

Tatarchenko*, M.; Richter*, S. R.; Ranftl, R.; Li, Z.; Koltun, V.; and Brox, T. 2019. What Do Single-view 3D Reconstruction Networks Learn?

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, Z.; Zhou, S.; Park, J. J.; Paschalidou, D.; You, S.; Wetzstein, G.; Guibas, L.; and Kadambi, A. 2023. ALTO: Alternating Latent Topologies for Implicit 3D Reconstruction. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Xu, Q.; Wang, W.; Ceylan, D.; Mech, R.; and Neumann, U. 2019. DISN: Deep Implicit Surface Network for High-quality Single-view 3D Reconstruction. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Zhang, B.; Nießner, M.; and Wonka, P. 2022. 3DILG: Irregular Latent Grids for 3D Generative Modeling. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16259–16268.