DIFFERENTIABLE GRADIENT SAMPLING FOR LEARNING IMPLICIT 3D SCENE RECONSTRUCTIONS FROM A SINGLE IMAGE

Shizhan Zhu, Sayna Ebrahimi, Angjoo Kanazawa, Trevor Darrell
UC Berkeley

ABSTRACT

Implicit neural shape functions, e.g. occupancy fields or signed distance functions (SDF), are promising 3D representations for modeling clean mesh surfaces. Existing approaches that use these representations for single-view object reconstruction require supervision signals from *every* location in the scene, posing difficulties when extending to real-world scenarios where ideal watertight geometric training data is difficult to obtain. In such cases, constraints on the spatial gradient of the implicit field, rather than the value itself, can provide a training signal, but this has not been employed as a source of supervision for single-view reconstruction in part due to the difficulties of differentiably sampling a spatial gradient from a feature map. In this paper, we derive a novel closed-form Differentiable Gradient Sampling (DGS) solution that enables backpropagation of the loss on spatial gradients to the feature maps, thus allowing training on large-scale scenes without dense 3D supervision. As a result, we demonstrate single view implicit surface reconstructions on real-world scenes via learning directly from a scanned dataset. Our model performs well when generalizing to unseen images from Pix3D or downloaded directly from the internet (Fig. 1). We comfortably advance the state-of-the-art with results on several established datasets including ShapeNet and ScannetV2; extensive quantitative analysis confirms that our proposed DGS module plays an essential role in achieving this performance improvement. Full codes are available in https://github.com/zhusz/ICLR22-DGS.

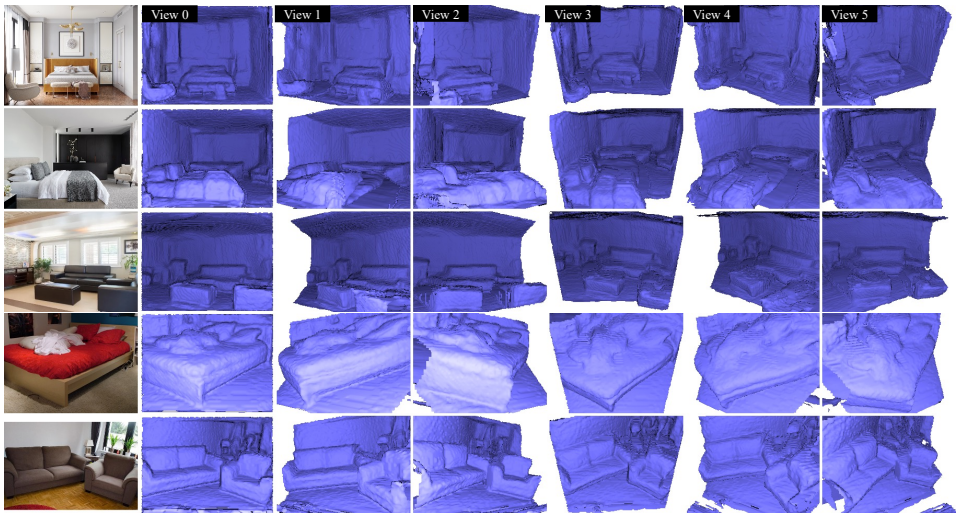

Figure 1: Given a single RGB image as the input (the first column in each row), our model can predict its 3D implicit surface reconstruction (shown in six novel views in the last six columns). The test images for the first 3 rows are downloaded from the Internet and the last 2 rows are from the pix3d dataset.

1 INTRODUCTION

Recent studies have shown promising learning capabilities of implicit models for 3D geometry representations (Park et al., 2019; Mescheder et al., 2019; Mildenhall et al., 2020; Sitzmann et al., 2019; Gropp et al., 2020; Saito et al., 2019; Yu et al., 2020). Rather than explicitly representing 3D geometry with a textured mesh (Kato et al., 2018) or point cloud (Fan et al., 2017), implicit representations express a function of points in the space. Among many recent promising implicit models, the occupancy field (Mescheder et al., 2019; Popov et al., 2020; Chen & Zhang, 2019b) and the Signed Distance Field (SDF) (Park et al., 2019; Xu et al., 2019; Li & Zhang, 2021) are particularly suitable for high resolution mesh surface reconstruction. Given a query point $(x, y, z) \in \mathbb{R}^3$ in the space, the occupancy function returns the occupancy status of the point $o(x, y, z) \in \{+, -\}$, while the SDF value encompasses the closest distance from a point to the surface of the given 3D geometry, as well as its sign for its occupancy. The mesh surface is defined by the classification boundary of the occupancy field or the zero level-set of the SDF (Lorensen & Cline, 1987).

Despite the strong representational power of these models and their success on predicting single view objects, learning to predict complex 3D geometry, such as scenes, from a real world image remains a challenge for this class of methods. Obtaining the occupancy label, or the sign, from noisy non-watertight meshes is non-trivial (Atzmon & Lipman, 2020) (Fig. 2). Furthermore, producing the label on the fly from a large number of triangle meshes, as encountered in typical real-world scenes with complex geometry, can be computationally prohibitive (Straub et al., 2019; Chang et al., 2017; Dai et al., 2017; Song et al., 2017). Storing sufficiently dense query points with pre-computed occupancies or SDF for complex geometries, or meta data of meshes like octrees or hashing, is a possibility, but incurs severe engineering and computational burdens on storage and runtime loading (Yu et al., 2021). The challenge is exacerbated for predicting scenes in particular, since unlike objects scenes in the view frustrum typically cannot be enclosed by a pre-defined range of space and do not have a categorical canonical coordinate system (Popov et al., 2020).

Recent studies on fitting a single scene geometry (Gropp et al., 2020; Sitzmann et al., 2020) have pointed out the value of the 3D spatial gradient of the implicit field. For instance, the gradients of the SDF on surface points are actually the surface normals and the 2-norm of the spatial gradients for any point in the space is 1, known as the Eikonal equation (Crandall & Lions, 1983). Similarly, for the occupancy case, we propose to regularize the gradient of the occupancy field to be zero away from the surface (Fig. 3(c)). Thus, one can train the full occupancy field or SDF with only the surface point clouds, thanks to these constraints on the gradients of the implicit shape models, even when the value of the occupancy or the SDF itself is not available everywhere. However, existing approaches (Gropp et al., 2020; Sitzmann et al., 2020) have only demonstrated this benefit in the cases of geometry only inputs, where a network is fit to a particular scene per model without any conditioning on input images. When the locally sampled image features are used for the feed-forward prediction, it is necessary to derive a differentiable gradient sampling solution over the feature pixel map.

In this paper, we tackle this challenge by deriving a closed-form differentiable gradient sampling (DGS) solution for learning a single-view 3D implicit reconstruction model (Fig. 4). Our novel propagation of the loss gradient of the spatial gradient back to the feature maps (Eq. 6) serves in addition to the existing spatial gradient sampling (Eq. 5) and loss gradient back-propagation (Eq. 4) used in existing deep learning frameworks (Paszke et al., 2019; Abadi et al., 2016). The resulting end-to-end learning scheme with supervision over the spatial gradients opens the potential to train a model that can generalize to unseen test cases of complex scenes with only surface point cloud supervision — a typical setting for real-world scene data — without requiring ground truth per-query-point labels.

Our contributions include a framework to propagate spatial gradients through a spatial feature sampling procedure, and a novel closed-form DGS solution. We further show that DGS enables training a single-image 3D implicit reconstruction model that can generalize to unseen scenes, with only imperfect surface annotations as the supervision (e.g. non-watertight meshes). Finally, experiments on synthetic (ShapeNet Chang et al. (2015)) and real-scanned data (ScannetV2 (Dai et al., 2017)) indicate that we clearly advanced state-of-the-art accuracy. To the best of our knowledge, DGS-enabled shape inference provides the first single-view implicit shape reconstruction on real scene datasets which can generalize accurately to unseen scenes from different datasets or domains (see Figure 1).

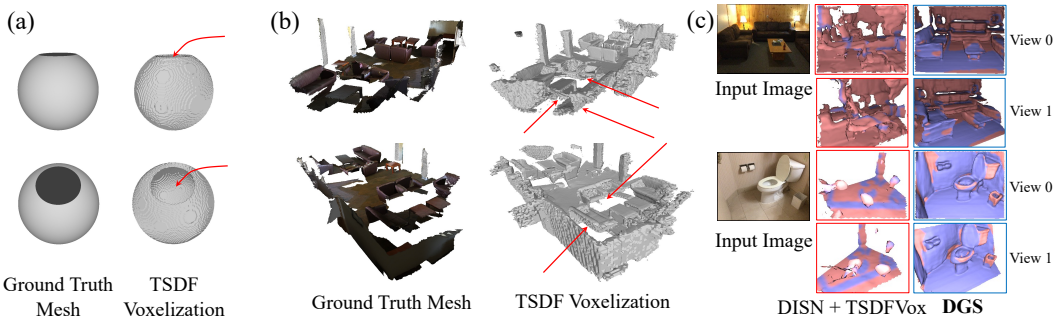

Figure 2: (a, b) Truncated SDF (TSDF) (Curless & Levoy, 1996) Voxelization results of the non-watertight ground truth meshes (each shown in two views). (a) is a simple sphere and (b) is a real scene from the ScannetV2 training data. After depth-fusion and internal-filling (Popov et al., 2020), the inside space of both geometries (a, b) remains empty (red arrows), causing severe noise for training the occupancy field or the SDF prediction model. This type of noise particularly affects the single-view prediction problem, as no additional predicted depth surface from other views are available. (c) As a result of learning the implicit prediction directly from the inaccurate and low-resolution (due to engineering constraints on runtime loading and memory bottleneck for the sufficiently dense pre-computed query point occupancy labels) TSDF voxels, the prediction result (DISN + TSDFVox) is clearly inferior compared to our results (DGS). The surface color denotes the evaluation of the "precision", with the larger blue region, the higher "precision".

2 Related Works

3D Implicit Representations Among all the 3D representations, implicit models are advantageous for arbitrarily high resolution modeling (unlike voxels with fixed resolution and no detailed surface modeling) and easy learning (unlike mesh with varying topological structures). Implicit models typically learn to establish a mapping between a query point and the prediction of the point. Groueix et al. proposed to learn a mapping from the uv texture map to the 3D surface point. More recent works (Park et al., 2019; Mescheder et al., 2019; Xu et al., 2019; Popov et al., 2020; Chen & Zhang, 2019a) focus on mapping the query point coordinates to the signed distance field or the occupancy field. In addition to these pure geometry modeling, recent works like (Saito et al., 2019) also model the surface texture via learning a mapping from the surface point to the RGB value, or use the RGB loss as the supervison signal (Niemeyer et al., 2020; Yariv et al., 2020). In contrast to defining textures explicitly on the surface, Mildenhall et al. (2020) proposed the volumetric rendering representation which maps the point coordinate and the viewing angle to volumetric textures. Oechsle et al. (2021); Wang et al. (2021); Yariv et al. (2021) use volume renderings for fitting high quality geometry of single scenes, but these are not generalizable to unseen scenes. In multi-view stereo setting, Murez et al. (2020); Sun et al. (2021); Božič et al. (2021) proposed to accumulate the multi-view predictions in a TSDF voxel volume. We believe our proposed DGS module is necessary when extending most of the existing implicit surface representations into feed-forward models that are conditioned on input image(s), and training these models with a loss on the gradient of the predicted field. In this work, we focus on pure geometrical reconstructions from single view, as an example of the application of the proposed DGS module.

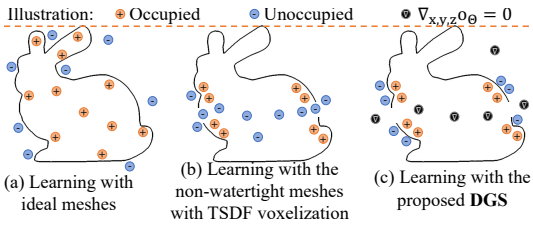

Figure 3: Illustration of the loss imposition for the occupancy prediction scenario. (a) When learning from the ideal mesh for ShapeNet objects, we can directly supervise the training with the accurate occupancy labels. (b) When extending to the real scenes (Fig. 2(b)), the TSDF voxelization demonstrates severe noise for training. Specifically, a considerable fraction of the objects are "empty" inside. (c) Our learning scheme with DGS alleviates these issues via enabling imposition of losses on the gradients.

Differentiable Operations End-to-end deep learning requires all modules in a computation path to be differentiable. Many differentiable modules are proposed to serve a particular functionality. The spatial transformer layer (Jaderberg et al., 2015) is one of the early attempts for differentiably sam-

pling the feature map. Kato et al. (2018) proposed to render the mesh into image in a differentiable manner so that we can optimize the mesh or the camera parameters to fit to a particular image. A similar idea has also been applied for rendering point clouds (Wiles et al., 2020; Yifan et al., 2019). Tulsiani et al. (2017) and Mildenhall et al. (2020) proposed to compute the ray rendering in an accumulating manner. One common feature of these differentiable operations is a strategy to soften the existing explicit modeling (Liu et al., 2019), and replace the gradient of the loss with approximate signals, making sure those signals provide guidance in the right direction. Our differentiable gradient sampler also seeks to soften the spatial gradient between the adjacent pixels.

3 LEARNING FRAMEWORK

Problem definition and notations. We aim to train a feed-forward deep model for predicting the 3D implicit surface reconstruction conditioned on a single image. We denote the input RGB image as $I \in \mathbb{R}^{m \times n \times 3}$. Like most feed-forward models, our model contains an image encoder. We denote the extracted 2D features as ϕ. Our model $\hat{f}_\Theta(\cdot)$, parameterized by Θ, takes in the image feature ϕ and predicts for each 3D location (x, y, z) the implicit value $\hat{f}_\Theta(\phi; x, y, z)$. We denote the ground truth value as $\tilde{f}(\phi; x, y, z)$. For the occupancy field, the implicit field value f represents the occupancy probability. We denote the predicted occupancy probability as $\hat{o}_\Theta(\phi; x, y, z) \in [0, 1]$ and the ground truth binary occupancy label as $\tilde{o}(\phi; x, y, z) \in \{“-”, “+”\}$, where “$-$” represents unoccupied space and “$+$” vice versa. For SDF, the implicit field value f represents the signed distance between the query point and its projection on the surface. We denote the predicted and the ground truth signed distance as $\hat{s}_\Theta(\phi; x, y, z) \in \mathbb{R}$ and $\tilde{s}(\phi; x, y, z) \in \mathbb{R}$ respectively.

Background. Typical fully supervised training (Xu et al., 2019; Popov et al., 2020) imposes the loss to each individual sampled query points by comparing the predicted field value $\hat{f}_\Theta(\phi; x, y, z)$ with the ground truth value $\tilde{f}(\phi; x, y, z)$. For occupancy predictions, we compute the loss as $\sum_{\{I,\mathcal{P}_I\}\in\mathcal{D}} \sum_{(x,y,z)\in\mathcal{P}_I} \text{BCE}(\hat{o}_\Theta(\phi; x, y, z), \tilde{o}_\Theta(\phi; x, y, z))$ (Fig. 3(a)), where “BCE” represents the binary cross entropy loss, $\mathcal{D}$ represents the whole single-view image training set, and $\mathcal{P}_I$ represents the set of all the possible query point sampling locations (x, y, z) within the view frustum. For SDF, we compute the loss as $\sum_{\{I,\mathcal{P}_I\}\in\mathcal{D}} \sum_{(x,y,z)\in\mathcal{P}_I} |\hat{s}_\Theta(\phi; x, y, z) - \tilde{s}(\phi; x, y, z)|$.

3.1 PROPOSED FRAMEWORK

As obtaining the full labels $\tilde{f}(\phi; x, y, z)$ for query points (x, y, z) from *all* the locations $\mathcal{P}_I$ is non-trivial for complex geometry from real-world (Fig. 2) and due to additional engineering constraints on runtime loading and memory bottleneck for the sufficiently dense pre-computed query point occupancy labels, we propose to incorporate the loss w.r.t. the spatial gradients. For occupancy,

$$\begin{aligned}\mathcal{L} = \sum_{\{I,\mathcal{P}_I\}\in\mathcal{D}} \Big(& \sum_{(x,y,z)\in\mathcal{P}_I - \mathcal{P}_I^0} \lambda_{or} \|\nabla_{x,y,z} \hat{o}_\Theta(\phi; x, y, z)\| \\ + & \sum_{(x,y,z)\in\mathcal{P}_I^{0+}} \text{BCE}(\hat{o}_\Theta(\phi; x, y, z), “+”) + \sum_{(x,y,z)\in\mathcal{P}_I^{0-}} \text{BCE}(\hat{o}_\Theta(\phi; x, y, z), “-”)\Big)\end{aligned} \tag{1}$$

where $\mathcal{P}_I^{0+}$ and $\mathcal{P}_I^{0-}$ denotes inward and outward near-surface query points, and λ_{or} represents the loss weight of the occupancy geometric regularization. The facing direction of the mesh surface (determining inward or outward) can be determined by normals (if available), or via rendering the surface in all views as in the RGBD captures (Dai et al., 2017) (with the surface facing toward camera as the “outward” side). The three terms in Eq. 1 correspond to the three types of losses in Fig. 3(c) (“∇”, “$+$”, “$-$”) respectively. For SDF, we set the loss similar to Gropp et al. (2020):

$$\begin{aligned}\mathcal{L} = \sum_{\{I,\mathcal{P}_I\}\in\mathcal{D}} \Big(& \sum_{(x,y,z)\in\mathcal{P}_I} \lambda_{sr} |\|\nabla_{x,y,z} \hat{s}_\Theta(\phi; x, y, z)\|_2 - 1| + \sum_{(x,y,z)\in\mathcal{P}_I^0} |\hat{s}_\Theta(\phi; x, y, z)| \\ & + \sum_{(x,y,z)\in\mathcal{P}_I^0} \lambda_{sn} \|\nabla_{x,y,z} \hat{s}_\Theta(\phi; x, y, z) - \nabla_{x,y,z} \tilde{s}(\phi; x, y, z)\|\Big)\end{aligned} \tag{2}$$

where $\mathcal{P}_I^0$ denotes the query points on the ground truth surface only, and λ_{sr} and λ_{sn} represents the loss weight for the signed distance geometric regularization term and normal term respectively. The three terms represents the Eikonal regularization (Crandall & Lions, 1983), surface zero-SDF loss, and the surface normal loss respectively, with the last term optional (Gropp et al., 2020). Note now

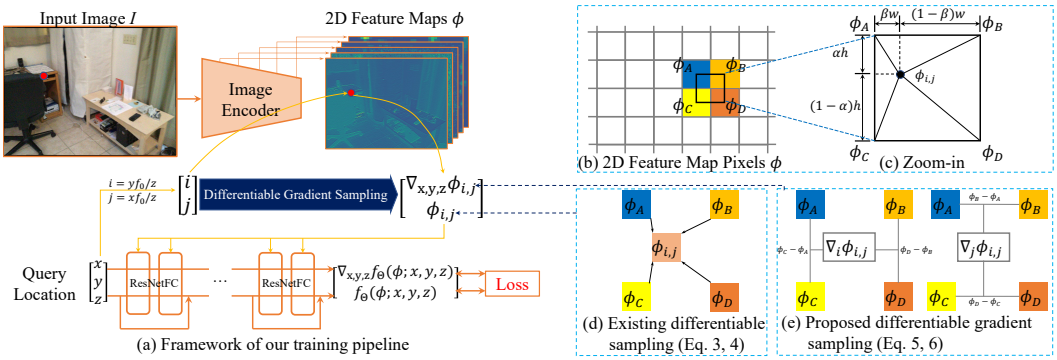

Figure 4: Overview of our learning framework (a) and differentiable gradient sampling (b, c, d, e).

the loss functions in Eq. 1, 2 no longer require the implicit ground truth label for query points far away from the ground truth surface, enabling training with the real-world imperfect scanned data. In both cases, the availability of surface normals is optional.

Practically, we found only our proposed loss for the occupancy version (Fig. 3(c), Eq. 1) demonstrates capability to be applicable for the real-world single-view feed-forward scenario, and use Eq. 1 rather than Eq. 2 in all of our experiments. First, learning with the Eikonal term (Eq. 2) suffers severely from its sensitivity of model initialization. We found either the SDF or the Truncated SDF (TSDF) representation model cannot easily converge during training. Specifically, we found it poses numerical difficulties when the model is learned to predict a large distance value or a constant truncation value for the majority of the query points in the air that is far from any surfaces. This becomes a major problem when we extend from the single objects scenario (Mescheder et al., 2019; Popov et al., 2020) or single scene fitting (Gropp et al., 2020). Second, our loss function (Eq. 1) can significantly save memory footprint during training and enable large batch size training, which is crucial in our learning framework. Unlike in Eq. 2 where all the query points require spatial gradient computation, which leads to $3\times$ higher of memory footprint, in our loss function (Eq. 1), only the non-surface query points do. We set the query point batch size of the non-surface points (512 in practice) to be much smaller than the critical near-surface points (4096 in practice), enabling learning with the image batch size of 32 in our real-scene training.

3.2 Sampling with Differentiable Gradients

A distinct difference compared to existing works is that our spatial gradient $\nabla_{x,y,z}\hat{f}_\Theta(\phi; x, y, z)$ is also conditioned on the pixels in ϕ. This is in stark contrast to existing works. Gropp et al. (2020) devised the model f_Θ to fit to a single scene, and the spatial gradient $\nabla_{x,y,z}\hat{f}_\Theta(x, y, z)$ can be conveniently computed because it does not involve the sampling procedure. Jiang et al. (2020); Lin et al. (2020) used a non-spatial global feature for inference and hence bypassed sampling. In our learning framework, the spatial gradient computation must undergo the sampling procedure.

We name our gradient computation involving the sampling operation as the *Pixel Conditioned Gradients* and derive a closed-form solution, *Differentiable Gradient Sampling* (DGS), for handling forward and backward propagation. Figure 4(a) provides an illustration of our training pipeline. Each layer in our network tracks both the response of the layer as well as its spatial gradient w.r.t. (x, y, z). While it is well established to track the layer-wise spatial gradient for fully-connected (FC) layers or convolutions in existing works (Gropp et al., 2020), tracking the spatial gradients and back-propagating the loss to the feature map pixels ϕ through the sampling module has not been studied. To this end, we derive the closed-form sampling scheme for tracking and propagating the spatial gradients $\frac{\partial \phi}{\partial x,y,z}$ through the sampling layer.

Background - 2D Differentiable Sampling. Differentiably sampling pixel values from a grid of 2D feature map with the given pixel locations (i, j) is a common operation. Throughout our paper, we define the pixel coordinates in the Normalized Device Coordinate (NDC) system that ranges from -1 to 1. As illustrated in Fig. 4(d), given the feature map ϕ and the sampling locations (i, j), the resulting sampled value $\phi(i, j)$ is

$$\phi(i, j) = (1 - \alpha)((1 - \beta)\phi_A + \beta\phi_B) + \alpha((1 - \beta)\phi_C + \beta\phi_D) \tag{3}$$

Please refer to Fig. 4(c) for the definitions of α, β and $\phi_A, \phi_B, \phi_C, \phi_D$. Without loss of generality we use bilinear interpolation. During training, the gradient from the loss can be back-propagated via

$$\frac{\partial L}{\partial \phi_A} = (1-\alpha)(1-\beta)\frac{\partial L}{\partial \phi(i,j)}. \tag{4}$$

2D Differentiable Gradient Sampling. Our learning framework (Sec. 3.1, Fig. 4(a)) requires the extension of the sampling capability from just the feature value response $\phi_{i,j}$ to its spatial gradient $\nabla_{i,j}\phi(i,j)$. During the forward and the backward propagation, both the sampled feature response $\phi_{i,j}$ and its spatial gradient $\nabla_{i,j}\phi(i,j)$ are recorded for further propagation (Fig. 4(e)):

$$\frac{\partial \phi(i,j)}{\partial i} = \frac{(1-\beta)(\phi_C - \phi_A) + \beta(\phi_D - \phi_B)}{h}. \tag{5}$$

Hence, during the forward pass, we compute the spatial gradient via Eq. 5 in addition to the existing value sampling (Eq. 3). During the backward pass, we compute the loss gradient over the spatial gradient via

$$\frac{\partial L}{\partial \phi_A} = (1-\alpha)(1-\beta)\frac{\partial L}{\partial \phi(i,j)} + (-\frac{1-\beta}{h})\frac{\partial L}{\partial \frac{\partial \phi(i,j)}{\partial i}} + (-\frac{1-\alpha}{w})\frac{\partial L}{\partial \frac{\partial \phi(i,j)}{\partial j}}, \tag{6}$$

where w and h are the width and height of a pixel in NDC, s.t. $w = 2/W$ and $h = 2/H$ for the feature map with the size $W \times H$.

3D Differentiable Gradient Sampling. We now extend the sampling to 3D. We model the camera as the pin-hole camera. For any point (x, y, z) in the camera space, we seek for its projected 2D locations (i, j) based on the focal length f_0 via $i = yf_0/z, j = xf_0/z$. The feed forward pass is

$$\begin{aligned}
\frac{\partial \phi(x,y,z)}{\partial x} &= \frac{\partial \phi(i,j)}{\partial i}\cdot\frac{\partial i}{\partial x} + \frac{\partial \phi(i,j)}{\partial j}\cdot\frac{\partial j}{\partial x} = \frac{\partial \phi(i,j)}{\partial j}\cdot\frac{f_0}{z},\\
\frac{\partial \phi(x,y,z)}{\partial y} &= \frac{\partial \phi(i,j)}{\partial i}\cdot\frac{\partial i}{\partial y} + \frac{\partial \phi(i,j)}{\partial j}\cdot\frac{\partial j}{\partial y} = \frac{\partial \phi(i,j)}{\partial i}\cdot\frac{f_0}{z},\\
\frac{\partial \phi(x,y,z)}{\partial z} &= \frac{\partial \phi(i,j)}{\partial i}\cdot\frac{\partial i}{\partial z} + \frac{\partial \phi(i,j)}{\partial j}\cdot\frac{\partial j}{\partial z} = \frac{\partial \phi(i,j)}{\partial i}\cdot(-\frac{yf_0}{z^2}) + \frac{\partial \phi(i,j)}{\partial j}\cdot(-\frac{xf_0}{z^2}).
\end{aligned} \tag{7}$$

During the backward propagation procedure, the DGS accumulates the gradient via

$$\begin{aligned}
\frac{\partial L}{\partial \phi_A} =& \frac{\partial L}{\partial \phi(x,y,z)}\cdot(1-\alpha)(1-\beta) + \frac{\partial L}{\partial \frac{\partial \phi(x,y,z)}{\partial x}}\cdot(-\frac{1-\alpha}{w})\frac{f_0}{z} + \frac{\partial L}{\partial \frac{\partial \phi(x,y,z)}{\partial y}}\cdot(-\frac{1-\beta}{h})\frac{f_0}{z}\\
&+ \frac{\partial L}{\partial \frac{\partial \phi(x,y,z)}{\partial z}}\cdot((\frac{1-\beta}{h})(\frac{yf_0}{z^2}) + (\frac{1-\alpha}{w})(\frac{xf_0}{z^2})).
\end{aligned} \tag{8}$$

4 EXPERIMENTS

4.1 LEARNING FROM SYNTHETIC DATA (SHAPENET)

Dataset. Following Popov et al. (2020), we conduct shapeNet experiments with the low-realism and the high-realsim renderings. For low-realism, we use the renderings of various models from Choy et al. (2016). Following Mescheder et al. (2019), we split the dataset into the training set of 30661 models, the validation set of 4371 models, and the test set of 8751 models. During testing, we follow Mescheder et al. (2019) to only test the first rendering of each model. Similar to all the prior works, we use the same 13 classes to report any performance via grouping all the test cases into the same class. For high-realism, we follow the split and evaluation protocol of the single-object scenes Popov et al. (2020) - we only use the 13 categories used by Choy et al. (2016), filter out repeated samples, and finally construct the data with 666,565 models from the training set, 96,084 from the validation set, and 189,748 from the test set. Following Popov et al. (2020), we only evaluate the first 1% test cases of the test set (1898 samples).

Comparison with state-of-the-art approaches. We choose to compare to recent representative state-of-the-art approaches, including OccNet (Mescheder et al., 2019), DISN (Xu et al., 2019),

Table 1: Intersection over Union % (IoU ↑) benchmarking result on the **high-realism** ShapeNet. Once again, our approach advances **state-of-the-art** approaches (re-trained) even given only near-surface occupancy labels. We emphasize that our comparison (**DGS**) to CoReNet is a direct ablation as we maintain exactly the same experimental setups in CoReNet except for the loss function.

Category	Craft	Rifle	Disp.	Lamp	Spk.	Box	Chair	Bench	Car	Plane	Sofa	Table	Phone	Mean
OccNet	54.7	48.7	55.5	38.6	57.1	52.0	48.9	39.7	70.6	48.9	63.6	49.2	71.7	53.8
DISN	60.9	63.5	56.9	46.4	61.5	55.2	57.4	46.7	72.5	57.2	68.2	55.7	74.5	59.7
CoReNet	62.5	65.5	63.2	48.2	**63.0**	56.7	60.6	48.3	**73.7**	58.1	69.8	55.0	75.1	61.5
D^2IM-Net	61.7	64.3	60.4	48.7	62.0	54.8	59.4	48.2	73.2	59.1	69.4	**57.2**	77.5	61.2
Abl.-NoGrad	14.1	6.1	27.3	18.1	37.3	40.4	28.7	13.0	26.2	9.6	30.5	23.4	25.1	23.1
Abl.-NoGrad (10 %)	28.2	25.8	39.3	32.4	48.2	46.5	42.0	24.6	42.4	20.0	42.7	37.1	49.4	36.8
Abl.-NoGrad (30 %)	34.1	25.1	46.6	30.8	49.8	47.9	45.1	28.1	51.4	26.5	54.9	38.0	52.7	40.8
Abl.-NoGrad (50 %)	47.8	41.5	47.6	35.1	54.1	49.2	46.8	32.1	66.0	37.9	59.6	44.4	60.2	47.9
Abl.-FixedE	52.6	57.0	47.8	37.4	52.6	49.7	47.4	33.0	69.2	51.0	60.2	42.9	69.2	51.5
OccNet w/ Eq. 1	56.8	52.6	56.8	39.3	58.1	53.6	51.1	42.4	71.7	49.7	65.5	50.9	76.0	55.7
DISN w/ Eq. 1	60.4	63.9	59.6	46.2	61.1	55.8	58.3	46.6	73.2	57.3	68.7	55.5	76.7	60.2
DGS	**63.3**	**70.4**	**65.7**	**49.0**	61.2	**57.0**	**62.1**	**50.9**	70.1	**62.8**	**70.3**	56.9	**78.3**	**62.9**

CoReNet (Popov et al., 2020) and D^2IM-Net (Li & Zhang, 2021). Quantitative results are reported in Tab. 3 for the low-realism evaluation setting, and in Tab. 1 for the high-realism setting, where our results outperform all the state-of-the-art results in all the settings. Please refer to Sec. A for qualitative comparison, implementation details and the evaluation metrics.

For our low-realism evaluation setting, all the baseline approaches reported their results in the papers. For OccNet and DISN, the reported results are based on knowing the category canonical view prior, which demonstrates considerable privilege with respect to accuracy Popov et al. (2020). Hence, we mark the results from the literature as OccNet-Privilege and DISN-Privilege (OccNet-Priv. and DISN-Priv. for short). Compared to their privileged setting, our results demonstrate superior results even without the category canonical view prior privilege. To further provide the the baseline results where these two approaches are without the category canonical view prior, we retrain their models with the released codes, and report the results in the "OccNet" and "DISN" row respectively. The results further provide evidence that the category canonical view prior demonstrates privilege on accuracy, as observed in Popov et al. (2020).

For our high-realism evaluating setting, since the results for Occnet and DISN are not directly available from the paper, we retrain their model from the released codes (without the category canonical view prior) on the high-realism data provided by Popov et al. (2020). We also retrain CoReNet using their released codes and obtain results better than reported in their original paper (61.5 as mIoU instead of 59.5). Our model still consistently outperforms all the baselines on almost all the categories. We emphasize that our comparison to CoReNet is a direct ablation comparison where we directly follow their experimental setup and model size. Furthermore, our approach only utilizes the close-to-surface occupancy labels, which is a subset of the whole occupancy labels for all the query point. Our model still outperforms the baseline and provides convincing evidence that our supervision loss function demonstrate better results than the original CoReNet approach. We attribute the superiority of our approach to our particular focus toward the difficult close-to-surface region similar to hard negative mining.

Ablation Study. We further conduct ablation studies to further validate the importance of derived DGS module, via attempting work-around training methods without DGS.

i) NoGrad - To test the performance of the baseline when only the surface data are provided, we train this ablation model in exactly the same way compared to its original model, with the only exceptions that only the near-surface points are equipped with training labels, and we mask out all the training labels from the non-surface query points (where it is not necessary to backpropagate the loss gradient of the spatial gradient using DGS). To further evaluate how the rate of known voxels affects the learning performance, we enlarge the near-surface region and evaluate when the rate is 10%, 30% and 50%. An increase in the performances among these baselines would indicate the importance of knowing more voxel labels if our proposed gradient loss (Eq. 1) is not imposed.

ii) FixedE - We train with both the near-surface as well as the spatial gradient supervision, without DGS - meaning the loss gradient of the spatial gradient would not back-propagate to any module before the sampling module - in our case, the feature encoder network. Note all the other losses without gradient sampling can still back-propagated to the feature maps.

We report the ablation results in Tab. 1. Both experiments are conducted with the high-realism ShapeNet data. Once again, we comfortably outperform all the baselines, validating the essential roles of DGS in our learning framework.

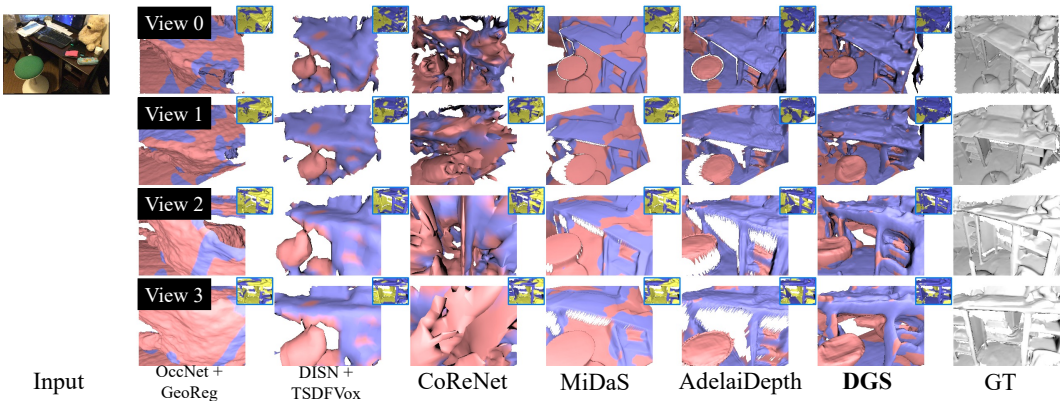

Figure 5: Qualitative comparisons on one challenging test case on ScannetV2. For each predicted surface with red and sky-blue colors, sky-blue indicates "positive precision" for that surface region, while red indicates "negative precision". For each ground truth surface on the top-right corner of each prediction with gold and navy-blue colors, navy-blue indicates "positive recall", while gold indicates "negative recall". The larger the blue region is, the higher F1 score would be.

Table 2: Benchmarking results of single view 3D surface reconstruction on ScannetV2 test set.

	Acc (↓)	Compl (↓)	Chamfer (↓)	Prec (↑)	Recall (↑)	F1-score (↑)
OccNet + GeoReg	18.3	18.5	18.4	30.4	28.5	26.8
DISN + TSDFVox	13.0	53.3	33.1	25.3	10.2	12.4
CoReNet	19.5	15.6	17.6	30.9	36.3	29.9
MiDaS	13.9	18.5	16.2	41.9	30.1	34.4
AdelaiDepth	**9.7**	18.8	14.2	**46.4**	33.2	37.9
Ablation-NoGrad	17.0	11.7	14.3	34.1	47.0	36.5
Ablation-FixedE	15.0	20.3	17.7	31.6	29.8	27.3
DGS	14.3	**11.5**	**12.9**	39.7	**49.4**	**41.6**

4.2 Learning from Real Scanned Datasets (ScannetV2)

Dataset. We use ScannetV2 (Dai et al., 2017) for training and evaluating the performance of the models on the real images. We follow the standard training / testing split as used in Sun et al. (2021) and Murez et al. (2020), where 1513 scenes are for trainval (with 1201 for training and 312 for validation), and 100 for testing. Each scene is provided with multiple image capture as well as the associated camera pose. We train the models with all the views given in the training / validation set (2423872 frames in total, after filtering out frames with invalid extrinsic poses), while for testing, we select 10 frames with different extrinsic poses for each test scene. Practically, since all the frames of the scenes are in the form of video clips, with adjacent frames associated with similar extrinsic cameras poses, we select the 10 frames for each frame via extracting every 100 frames from each scene video (e.g. Frame 1, 101, 201, ..., 901), resulting in 1000 frames in total in our test set (10 frames per scene, with 100 test scenes).

Metrics. We use the same evaluation metrics following Sun et al. (2021) and Murez et al. (2020). Since we are the first, to our knowledge, to evaluate single view 3D implicit reconstruction on the ScannetV2 benchmark, and here we only evaluate the geometries within the camera view frustum rather than the whole scene geometries as in Sun et al. (2021); Murez et al. (2020). In addition, we also only evaluate the geometries in front of the *amodal* depth for each pixel ray. We define the amodal depth for a pixel ray to be delineated by the minimum between the closest structure-category surface (e.g. walls and doors, etc) and the farthest surface. In practice, in order to accommodate the evaluation of the surfaces right on the amodal depth, we slack the evaluation scope with a factor λ (1.05 in our case) multiplied with the amodal depth. This evaluation protocol would be equivalent to the "single-layered" protocol used in Sun et al. (2021), within our single-view scenario.

Due to the inherent ambiguity of the scaling and shifting of the predicted 3D single view geometry (Ranftl et al., 2019; Yin et al., 2021), we follow them by regressing the best scale and shift comparing the predicted depth with the ground truth depth. For approaches that predict 3D surfaces, the rendered depth map from the predicted mesh would be used for calculating the scale and shift.

Baselines. Since we demonstrate the first attempt to conduct the implicit reconstruction of single-view scenes, very few exiting works provide a direct baseline performance to our task. For OccNet (Mescheder et al., 2019), since its image feature extraction is not local, and its gradient propa-

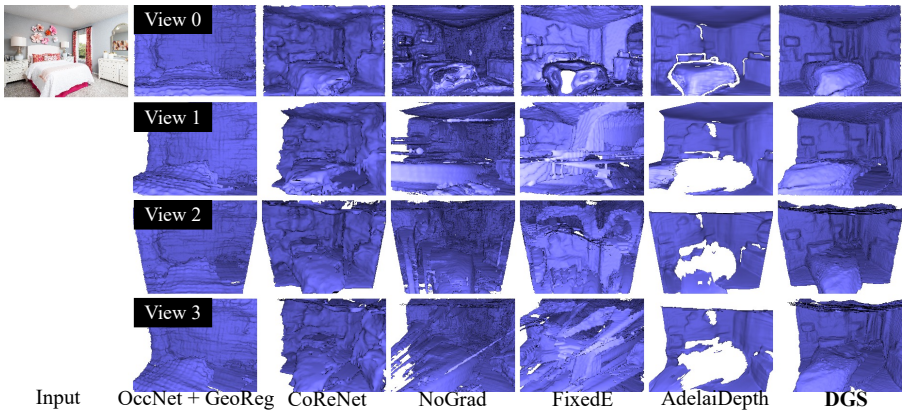

Figure 6: Qualitative comparisons on an unseen test image downloaded from the internet.

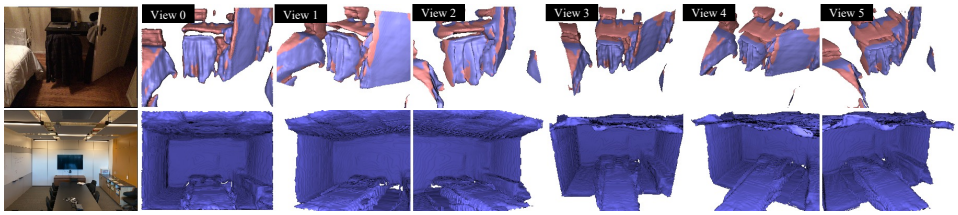

Figure 7: Two representative failure cases of our approach.

gation does not require sampling, we use its direct application with Gropp et al. (2020) "OccNet + GeoReg" as one baseline. We train DISN (Xu et al., 2019) with the TSDF voxelization labels (Murez et al., 2020; Sun et al., 2021; Božič et al., 2021). For CoReNet (Popov et al., 2020), we stick to its own voxelization and internal filling toolbox for obtaining the training labels. Since CoReNet can only predict geometries within the fixed range of space, we tried our best to pick the best cube location based on the dataset statistics. We also incorporate depth approaches (Ranftl et al., 2019; Yin et al., 2021) for comparison by finetuning their weights on ScannetV2. Lastly, we compare with the two ablation models *NoGrad* and *FixedE* as introduced in Sec. 4.1.

Results. We provide quantitative comparison in Tab. 2 and qualitative comparison in Fig. 5 respectively. Our model outperforms all state-of-the-art approaches as well as the ablation models. Compared to the synthetic data scenario in Tab. 3 and 1, our model demonstrates even larger advantages when compared to existing state-of-the-art approaches, as our motivation stems from addressing learning from imperfect 3D labels directly from the real scan data. Compared to single-view depth prediction approaches trained on massive data (Lasinger et al., 2019; Yin et al., 2021), our approach does not prevail on "Acc" and "Prec". This is due to the fact that these two metrics only project the predicted surface to the ground truth surface, giving advantages to approaches that only predict the visible surface. Our approach still prevails for other metrics (Chamfer Distance and F1), which are considered as the most important metrics (Božič et al., 2021; Sun et al., 2021; Murez et al., 2020).

Generalization to Unseen Scenes (Pix3d and Open-Domain Images). We further test our model without further finetuning directly on unseen scenes for evaluating the generalizability of the learned model. We tried our model on Pix3d Sun et al. (2018) as well as test images downloaded directly from the internet. We provide a detailed qualitative comparison in Fig. 6 and more results in Fig. 1. The results further indicate our learning framework exhibits promise for unseen scene generalization.

Failure Cases. We provide two representative failure cases of our approach in Fig. 7. The first case (the first row) demonstrates difficulties in predicting the floor occupancy as a result of the noisy and non-watertight mesh during training. The second case (the second row) shows that our model cannot identify objects (e.g. chairs) and well predicts the invisible part of them.

5 Conclusions

We have presented our learning framework for real-world 3D implicit surface reconstruction from a single view image. Owing to our unique learning framework that directly trains the model from the raw scan data and our novel occupancy loss function over the gradients, we are able to go beyond the existing works for single-objects reconstruction or single view fitting. Thanks to our differentiable gradient sampling module we enable efficient and memory-economic end-to-end training and achieved scene-level single view reconstruction for the first time, to our knowledge.

REFERENCES

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pp. 265–283, 2016.

Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2565–2574, 2020.

Aljaž Božič, Pablo Palafox, Justus Thies, Angela Dai, and Matthias Nießner. Transformerfusion: Monocular rgb scene reconstruction using transformers. *arXiv preprint arXiv:2107.02191*, 2021.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5939–5948, 2019a.

Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2019b.

Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pp. 628–644. Springer, 2016.

Michael G Crandall and Pierre-Louis Lions. Viscosity solutions of hamilton-jacobi equations. *Transactions of the American mathematical society*, 277(1):1–42, 1983.

Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 303–312, 1996.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5828–5839, 2017.

Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 605–613, 2017.

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020.

T Groueix, M Fisher, VG Kim, BC Russell, and M Aubry. Atlasnet: a papier-mâché approach to learning 3d surface generation (2018). *arXiv preprint arXiv:1802.05384*, 11.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. 2016.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28:2017–2025, 2015.

Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1251–1261, 2020.

Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3907–3916, 2018.

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

Katrin Lasinger, René Ranftl, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*, 2019.

Manyi Li and Hao Zhang. D2im-net: Learning detail disentangled implicit fields from single images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10246–10255, 2021.

Chen-Hsuan Lin, Chaoyang Wang, and Simon Lucey. Sdf-srn: Learning signed distance 3d object reconstruction from static images. *arXiv preprint arXiv:2010.10505*, 2020.

Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019.

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

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4460–4470, 2019.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020.

Zak Murez, Tarrence van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pp. 414–431. Springer, 2020.

Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3504–3515, 2020.

Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. *arXiv preprint arXiv:2104.10078*, 2021.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 165–174, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pp. 8026–8037, 2019.

Stefan Popov, Pablo Bauszat, and Vittorio Ferrari. Corenet: Coherent 3d scene reconstruction from a single rgb image. In *European Conference on Computer Vision*, pp. 366–383. Springer, 2020.

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*, 2019.

Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2304–2314, 2019.

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *arXiv preprint arXiv:1906.01618*, 2019.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020.

Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1746–1754, 2017.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15598–15607, 2021.

Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2974–2983, 2018.

Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2626–2634, 2017.

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. *arXiv preprint arXiv:2106.10689*, 2021.

Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7467–7477, 2020.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems*, pp. 490–500, 2019.

Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *arXiv preprint arXiv:2003.09852*, 2020.

Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *arXiv preprint arXiv:2106.12052*, 2021.

Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics (TOG)*, 38(6):1–14, 2019.

Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 204–213, 2021.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. *arXiv preprint arXiv:2012.02190*, 2020.

Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. *arXiv preprint arXiv:2103.14024*, 2021.

A ADDITIONAL DETAILS AND RESULTS ON SHAPENET

Metrics. We use the Intersection-of-Union (IoU) for evaluating the model performances. For low-realism renderings, we follow the protocol of IoU benchmarking from Mescheder et al. (2019) that evaluating occupancy of the specified 100K query points in the space. For high-realism renderings, we follow the protocol from Popov et al. (2020) to evaluate occupancy prediction of the $128 \times 128 \times 128$ grid. Following Popov et al. (2020), the grid cube is a unit cube that spans from -0.5 to 0.5 for x and y, and from $f/2$ to $1 + f/2$ for z in the camera coordinate system.

Implementation details of our model (DGS). We build our model on top of one representative state-of-the-art model - CoReNet (Popov et al., 2020) for its 3D volumetric implicit modeling capability. This is because the space of the geometry predictions are always within the unit cube volume. We maintain exactly the same encoder and decoder architecture (ResNet50 (He et al., 2016) and the ray traced skip connection respectively), optimization parameters (e.g. the learning rate and the epsilon of Adam Kingma & Ba (2014)) and the prediction format (per-query occupancy probability). We set λ_{or} to be 0.01 in our loss function (Eq. 1).

Qualitative Results. We provide qualitative results in Fig. 9-13 as the additional illustration of the performance of all the approaches on ShapeNet.

Table 3: Intersection over Union % (IoU ↑) benchmarking result on the **low-realism** ShapeNet. Our proposed DGS learning advances **state-of-the-art** approaches (OccNet, DISN, CoReNet and D^2IM-Net) compared to the reported performance from the literatures. Note that we outperform existing approaches (OccNet, DISN, CoReNet and D^2IM-Net) even when they possess extra advantages given full occupancy / SDF supervision for any query points, while our approach only allows occupancy label for near-surface points.

Category	Craft	Rifle	Disp.	Lamp	Spk.	Box	Chair	Bench	Car	Plane	Sofa	Table	Phone	Mean
OccNet-Priv.	53.0	47.4	47.1	37.1	64.7	**73.3**	50.1	48.5	73.7	57.1	68.0	50.6	72.0	57.1
DISN-Priv.	60.2	68.0	57.7	39.7	55.9	53.1	54.9	54.2	77.0	61.7	67.1	48.9	73.6	59.4
OccNet	49.9	48.0	55.4	39.5	57.0	43.8	58.5	45.1	54.0	45.8	68.0	50.7	68.3	52.6
DISN	49.0	44.4	55.5	39.0	**67.3**	71.7	49.0	41.2	64.7	45.0	66.4	50.7	70.5	55.0
CoReNet	54.0	64.6	57.2	42.1	60.8	50.9	**63.0**	50.8	57.3	53.0	**70.6**	55.5	73.1	57.9
D^2IM-Net	**63.4**	**68.1**	52.7	42.1	51.8	48.6	56.1	**55.0**	**79.8**	55.8	65.4	53.7	76.2	59.1
DGS	57.0	65.7	**58.6**	**45.5**	58.6	55.2	59.8	48.2	71.6	**56.8**	68.1	**55.6**	**78.4**	**60.0**

B ADDITIONAL RESULTS ON SCANNETV2

Implementation details for DGS. Since the DISN variant architectures (Xu et al., 2019; Niemeyer et al., 2020; Yu et al., 2020) demonstrate higher degress of flexibility of representing any point in the space, in contrast to 3D convolutions that are fixed for a particular range of space (Popov et al., 2020), we build our model based on the former architectures. Similar to Niemeyer et al. (2020); Yu et al. (2020), we use 5 residual blocks in the fully connected part with the first 3 blocks receiving the 2D features and its spatial gradients. Our image encoder uses the same RexNext101 architecture (Xie et al., 2017) as in Yin et al. (2021) and starts the training with the pre-trained weights. We use a batch size of 32 images with 2048 near-surface points and 512 non-surface points per image. Similar to the ShapeNet experiments, we set λ_{or} to be 0.01. We found in the real-world setting, this parameter setting is crucial for the convergence of the learning procedure.

Quantitative Results for 2.5D evaluation metrics. We provide the quantitative 2.5D evaluation results in Tab. 4. Our approach constantly outperforms all other implicit based baselines. While our approach falls short slightly when compared to the existing depth-based approaches, we claim that our approach is not directly trained with the massive depth training data. Our results indicate that our depth performance is still on par with the state-of-the-art depth prediciton approaches.

Additional qualitative results. We provide additional qualitative results on ScannetV2 Fig. 14-18.

Table 4: Benchmarking results of the depth metrics on ScannetV2.

	AbsRel (↓)	AbsDiff (↓)	SqRel (↓)	RMSE (↓)	LogRMSE (↓)	δ_1 (↑)	δ_2 (↑)	δ_3 (↑)
OccNet + GeoReg	13.5	23.7	7.2	33.2	17.3	82.2	95.5	99.0
DISN + TSDFVox	15.8	26.9	9.2	34.9	18.8	77.3	94.1	98.5
CoReNet	12.6	21.4	6.6	30.4	16.3	84.5	96.0	98.9
MiDaS	9.3	15.8	3.1	21.2	11.7	91.4	98.7	99.8
AdelaiDepth	6.1	10.5	1.9	15.7	8.6	95.3	99.1	99.8
Ablation-Fair	10.5	17.6	5.3	26.6	14.9	88.2	96.4	98.9
Ablation-FixedE	14.4	24.5	7.6	32.9	17.9	81.4	95.4	98.9
DGS	8.9	14.7	4.1	22.7	12.7	91.1	97.5	99.3

C Analysis with Numerical Gradient Approximation

As part of our proposed learning framework with the loss imposed on the spatial gradient, the differentiable gradient sampling module (Sec. 3.2) has a numerical alternative where we can perturb the query points to get its simulated numerical gradients. We further compare the closed-form performance with the numerical counterpart in Tab. 5. Please find qualitative comparison in Fig. 9-13.

We provide a closer look at our comparison to the simulated gradient baseline. We found in our experiments that the simulated gradient baseline demonstrates relatively severe convergence difficulties. As shown in Fig. 8, the simulated gradient model (red) does not observe loss drop in the first 10k training iterations. Despite its subsequent loss drop, which indicates the simulated gradient model can still learn to predict the geometry in moderate accuracy, its converged training loss is still higher than our DGS (blue). This aligns with our benchmarking evaluations in Tab. 5 that the closed-form variant (DGS) achieve better performance.

Table 5: Intersection over Union % (IoU ↑) benchmarking comparison between the closed-form solution and the numerical gradients on the **high-realism** ShapeNet.

Category	Craft	Rifle	Disp.	Lamp	Spk.	Box	Chair	Bench	Car	Plane	Sofa	Table	Phone	Mean
Numerical	62.4	68.3	63.7	46.2	60.6	54.9	61.0	49.0	**70.7**	61.1	69.2	55.6	77.2	61.5
Closed-form	**63.3**	**70.4**	**65.7**	**49.0**	**61.2**	**57.0**	**62.1**	**50.9**	70.1	**62.8**	**70.3**	**56.9**	**78.3**	**62.9**

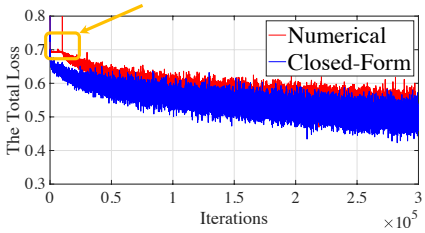

Figure 8: Convergence Analysis for the comparison between the closed-form (blue) and the numerical counterpart (red). Notably, the numerical counterpart does not observe loss drop in the first 10k iterations.

D Additional Generalizability Qualitative Results

We provide more qualitative results on our model generalizing to unseen images downloaded from the Internet in Fig. 19. Each result is visualized in 6 views. Once again, these additional visual results further indicate that our model demonstrates good generalizability capability for handling unseen indoor scene images.

E Evaluation of the Generalizability to Matterport3D dataset

To quantatively evaluate our generaizability to new datasets, we provide our results on the Matterport3D dataset (Chang et al., 2017) using our model trained from ScannetV2 as used in Sec. 4.2.

Table 6: Benchmarking results of single view 3D surface reconstruction on Matterport3D test set (trained by the ScannetV2 dataset).

	Acc (↓)	Compl (↓)	Chamfer (↓)	Prec (↑)	Recall (↑)	F1-score (↑)
OccNet + GeoReg	24.8	33.3	29.0	21.9	27.7	23.4
DISN + TSDFVox	32.7	25.6	29.1	19.1	33.3	22.9
CoReNet	35.2	23.3	29.3	20.8	35.5	24.9
AdelaiDepth	**14.1**	32.9	23.5	**33.7**	28.7	29.9
Ablation-NoGrad	39.5	**20.7**	30.1	22.9	**43.6**	28.7
Ablation-FixedE	22.7	34.7	28.7	19.5	24.0	20.2
DGS	24.4	22.1	**23.2**	25.7	41.8	**30.5**

The Matterport3d dataset (Chang et al., 2017) collects indoor scenes of 90 full houses. Textured meshes are provided along with the scanned real images. Since the Matterport3D dataset does not provide an official train/test split, and our model is not trained on Matterport3d, we evaluate our model over all the 90 scenes in the dataset. In particular, we use the 129600 images (without the elevated pitch views where the camera is looking up at the ceiling) from the dataset, use the 10 first images from each scene (resulting in 900 test images in total), and test their reconstruction performance in the same way as we tested on ScannetV2. We provide the quantative evaluation result in Tab. 6. We can see from the table that the results align with our evaluation in Tab. 2 that our model demonstrates clear advantages. We also noticed that the AdelaiDepth Yin et al. (2021) baseline (finetuned on ScannetV2) performance is closer to our results (F1) score compared to our ScannetV2 evaluation. This is probably due to the fact that AdelaiDepth was directly pre-trained from a massive number of training images and potentially gives it some privilege when generalize to a new dataset.

F Details of the formulation and derivation of DGS

In this section, we provide the full formulation as well as the derivation procedure of our DGS in this section as the supplementary to Sec. 3.2.

In Eq. 4, we provided the backward gradient for differentiable sampling for Pixel A. The backward gradient for Pixel B, C and D are

$$\begin{aligned}\frac{\partial L}{\partial \phi_B} &= (1-\alpha)\beta \cdot \frac{\partial L}{\partial \phi(i,j)} \\ \frac{\partial L}{\partial \phi_C} &= \alpha(1-\beta) \cdot \frac{\partial L}{\partial \phi(i,j)} \\ \frac{\partial L}{\partial \phi_D} &= \alpha\beta \cdot \frac{\partial L}{\partial \phi(i,j)}\end{aligned} \tag{9}$$

In Eq. 5, we provided the forward gradient in the vertical pixel direction. The forward gradient of the horizontal pixel direction is

$$\frac{\partial \phi(i,j)}{\partial j} = \frac{(1-\alpha)(\phi_B - \phi_A) + \alpha(\phi_D - \phi_C)}{w} \tag{10}$$

In Eq. 6, we provided the backward gradient for only Pixel A. The backward gradient for Pixel B, C and D are

$$\begin{aligned}\frac{\partial L}{\partial \phi_B} &= (1-\alpha)\beta \cdot \frac{\partial L}{\partial \phi(i,j)} + (-\frac{\beta}{h}) \cdot \frac{\partial L}{\partial \frac{\phi(i,j)}{\partial i}} + \frac{1-\alpha}{w} \cdot \frac{\partial L}{\partial \frac{\phi(i,j)}{\partial j}} \\ \frac{\partial L}{\partial \phi_C} &= \alpha(1-\beta) \cdot \frac{\partial L}{\partial \phi(i,j)} + \frac{1-\beta}{h} \cdot \frac{\partial L}{\partial \frac{\phi(i,j)}{\partial i}} + (-\frac{\alpha}{w}) \cdot \frac{\partial L}{\partial \frac{\phi(i,j)}{\partial j}} \\ \frac{\partial L}{\partial \phi_D} &= \alpha\beta \cdot \frac{\partial L}{\partial \phi(i,j)} + \frac{\beta}{h} \cdot \frac{\partial L}{\partial \frac{\phi(i,j)}{\partial i}} + \frac{\alpha}{w} \cdot \frac{\partial L}{\partial \frac{\phi(i,j)}{\partial j}}\end{aligned} \tag{11}$$

In Eq. 8, we provided the backward gradient for only Pixel A in the 3D setting. The backward gradient for Pixel B, C and D are

$$
\begin{aligned}
\frac{\partial L}{\partial \phi_B} =& \frac{\partial L}{\partial \phi(x,y,z)} \cdot (1-\alpha)\beta + \frac{\partial L}{\partial \frac{\partial \phi(x,y,z)}{\partial x}} \cdot \frac{1-\alpha}{w} \cdot \frac{f_0}{z} + \frac{\partial L}{\partial \frac{\partial \phi(x,y,z)}{\partial y}} \cdot (-\frac{\beta}{h}) \cdot \frac{f_0}{z} \\
&+ \frac{\partial L}{\partial \frac{\partial \phi(x,y,z)}{\partial z}} \cdot (\frac{\beta}{h} \cdot \frac{y f_0}{z^2} - \frac{1-\alpha}{w} \cdot \frac{x f_0}{z^2}) \\
\frac{\partial L}{\partial \phi_C} =& \frac{\partial L}{\partial \phi(x,y,z)} \cdot \alpha(1-\beta) + \frac{\partial L}{\partial \frac{\partial \phi(x,y,z)}{\partial x}} \cdot (-\frac{\alpha}{w}) \cdot \frac{f_0}{z} + \frac{\partial L}{\partial \frac{\partial \phi(x,y,z)}{\partial y}} \cdot \frac{1-\beta}{h} \cdot \frac{f_0}{z} \\
&+ \frac{\partial L}{\partial \frac{\partial \phi(x,y,z)}{\partial z}} \cdot (-\frac{1-\beta}{h} \cdot \frac{y f_0}{z^2} + \frac{\alpha}{w} \cdot \frac{x f_0}{z^2}) \\
\frac{\partial L}{\partial \phi_D} =& \frac{\partial L}{\partial \phi(x,y,z)} \cdot \alpha\beta + \frac{\partial L}{\partial \frac{\partial \phi(x,y,z)}{\partial x}} \cdot \frac{\alpha}{w}) \cdot \frac{f_0}{z} + \frac{\partial L}{\partial \frac{\partial \phi(x,y,z)}{\partial y}} \cdot \frac{\beta}{h} \cdot \frac{f_0}{z} \\
&+ \frac{\partial L}{\partial \frac{\partial \phi(x,y,z)}{\partial z}} \cdot (-\frac{\beta}{h} \cdot \frac{y f_0}{z^2} - \frac{\alpha}{w} \cdot \frac{x f_0}{z^2})
\end{aligned}
\tag{12}
$$

As an illustration of the derivation for Eq. 8, we can obtain the backward gradient via

$$
\frac{\partial L}{\partial \phi_A} = \frac{\partial L}{\partial \phi} \cdot \frac{\partial \phi}{\partial \phi_A} + \frac{\partial L}{\partial \frac{\partial \phi}{\partial x}} \cdot \frac{\partial \frac{\partial \phi}{\partial x}}{\partial \phi_A} + \frac{\partial L}{\partial \frac{\partial \phi}{\partial y}} \cdot \frac{\partial \frac{\partial \phi}{\partial y}}{\partial \phi_A} + \frac{\partial L}{\partial \frac{\partial \phi}{\partial z}} \cdot \frac{\partial \frac{\partial \phi}{\partial z}}{\partial \phi_A} \tag{13}
$$

The four partial derivatives with respect to ϕ_A in Eq. 13 can be determined based on the sampling rules. For example, based on Eq. 3, we can quickly obtain the first derivative with respect to ϕ_A via

$$
\frac{\partial \phi}{\partial \phi_A} = (1-\alpha)(1-\beta) \tag{14}
$$

For the other three derivatives with respect to ϕ_A in Eq. 13, we can substitute Eq. 5, 10 into Eq. 7, and obtain them via

$$
\begin{aligned}
\frac{\partial \frac{\partial \phi}{\partial x}}{\partial \phi_A} &= -\frac{f_0}{z} \cdot \frac{1-\alpha}{w} \\
\frac{\partial \frac{\partial \phi}{\partial y}}{\partial \phi_A} &= -\frac{f_0}{z} \cdot \frac{1-\beta}{h} \\
\frac{\partial \frac{\partial \phi}{\partial z}}{\partial \phi_A} &= \frac{y f_0}{z^2} \cdot \frac{1-\beta}{h} + \frac{x f_0}{z^2} \cdot \frac{1-\alpha}{w}
\end{aligned}
\tag{15}
$$

We can then obtain Eq. 8 via substituting Eq. 14, 15 into Eq. 13.

Figure 9: Quantitative comparison on the high-realism ShapeNet (without handpick: test case number 0 and 100). The reconstruction result of each approach is visualized in six different views, with the first view the same as the camera view, the first three views the same elevation as the camera view, and the last three view horizontal view.

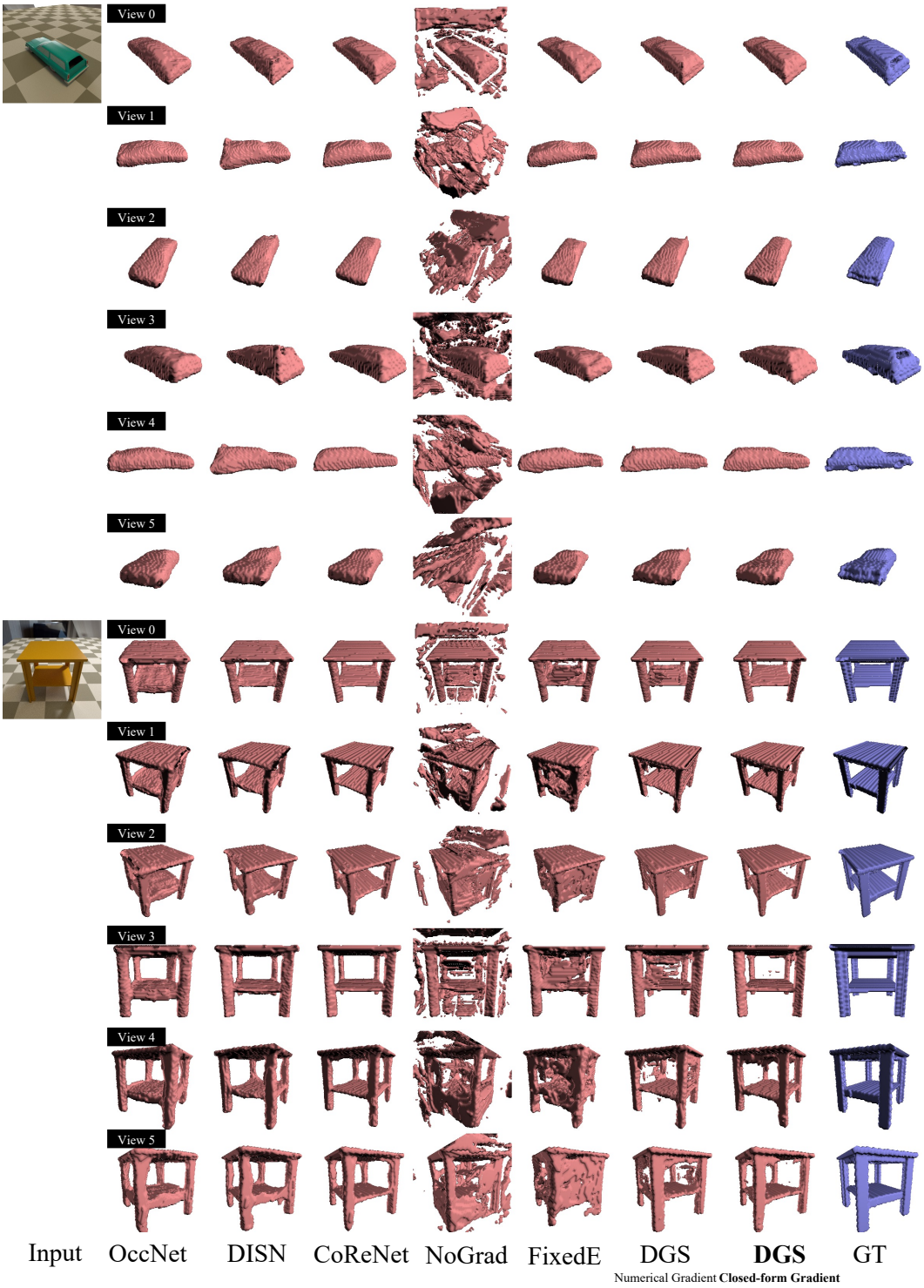

Figure 10: Quantitative comparison on the high-realism ShapeNet (without handpick: test case number 200 and 300). The reconstruction result of each approach is visualized in six different views, with the first view the same as the camera view, the first three views the same elevation as the camera view, and the last three view horizontal view.

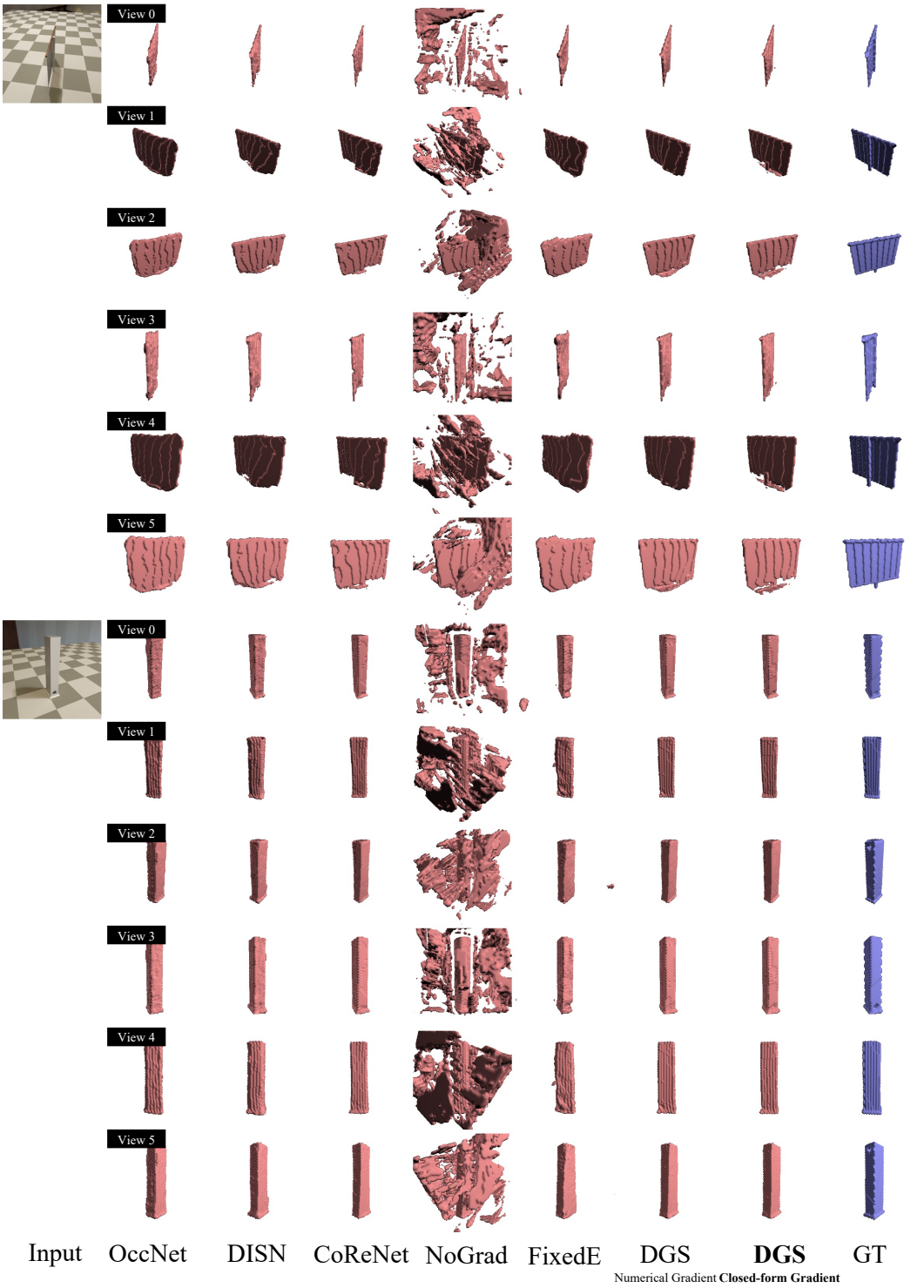

Figure 11: Quantitative comparison on the high-realism ShapeNet (without handpick: test case number 400 and 500). The reconstruction result of each approach is visualized in six different views, with the first view the same as the camera view, the first three views the same elevation as the camera view, and the last three view horizontal view.

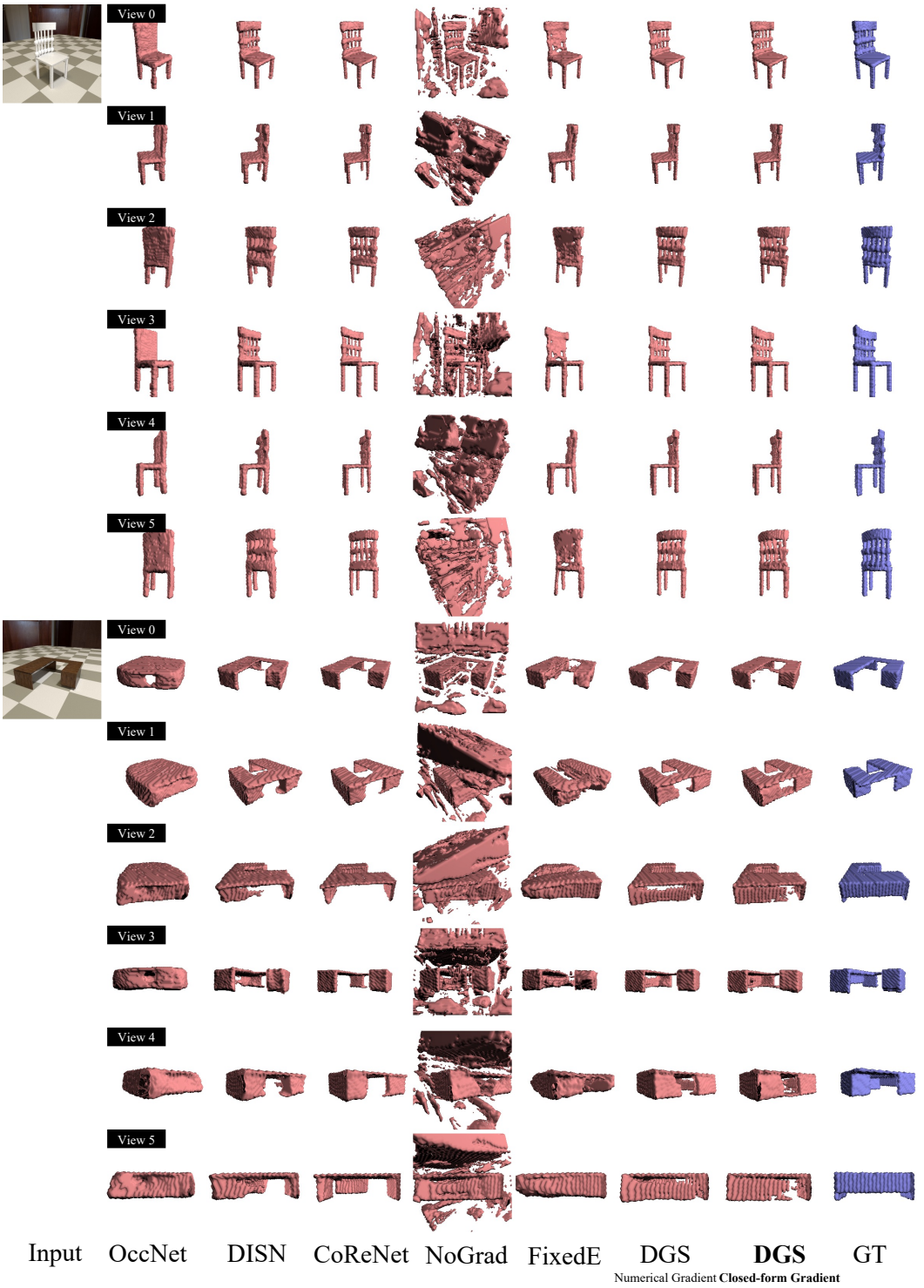

Figure 12: Quantitative comparison on the high-realism ShapeNet (without handpick: test case number 600 and 700). The reconstruction result of each approach is visualized in six different views, with the first view the same as the camera view, the first three views the same elevation as the camera view, and the last three view horizontal view.

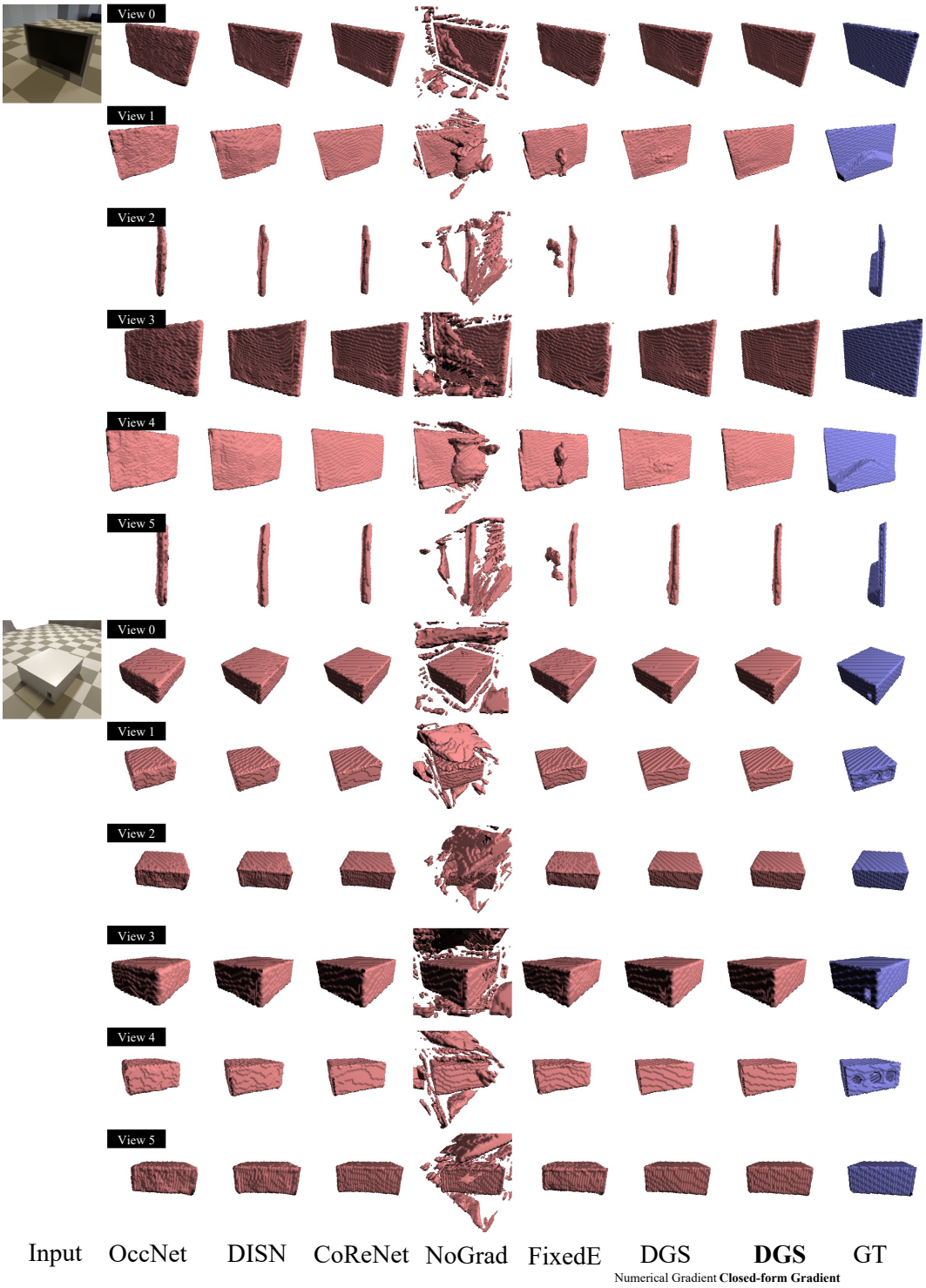

Figure 13: Quantitative comparison on the high-realism ShapeNet (without handpick: test case number 800 and 900). The reconstruction result of each approach is visualized in six different views, with the first view the same as the camera view, the first three views the same elevation as the camera view, and the last three view horizontal view.

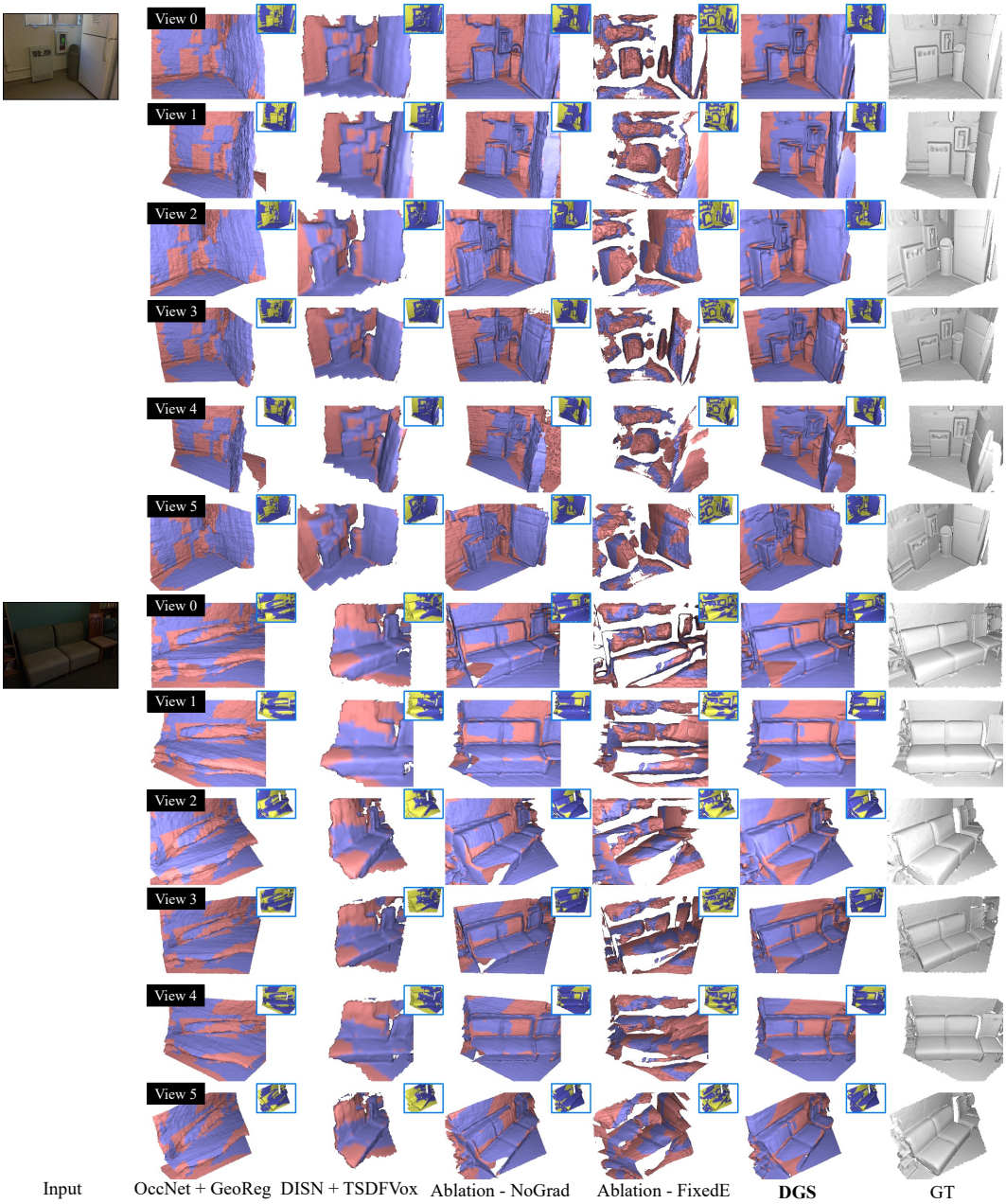

Figure 14: Quantitative comparison on the ScannetV2 (without handpick: the first frame of the 1st and 2nd test scene in ScannetV2). The reconstruction result of each approach is visualized in six different views, with the first view the same as the camera view, the first three views the same elevation as thecamera view, and the last three view elevated view.

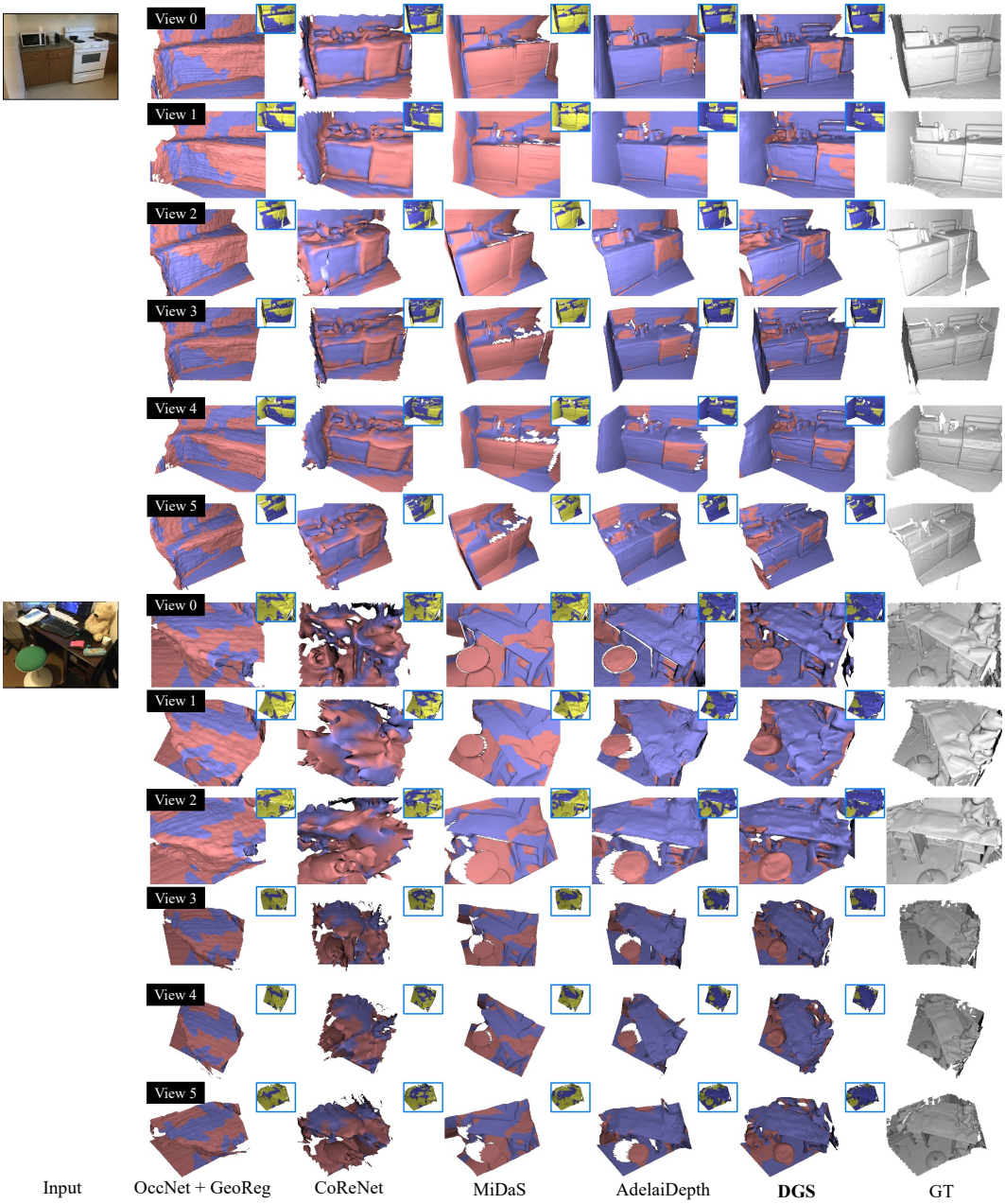

Figure 15: Quantitative comparison on the ScannetV2 (without handpick: the first frame of the 3rd and 4th test scene in ScannetV2). The reconstruction result of each approach is visualized in six different views, with the first view the same as the camera view, the first three views the same elevation as thecamera view, and the last three view elevated view.

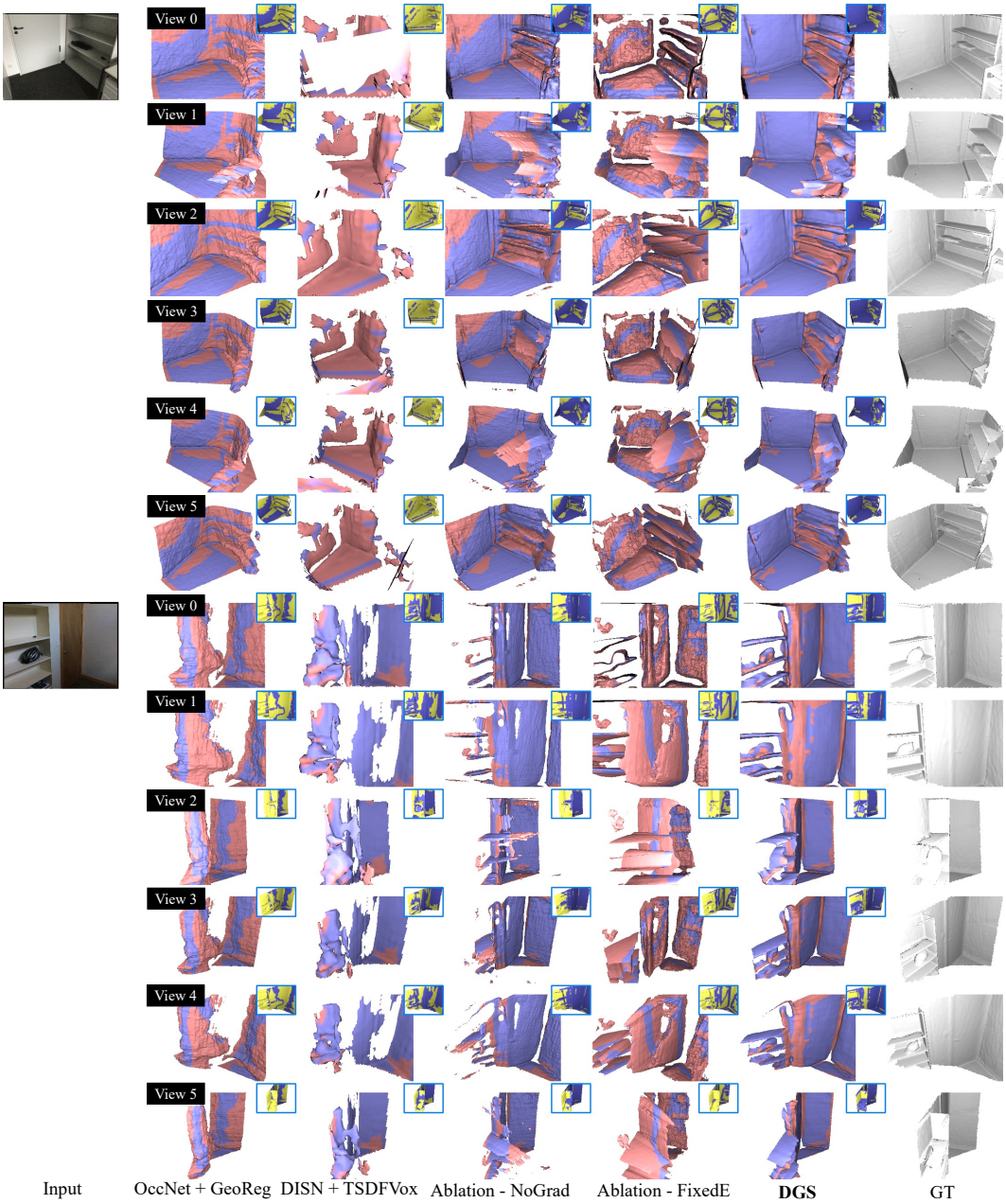

Figure 16: Quantitative comparison on the ScannetV2 (without handpick: the first frame of the 5th and 6th test scene in ScannetV2). The reconstruction result of each approach is visualized in six different views, with the first view the same as the camera view, the first three views the same elevation as thecamera view, and the last three view elevated view.

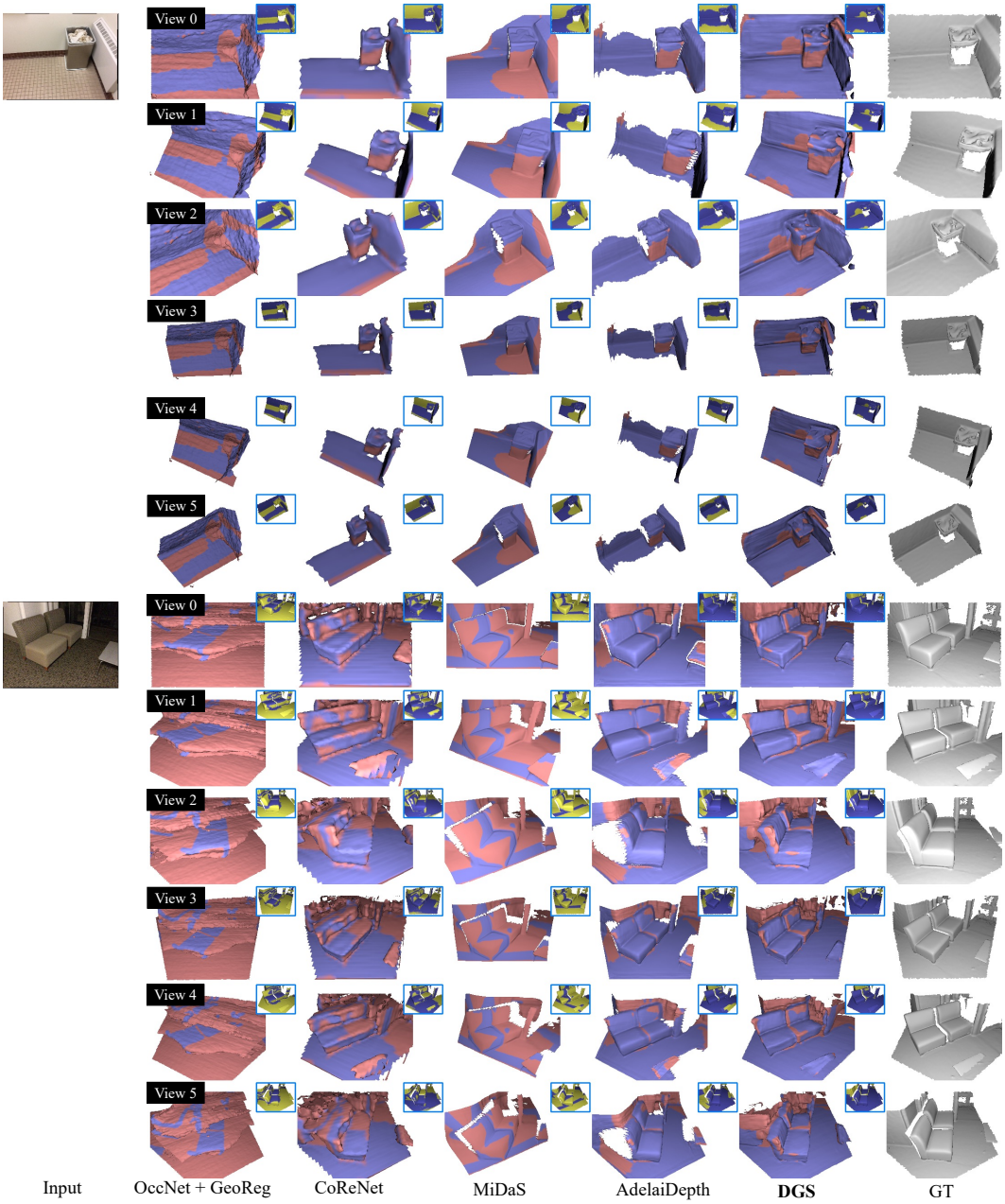

Figure 17: Quantitative comparison on the ScannetV2 (without handpick: the first frame of the 7th and 8th test scene in ScannetV2). The reconstruction result of each approach is visualized in six different views, with the first view the same as the camera view, the first three views the same elevation as thecamera view, and the last three view elevated view.

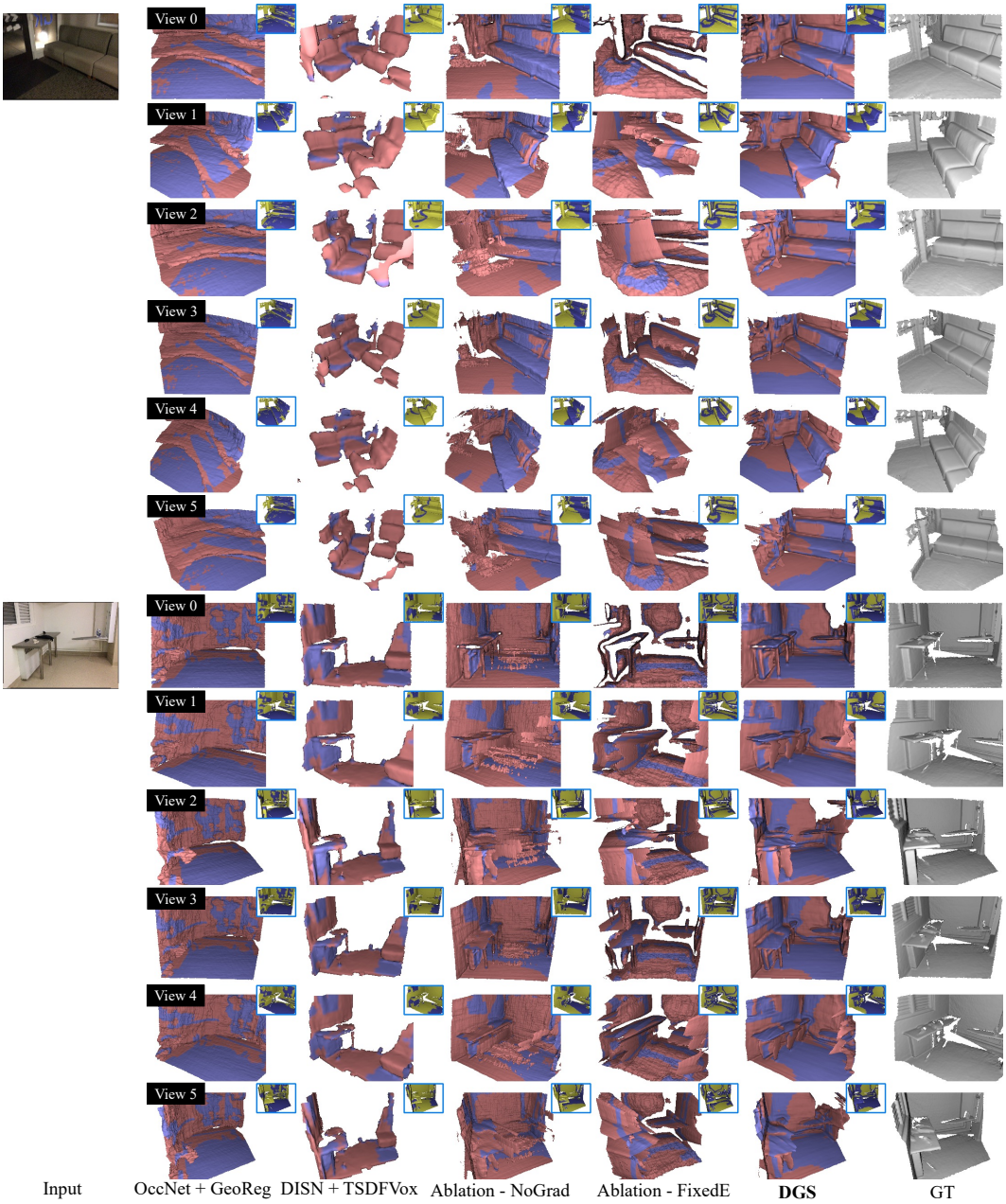

Figure 18: Quantitative comparison on the ScannetV2 (without handpick: the first frame of the 9th and 10th test scene in ScannetV2). The reconstruction result of each approach is visualized in six different views, with the first view the same as the camera view, the first three views the same elevation as thecamera view, and the last three view elevated view.

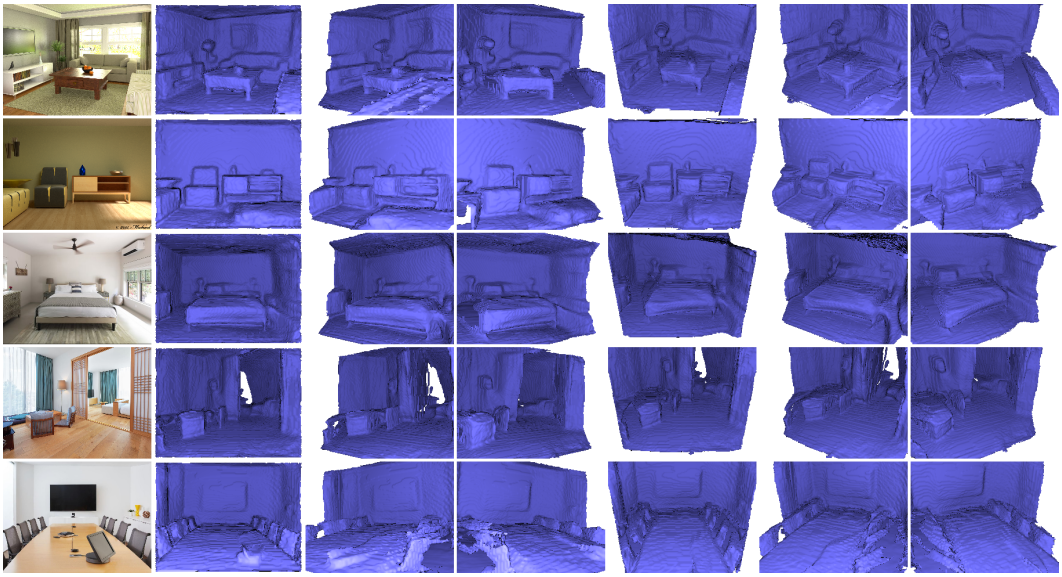

Figure 19: Additional Qualitative results of our model generalizing to unseen test images downloaded from the Internet.