Implicit Modeling for 3D Applications

Shizhan Zhu

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2023-266
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2023/EECS-2023-266.html

December 11, 2023

Implicit Modeling for 3D Applications

by

Shizhan Zhu

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Trevor Darrell, Chair
Professor Angjoo Kanazawa
Professor Daniel Klein
Doctor Christoph Lassner

Fall 2023

Implicit Modeling for 3D Applications

by

Shizhan Zhu

Abstract

Implicit Modeling for 3D Applications

by

Shizhan Zhu
Doctor of Philosophy in Computer Science

University of California, Berkeley
Professor Trevor Darrell, Chair

Implicit model has emerged as a promising representation for numerous applications, including signal processing, image representation as well as 3D modeling. In this thesis, we will showcase its applications in three domains. First, we show that implicit representation could aid single view surface reconstruction of scenes by learning from a large collections of indoor scene data, and generalize well to unseen indoor scenes. Second, we show that implicit modeling can facilitate grounding of the language query in the 3D space. Lastly, we show that neural relighting, a type of algorithm that incorporates the neural field for modeling the light field, could achieve promising results. On one hand, by incorporating the pre-computed radiance transfer into the neural radiance modeling, we can enable handling of the type of materials with subsurface scattering effects. A hypernet-based representation could further facilitate fast image-based relighting. On the other hand, by carefully modeling the local micro-geometry with the surface normal modeling, as well as incorporating various hints for lighting reflection and self reflection, we could faithfully recover reflection highlights with varying material roughnesses.

To my parents.

Contents

Acknowledgments

I am more than fortunate to have Professor Trevor Darrell and all the labmates in the Darrell-group, work hand-in-hand, all along the hard journey, to finally reach the destination of the degree – with the acquisition of real solid skills in technical research. The Ph.D. study in schools like Berkeley will not be easy – "survival" is the kind of word that is the frequent description of the academic status of most of the students. I could still remember that at the start of the study program when I was seemingly able to conduct research on a relatively narrower range of topics where topics are familiar, benchmarking is trivial to implement, datasets are already readily available, loss curve is the only research debugging signal and is proportional to the benchmarking, and baseline approaches already work (where we are just to improve it to a slightly better level). These are not the types of research that are well acceptable to Berkeley. The last piece of my master period's work could be a good start to this journey, while it took time to incorporate that accidental surprise into my regular research outputs.

The exact reason that pushed me to work on 3D, a brand new area that I do not have particular technical experience inherited from my master period, is that I had some sort of epiphany that a view-consistent, out-of-plane rotation friendly representation (like 3D) would be a good catch for image generation, particularly with structural generation (e.g. the moving object boundary and segments, in contrast to textual / pixel modeling only) and temporal motion handling. My goal is to handle the capturing of the real scenes beyond just those "clean" graphics rendering. It can go from solid objects to articulated ones and finally to fluids. Professor Trevor Darrell is the faculty that is willing to take me warmly into the group, even if this area seems to be far away from his main area (indeed, time has proven that languages and robotics are now a real hotspot of the vision community!). As all his past students have told me, Professor Trevor Darrell is so supportive of his students on almost all fronts, including free exploration of research topics in the first several years of the Ph.D., encouraging technical discussion among students working on similar areas, sufficient caring for students coming from non-American background, anytime mental health coaching, as well as all the pieces of administrative side of support. I could still remember that in my first year I could even hardly understand people's English speaking, it was my advisor that always encouraged me to participate in all kinds of discussions. Professor Trevor Darrell is a well-established professor and is a technical expert on topics that belong to his golden days (like scape human models, active shape models, features and the early deep learning era), while still making great efforts to keep pace with all the new concepts that come into being in recent years (like NeRF) so that he could have a good understanding of everyone's work. My advisor also seeks external help when he believes a more technically dedicated researcher could further aid my program of study, like Angjoo Kanazawa and Christoph Lassner. I am really moved that I could have such a caring and supportive advisor and it is

so difficult to seek for another in Berkeley. My PhD study has also spanned such a special time period that my country of residence and my country of origin were on a free fall of total national relations, in parallel with all the sorrows and saddens hit by the covid era. It is my advisor who reassures me that he will carry and accompany me till the end of the program of study no matter what happens. Now at the point of graduation, how I wish I could have been establishing more works with my advisor to let them go with all of my gratitude to him over these years.

My gratitude also goes to my collaborators over the years in my program of study. My Meta internship manager Dr. Christoph Lassner often reminds me of my master advisor Professor Chen Change Loy that he is so devoted, so caring, so faithful to all aspects of our collaborations. My most recent project on neural relighting was a great opportunity that I learned from him and all the expert collaborators. I always remember that he points to one of the unsolved problems in the existing neural relighting algorithm that subsurface scattering (SSS) effects is not yet able to be handled, that opens up the whole page of our research on neural relighting with SSS. He also pointed out that the local micro geometry can be well captured under the light stage – instead, a single lighting capture or any single point light condition cannot fully reveal all the tiny geometric details of the captured objects and the scenes. He is always there that I could turn to for help on technical problems or high-level difficulties, including holidays and weekends. We have fruitful technical discussion roughly twice a week and I feel so confident to present my progress during the whole-loop weekly meeting. I was so benefited from his wonderful Meta onboarding video sequences so that I could quickly get the internal infrastructure of Meta Reality Labs in 2 days. He gave me all the courage on technical research to avoid self-doubting, and all the strength and belief that the problem shall always be resolved eventually. He keeps accompanying me until my project and finished, and eventually, I get the honor of having him in my thesis committee as well. I am also very fortunate to work with and learn from all the nice, helpful, expert collaborators including Dr. Shunsuke Saito, who is so expertised in relighting that pointed me to the pre-computed radiance transfer literatures so that I could establish a NeRF-incorporated algorithm that can handle global illumination and SSS (in parallel to other similar work coming out), Dr. Aljaz Bozic, who is so reliable and productive that I could always turn to for help on any technical details, and Dr. Carlos Aliaga, who teaches me all the graphics aspects, the rendering techniques, the visualization tools as well the detailed specs about Blender.

I also wish to thank Professor Angjoo Kanazawa for her generous guidance and help when she was a postdoc of Professor Trevor Darrell. I do believe my initial established sets of 3D skills and tooling are all at her generous help. Professor Angjoo Kanazawa pointed to one of the emerging hotspot topics of the implicit model that eventually constructed the soul of this thesis. She understood one of my research difficulties is how to show weekly progress well, especially in the visualized way, to senior collaborators. She showcased to me her own slides during her presentation to

more senior faculties. She taught me how to do conceptual debugging, which is a material missing piece of my past research conduct. I shall always keep in mind all the technical skills she has taught me for stronger research conduct.

I also wish to thank Dr. Sayna Ebrahimi in the Darrell-group, who is such a strong person and researcher, that always encourages me to be courageous when facing all types of difficulties. She fainted when she was doing research and our advisor Professor Trevor Darrell physically accompanied her to the hospital. She continued to work on research after leaving the hospital with her strong will, and with that level of pain and physical difficulties, with even the suffering of "forgetting", she still managed to finish her work on"learning without forgetting". During the covid era, Sayna was one of my friends who kept encouraging me to stay up and work with high spirits. Her own fighting spirit and her encouragement to all the students in the group shall always shine in our hearts to keep the faith and stay encouraged.

I also wish to add my thankfulness to the collaborators in Adobe as well as Dr. Tinghui Zhou who are willing to offer me the opportunity of the collaboration although not achieved eventual publications. I do believe those are also critical experiences for me to discover the insufficiency at that point and I do believe it will be a successful project if it happens in my fifth year of PhD.

Last as always, my thanks to my parents as well as my grandparents for all their unconditional love and support over my years during the PhD period.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Implicit Models

Implicit models have emerged as one of the promising representations that become widely used in various fields, including signal processing, image representations, 3D modeling, etc. In recent years, implicit models have particularly drawn attention from the 3D research community and a considerable fraction of research is conducted based on the 3D implicit models. A 3D implicit model is a function representation that maps space coordinates or other parameterizations (e.g. view directions or lighting directions) to the prediction space, and use this function instead of any explicit representation for representing a 3D model. For instance, a 3D occupancy field could serve as an example, where the implicit function f maps 3D coordinates $\mathbf{x} \in \mathbb{R}^3$ to a probability scaler $o \in \mathbb{R}(0 \leq o \leq 1)$. A relatively large probability scaler o would indicate the space location $\mathbf{x}$ is more likely to be occupied, while a relatively small s would indicate free space. The occupancy of the space would then be easily modeled by this implicit function f. Compared to other representations such as voxel grids or meshes, the implicit representation prevails in several manners. First, its representation is relatively compact as this function could be generally modeled by just a handful of fully connected layers, compared to a total enumeration of the occupancy as in the voxel grids or all the triangles as in the mesh. Second, due to the continuity nature of the function space, an implicit representation could in theory represent an extremely high resolution of the represented space. Lastly, the implicit model is flexible regarding input dimensions and their physical meaning, which possesses the potential of representing space that is higher than 3 dimensions (e.g. a Neural Radiance Field).

There are several important recent implicit models that have been widely used by the 3D research community:

• **Signed Distance Field (SDF)**. A signed distance function takes in the 3-dimensional 3D coordinates as the input, and maps to the signed distance value $s \in \mathbb{R}$, where its sign represents occupancies (generally positive for unoccupied and

negative for occupied) and its value represents the shortest distance to a point on the surface (i.e. the projection distance). As such, the zero-level set would just be the surface. One can extract the mesh via the marching cube algorithm over either a pre-evaluated signed distance volume grid or a coarse-to-fine online querying of the signed distance function (e.g. the toolbox MISE). Both the occupancy field and the signed distance field would serve as important 3D geometry representations. Our 3D surface prediction tasks in Chapter 2 would deal with this type of representation.

• **Neural Radiance Field (NeRF)**. The seminal work on the Neural Radiance Field addresses not only the geometry but also the appearance of a scene when observed from a particular view angle. More precisely, NeRF advocates a volume rendering scheme for getting the pixel RGB values of each camera ray accumulated with a plenoptic function. For a single point $\mathbf{x} \in \mathbb{R}^3$ in the space that observes from the direction of $\mathbf{d} \in \mathbb{R}^3$ ($\|\mathbf{d}\|_2 = 1$), the function would return a density $\sigma \in \mathbb{R}(\sigma > 0)$ that is only conditioned on $\mathbf{x}$, as well as an RGB value that is conditioned on both $\mathbf{x}$ and $\mathbf{d}$. Both the density and the RGB value would contribute to the volume rendering accumulation along the pixel ray for computing the final pixel RGB. When provided with calibrated camera poses, NeRF could provide a faithful reconstruction of the observed objects or scenes from the multi-view image captures. Our neural relighting algorithms in Chapter 4 are built upon this type of representation.

1.2 Recent 3D tasks

Great attention has been put to the 3D vision tasks by the vision community in recent years. A rough categorization of several hotspot research problems in the most recent years could be *i)* Reconstruction - from 2D observations (e.g. RGB) to 3D prediction; *ii)* 3D generation or editing: from an old 3D structure or generation specification to a new 3D structure; *iii)* Differentiable rendering - from 3D structure to 2D observations; *iv)* Semantic labeling - from 3D structure to semantic label prediction; and *v)* 3D structure as the underlying representation - e.g. view consistent image generation. Part of the research problem or algorithm could be a mixture of components from multiple categories from above - e.g. Neural Radiance Field is both for "reconstruction with known camera poses" (similar to Multi-view Stereo) and "differentiable rendering" (volume rendering). Our work on single image surface reconstruction (Chapter 2) falls into the "reconstruction" category, our work on language grounding (Chapter 3) falls into the "semantic labeling" category, while our neural relighting (Chapter 4) is connected with both "reconstruction" and "differentiable rendering" following the categorization of NeRF.

1.2.1 Generalizability: Fitting without Hallucination v.s. Learning with Hallucination

One of the crucial distinctions between various tasks is the generalizability of the particular task. On one hand, a task can be just a total reconstruction of the 3D object or the scene from all the collected observations. Such reconstruction is committed to reflecting all the observed cues without adding prior knowledge by the model from learning from a large collection of training sets. Typically this type of approach is supposed to address the given particular 3D object or the scene very accurately, while it cannot generalize to unseen 3D samples at all. We name this type as "fitting without hallucination". Representative algorithms including Neural Radiance Field reconstruction, neural surface reconstruction, and neural relighting given image captures under multiple lighting conditions. On the other hand, a task can be more machine-learning flavored, that the model is supposed to learn from a large collection of training samples among multiple 3D objects or scenes. The learned prior can serve as the hint to incorporate to the inference procedure over unseen test cases. Generally the test case is also under partial observation only (e.g. single view prediction or material decomposition under a single lighting condition) and needs extra learned priors for a complete plausible prediction. We name this type as "learning with hallucination".

It is worth pointing out that the borderline between the two different types above could be blurred - one can also incorporate a learned prior as the regularizer in the process of single sample fitting. For instance, in [12], the diffusion prior is learned for free space floater removal. It is also generally not clear how many lighting conditions or views are considered "sufficient observations" that do not need any learned prior knowledge from abundant external training data.

In this thesis, our efforts on the single view implicit surface reconstruction (Chapter 2) fall into the "learning with hallucination" category while our work on neural relighting from the light stage data (Chapter 4) falls into the "fitting without hallucination" category.

1.3 Thesis Organization

In Chapter 2, we introduce our work on single-view surface reconstruction. Existing approaches for single object reconstruction impose supervision signals based on the loss of the signed distance value from all locations in a scene, posing difficulties when extending to real-world scenarios. The spatial gradient of the signed distance field, rather than the SDF value itself, has not been typically employed as a source of supervision for single-view reconstruction, in part due to the difficulties of differentiable sampling a spatial gradient from the feature map. In this study, we derive a novel closed-form gradient sampling solution for Differentiable Gradient Sampling

(DGS) that enables backpropagation of the loss of the spatial gradient back to the feature map pixels, thus allowing the imposition of the loss efficiently on the spatial gradient. As a result, we achieve high-quality single-view indoor scene reconstruction results by learning directly from a real-world scanned dataset (e.g. ScannetV2). Our model also performs well when generalizing to unseen images downloaded directly from the internet.

In Chapter 3, we present the Voxel-informed Language Grounder (VLG), a language grounding model that leverages 3D geometric information in the form of voxel maps derived from the visual input using a volumetric reconstruction model. We show that VLG significantly improves grounding accuracy on SNARE, an object reference game task. At the time of writing, VLG holds the top place on the SNARE leaderboard, achieving SOTA results with a 2.0% absolute improvement.

In Chapter 4, we conduct reconstructing and relighting objects and scenes under varying lighting conditions. Existing neural rendering methods often cannot handle the complex interactions between materials and light. Incorporating pre-computed radiance transfer techniques enables global illumination, but still struggles with materials with subsurface scattering effects. We propose a novel framework for learning the radiance transfer field via volume rendering and utilizing various appearance cues to refine geometry end-to-end. This framework extends relighting and reconstruction capabilities to handle a wider range of materials in a data-driven fashion. The resulting models produce plausible rendering results in existing and novel conditions.

Chapter 2

Single View Implicit Scene Reconstruction with Differentiable Gradient Sampling

2.1 Background and Related Works

Recent studies have shown promising learning capabilities of implicit models for 3D geometry representations [13, 14, 15, 16, 17, 18, 19]. Rather than explicitly representing 3D geometry with a textured mesh [20] or point cloud [21], implicit representations express a function of points in the space. Among many recent promising implicit models, the occupancy field [14, 2, 22] and the Signed Distance Field (SDF) [13, 23, 24] are particularly suitable for high resolution topology free 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 returns the closest distance from a point to the surface of the given 3D geometry, as well as its sign for its occupancy. The surface is defined by the classification boundary of the occupancy field or the zero level-set of the SDF [25].

Despite the strong representational power of these models and their success on predicting 3D objects from single image, 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 [26] (Fig. 2.2). 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 [27, 28, 29, 30]. Storing dense query points sufficiently 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 [31]. 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 [2].

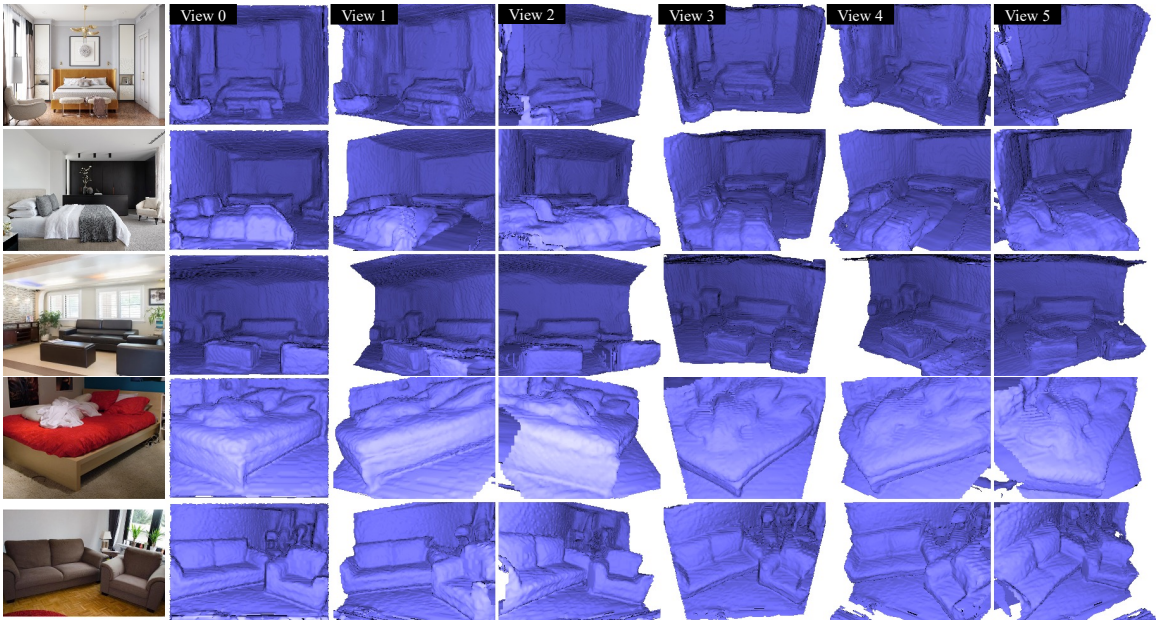


Figure 2.1: Given a single RGB image as the input (the first column), our model can predict its 3D implicit surface reconstruction (shown in six novel views in the last six columns). The test images for the first 3 rows are downloaded from the Internet and the last 2 rows are from the pix3d dataset.

Recent studies on fitting a single scene geometry to partial 3D observations [17, 32] 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 [33]. Similarly, in this work, we propose to regularize the gradient of the occupancy field to be zero away from the surface (Fig. 2.3(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 [17, 32] have only demonstrated this benefit in the cases of geometry inputs only, where a network is fit to a particular scene per model without any conditioning on input images. When locally sampled image features are used in 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

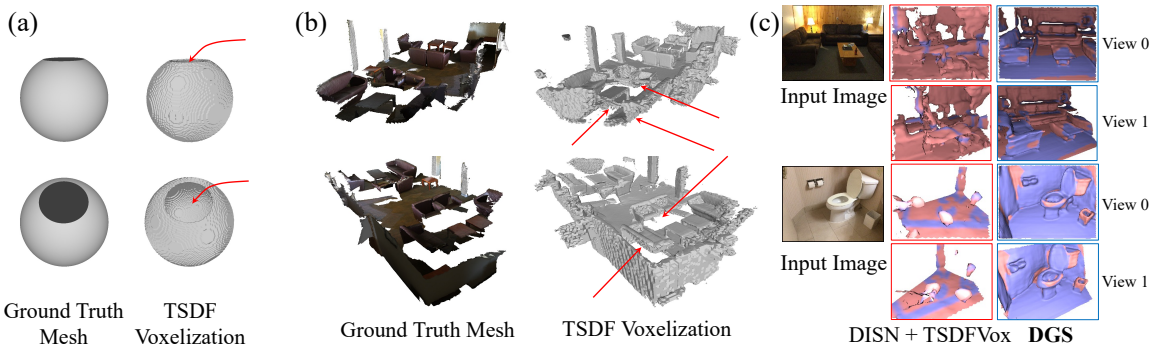


Figure 2.2: (a, b) Truncated SDF (TSDF) [1] Voxelization results of the non-watertight ground truth meshes (each shown in two views). (a) is a simple sphere and (b) is a real scene from the ScannetV2 training data. After depth-fusion and internal-filling [2], the inside space of both geometries (a, b) 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 TSDF voxels (due to engineering constraints on runtime loading and memory bottleneck for the sufficiently dense pre-computed query point occupancy labels), 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".

model (Fig. 4.11). Our novel propagation of the loss gradient of the spatial gradient back to the feature maps (Eq. 2.6) serves in addition to the existing spatial gradient sampling (Eq. 2.5) and loss gradient back-propagation (Eq. 2.4) used in existing deep learning frameworks [34, 35]. The resulting end-to-end learning scheme with supervision over the spatial gradients opens the potential to train a model generalizable to unseen test cases of complex scenes with only surface point cloud supervision — a typical setting for real-world 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. Experiments on real-scanned data (ScannetV2 [29]) shows 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). Experiments on synthetic data (ShapeNet [36]) indicate that our learning framework without using dense per-query-point training labels demonstrates competitive performance compared to the oracle scenario where dense occupancy labels are available. 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 2.1).

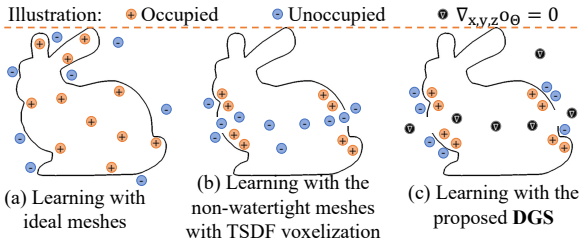


Figure 2.3: Illustration of the loss imposition for the occupancy prediction scenario. (a) When learning from the ideal mesh for ShapeNet objects, we can directly supervise the training with the accurate occupancy labels. (b) On scans of real scenes with imperfections (Fig. 2.2(b)), the TSDF voxelization produces 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 all the way back to image features.

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 meshes which assume a fixed topology). Implicit models typically learn to establish a mapping between a query point and the prediction of the point. [37] proposed to learn a mapping from the uv texture map to the 3D surface point. More recent works [13, 14, 23, 2, 38] 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 [18] also model the surface texture via learning a mapping from the surface point to the RGB value, or use the RGB loss as the supervision signal [39, 40]. In contrast to defining textures explicitly on the surface, [15] proposed the volumetric rendering representation which maps the point coordinate and the viewing angle to volumetric textures. [41, 7, 42] use volume renderings for fitting high quality geometry of single scenes, but these are not generalizable to unseen scenes. In multi-view stereo setting, [43, 44, 45] 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 3D geometry reconstruction reconstructions from single view, as an example of the application of the proposed DGS module.

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 [46] is one of the early approaches for differentiable sampling of the feature map. [20] proposed to render the mesh into image in a differentiable manner to optimize the mesh or the camera parameters to fit to a particular image. A similar idea has also been applied for rendering point clouds [47, 48]. [49] and [15] 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 [50], 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.

2.2 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 employs 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 $f(\phi; x, y, z)$. 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 $o(\phi; x, y, z) \in \{\text{“}-\text{”}, \text{“}+\text{”}\}$, 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 $s(\phi; x, y, z) \in \mathbb{R}$ respectively.

Background. Typical fully supervised training [23, 2] 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 $f(\phi; x, y, z)$. For occupancy predictions, we compute the loss as $\sum_{\{I,\mathcal{P}_I\}\in\mathcal{D}} \sum_{(x,y,z)\in\mathcal{P}_I} \text{BCE}(\hat{o}_\Theta(\phi; x, y, z), o_\Theta(\phi; x, y, z))$ (Fig. 2.3(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 $\sum_{\{I,\mathcal{P}_I\}\in\mathcal{D}} \sum_{(x,y,z)\in\mathcal{P}_I} |\hat{s}_\Theta(\phi; x, y, z) - s(\phi; x, y, z)|$.

2.2.1 Proposed Framework

As obtaining the full labels $f(\phi; x, y, z)$ for query points (x, y, z) from every location in $\mathcal{P}_I$ is non-trivial for complex geometry from real-world (Fig. 2.2) 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 of the implicit field,

$$\begin{aligned}\mathcal{L} = \sum_{\{I,\mathcal{P}_I\}\in\mathcal{D}} \Big(& \sum_{(x,y,z)\in\mathcal{P}_I - \mathcal{P}_I^0} \lambda_{or} \|\nabla_{x,y,z} \hat{o}_\Theta(\phi; x, y, z)\| \\ + & \sum_{(x,y,z)\in\mathcal{P}_I^{0+}} \text{BCE}(\hat{o}_\Theta(\phi; x, y, z), \text{“}+\text{”}) + \sum_{(x,y,z)\in\mathcal{P}_I^{0-}} \text{BCE}(\hat{o}_\Theta(\phi; x, y, z), \text{“}-\text{”})\Big)\end{aligned} \tag{2.1}$$

where $\nabla_{x,y,z} \hat{o}_\Theta(\phi; x, y, z) = [\nabla_x \hat{o}_\Theta(\phi; x, y, z), \nabla_y \hat{o}_\Theta(\phi; x, y, z), \nabla_z \hat{o}_\Theta(\phi; x, y, z)]^\top$ denotes the spatial gradient of the occupancy prediction, $\mathcal{P}_I^{0+}$ and $\mathcal{P}_I^{0-}$ denote 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

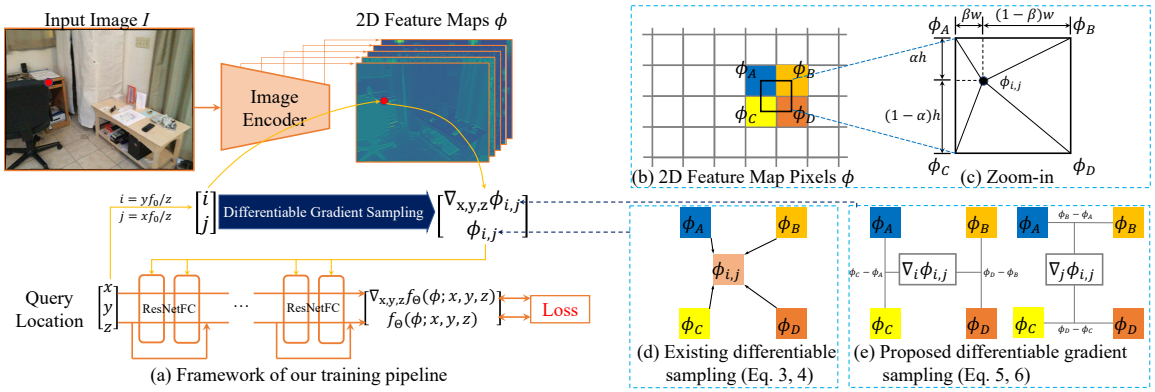


Figure 2.4: Overview of our learning framework (a) and differentiable gradient sampling (b, c, d, e).

inward or outward) can be determined by normals (if available), or via rendering the surface in all views as in the RGBD captures [29] (with the surface facing toward camera as the "outward" side). The three terms in Eq. 2.1 correspond to the three types of losses in Fig. 2.3(c) ("∇", "$+$", "$-$") respectively. For SDF, we set the loss similar to [17]:

$$\begin{aligned}\mathcal{L} = \sum_{\{I,\mathcal{P}_I\}\in\mathcal{D}} \Big(\sum_{(x,y,z)\in\mathcal{P}_I} \lambda_{sr} \big| \|\nabla_{x,y,z}\hat{s}_\Theta(\phi; x, y, z)\|_2 - 1 \big| + \sum_{(x,y,z)\in\mathcal{P}_I^0} |\hat{s}_\Theta(\phi; x, y, z)| \\ + \sum_{(x,y,z)\in\mathcal{P}_I^0} \lambda_{sn} \|\nabla_{x,y,z}\hat{s}_\Theta(\phi; x, y, z) - \nabla_{x,y,z} s(\phi; x, y, z)\| \Big)\end{aligned} \tag{2.2}$$

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 are the Eikonal regularization [33], surface zero-SDF loss, and the surface normal loss respectively, where the last term is optional [17]. Note now the loss functions in Eq. 2.1, 2.2 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 the proposed loss for the occupancy model (Fig. 2.3(c), Eq. 2.1) to be effective in the real-world single-view feed-forward scenario, and use Eq. 2.1 rather than Eq. 2.2 in all of our experiments. There are a few reasons. First, learning with the Eikonal term (Eq. 2.2) 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 depart from the single objects scenario [14, 2] or single scene fitting [17] to large schale scenes, which is our focus. Second, our loss function (Eq. 2.1) can significantly save memory footprint during training and enable large batch size training, which is crucial in our learning framework. Unlike in Eq. 2.2 where all the query points require spatial gradient computation, which leads to 3× higher of memory footprint, in our loss function (Eq. 2.1), 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.

2.2.2 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. [17] 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 image sampling procedure. [51, 52] used a non-spatial global feature for inference and hence bypassed 2D 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 4.11(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 [17], 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 $\nabla_{x,y,z}\phi = [\nabla_x\phi, \nabla_y\phi, \nabla_z\phi]^\top$ 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 coordinate system that ranges from -1 to 1. As illustrated in Fig. 4.11(d), given the feature map ϕ and the sampling locations (i, j), the resulting sampled value $\phi(i, j)$ is

$$\phi(i, j) = (1 - \alpha)((1 - \beta)\phi_A + \beta\phi_B) + \alpha((1 - \beta)\phi_C + \beta\phi_D) \tag{2.3}$$

Please refer to Fig. 4.11(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

$$\frac{\partial L}{\partial \phi_A} = (1-\alpha)(1-\beta)\frac{\partial L}{\partial \phi(i,j)}. \tag{2.4}$$

Equation 2.4 is for Pixel A and similarly to the other 3 pixels (please refer Eq. 2.9).

2D Differentiable Gradient Sampling. Our learning framework (Sec. 2.2.1, Fig. 4.11(a)) requires the extension of the sampling capability from just the feature value response $\phi_{i,j}$ to its spatial gradient $\nabla_{i,j}\phi(i,j) = [\nabla_i\phi(i,j), \nabla_j\phi(i,j)]^\top$. 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. 4.11(e)):

$$\nabla_i\phi(i,j) = \frac{(1-\beta)(\phi_C - \phi_A) + \beta(\phi_D - \phi_B)}{h}. \tag{2.5}$$

Please refer to Eq. 2.10 for $\nabla_j\phi(i,j)$. Hence, during the forward pass, we compute the spatial gradient via Eq. 2.5 in addition to the existing value sampling (Eq. 2.3). During the backward pass, we compute the loss gradient over the spatial gradient via

$$\begin{aligned}\frac{\partial L}{\partial \phi_A} &= \frac{\partial L}{\partial \phi(i,j)}\frac{\partial \phi(i,j)}{\partial \phi_A} + \frac{\partial L}{\partial \nabla_i\phi(i,j)}\frac{\partial \nabla_i\phi(i,j)}{\partial \phi_A} + \frac{\partial L}{\partial \nabla_j\phi(i,j)}\frac{\partial \nabla_j\phi(i,j)}{\partial \phi_A} \\ &= (1-\alpha)(1-\beta)\frac{\partial L}{\partial \phi(i,j)} + (-\frac{1-\beta}{h})\frac{\partial L}{\partial \nabla_i\phi(i,j)} + (-\frac{1-\alpha}{w})\frac{\partial L}{\partial \nabla_j\phi(i,j)},\end{aligned} \tag{2.6}$$

where w and h are the width and height of a pixel in normalized coordinate system, s.t. $w = 2/W$ and $h = 2/H$ for the feature map with the size $W \times H$. Please refer to Eq. 2.11 for $\frac{\partial L}{\partial \phi_B}$, $\frac{\partial L}{\partial \phi_C}$ and $\frac{\partial L}{\partial \phi_D}$.

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 $i = \frac{yf_0}{z}, j = \frac{xf_0}{z}$. The feed forward pass is

$$\begin{aligned}\nabla_x\phi(x,y,z) =& \nabla_i\phi(i,j)\cdot\frac{\partial i}{\partial x} + \nabla_j\phi(i,j)\cdot\frac{\partial j}{\partial x} = \nabla_j\phi(i,j)\cdot\frac{f_0}{z}, \\ \nabla_y\phi(x,y,z) =& \nabla_i\phi(i,j)\cdot\frac{\partial i}{\partial y} + \nabla_j\phi(i,j)\cdot\frac{\partial j}{\partial y} = \nabla_i\phi(i,j)\cdot\frac{f_0}{z}, \\ \nabla_z\phi(x,y,z) =& \nabla_i\phi(i,j)\cdot\frac{\partial i}{\partial z} + \nabla_j\phi(i,j)\cdot\frac{\partial j}{\partial z} = \nabla_i\phi(i,j)\cdot(-\frac{yf_0}{z^2}) + \nabla_j\phi(i,j)\cdot(-\frac{xf_0}{z^2}).\end{aligned} \tag{2.7}$$

Note that here, our notation of the sampled 2D image feature $\phi(x,y,z)$ refers to the same as $\phi(i,j)$, as (i,j) is the projected pixel coordinates of (x,y,z) that represent the exact projected 2D locations when extracting the 2D features.

Table 2.1: Intersection over Union % (IoU ↑) benchmarking result on the high-realism ShapeNet. Our approach (the last 4 rows) demonstrates competitive performance compared to state-of-the-art approaches (the top 4 rows) even trained without the dense occupancy labels as used in the oracle settings of these existing works. Our comparisons with the state-of-the-art approaches are direct ablations as we maintain exactly the same experimental setups except for the loss function. In addition, our approach also comfortably outperforms the ablation baselines (the middle 5 approaches).

Category	Craft	Rifle	Disp.	Lamp	Spk.	Box	Chair	Bench	Car	Plane	Sofa	Table	Phone	Mean
OccNet	49.6	39.7	49.7	33.3	49.3	42.1	42.8	30.9	57.2	41.7	60.7	42.4	64.8	46.5
DISN	54.5	52.5	50.2	39.2	53.3	46.0	50.6	37.1	58.6	48.5	64.9	48.4	67.6	51.6
CoReNet	60.5	67.5	61.0	46.9	56.8	51.3	59.7	47.1	61.1	58.4	68.7	56.9	77.3	59.5
DISN(Res50)+DVR	61.1	64.5	61.9	46.9	58.2	54.4	59.6	48.0	59.4	58.4	69.5	57.2	78.7	59.8
Abl.-NoGrad	11.9	4.9	23.6	15.4	31.4	30.6	25.1	10.1	20.8	7.7	28.3	20.1	21.2	19.3
Abl.-NoGrad (10 %)	23.7	19.9	33.1	26.5	40.0	35.2	35.6	18.6	33.0	15.7	38.8	30.7	41.0	30.1
Abl.-NoGrad (30 %)	26.4	15.5	36.8	23.6	40.6	36.0	36.8	19.5	38.5	19.1	47.8	30.9	39.4	31.6
Abl.-NoGrad (50 %)	39.7	30.0	39.7	28.4	45.0	37.5	39.0	23.6	50.6	29.0	53.5	36.2	49.7	38.6
Abl.-FixedE	49.6	52.5	44.4	33.0	46.3	39.9	43.4	27.5	57.5	46.9	58.8	39.1	68.0	46.7
OccNet w/ Eq. 2.1	50.7	42.6	50.4	33.0	50.1	43.7	44.4	32.9	56.8	41.6	60.9	44.0	68.4	47.7
DISN w/ Eq. 2.1	53.3	51.2	51.9	38.7	52.1	43.8	50.8	36.2	58.8	47.4	64.6	47.4	66.6	51.0
CoReNet w/ Eq. 2.1 (DGS)	61.1	67.5	62.7	44.2	54.8	49.6	59.5	45.4	59.4	59.9	69.8	55.1	78.0	59.0
DISN(Res50)+DVR w/ Eq. 2.1 (DGS Best)	60.8	62.6	62.3	47.1	57.7	53.5	59.8	47.4	59.2	58.6	70.7	57.4	77.0	59.6

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

$$\begin{aligned}\frac{\partial L}{\partial \phi_A} =& \frac{\partial L}{\partial \phi(x,y,z)}\frac{\partial \phi(x,y,z)}{\partial \phi_A} + \frac{\partial L}{\partial \nabla_x \phi(x,y,z)}\frac{\partial \nabla_x \phi(x,y,z)}{\partial \phi_A} \\ &+ \frac{\partial L}{\partial \nabla_y \phi(x,y,z)}\frac{\partial \nabla_y \phi(x,y,z)}{\partial \phi_A} + \frac{\partial L}{\partial \nabla_z \phi(x,y,z)}\frac{\partial \nabla_z \phi(x,y,z)}{\partial \phi_A} \\ =& \frac{\partial L}{\partial \phi(x,y,z)} \cdot (1-\alpha)(1-\beta) + \frac{\partial L}{\partial \nabla_x \phi(x,y,z)} \cdot (-\frac{1-\alpha}{w}) \cdot \frac{f_0}{z} \\ &+ \frac{\partial L}{\partial \nabla_y \phi(x,y,z)} \cdot (-\frac{1-\beta}{h}) \cdot \frac{f_0}{z} + \frac{\partial L}{\partial \nabla_z \phi(x,y,z)} \cdot (\frac{1-\beta}{h} \cdot \frac{y f_0}{z^2} + \frac{1-\alpha}{w} \cdot \frac{x f_0}{z^2}).\end{aligned} \tag{2.8}$$

Please refer to Eq. 2.12 for $\frac{\partial L}{\partial \phi_B}$, $\frac{\partial L}{\partial \phi_C}$, and $\frac{\partial L}{\partial \phi_D}$.

2.3 Experiments

2.3.1 Learning from Synthetic Data (ShapeNet)

Dataset. Following [2], we conduct experiments on ShapeNet with the low-realism and the high-realsim renderings. For low-realism, we use the renderings of various models from [53]. Following [14], 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 [14] 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 [2] - we only use the 13 categories used by [53], 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 [2], we only evaluate the first 1% test cases of the test set (1898 samples).

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 [14] that evaluating occupancy of the specified 100K query points in the space. For high-realism renderings, we follow the protocol from [2] to evaluate occupancy prediction of the $128 \times 128 \times 128$ grid. Following [2], 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.

Comparison with state-of-the-art approaches. On high-realism ShapeNet, we build our model on top of the representative state-of-the-art models - OccNet [14], DISN [23], CoReNet [2] for their 3D volumetric implicit modeling capability. We also incorporate a strong baseline named "DISN (ResNet50) + DVR [39]", where we replace the originally used VGG [54] with ResNet-50 as the image encoder as used in CoReNet [2], and devised the 5-layer fully connected ResNet as the decoder as used in DVR [39]. In each DGS experiment, we maintain exactly the same encoder and decoder architecture, optimization parameters (e.g. the learning rate and the epsilon of Adam [55]) and the prediction format (per-query occupancy probability). We set λ_{or} to be 0.01 in our loss function (Eq. 2.1). Note that the DGS experiments can only access nearsurface training signals, in contrast to the oracle learning setting as in the state-of-the-art approaches where models were trained with dense occupancy labels. We name the DGS version of the CoReNet model as the **DGS** model, and the DISN+DVR counterpart as the **DGS-Best** model. Quantitative results are reported in Tab. 2.4 for the low-realism evaluation setting, and in Tab. 2.1 for the high-realism setting[1]. Our competitive experimental results indicate that our learning framework along with the differentiable gradient sampling layer implementation plays the critical role when learning with only the near-surface training labels, and can achieve similar performance without the dense occupancy training labels.

Please refer to Sec. 2.5 for ablation baselines details and low-realism ShapeNet experimental settings.

2.3.2 Learning from Real Scanned Datasets (ScannetV2)

Dataset. We use ScannetV2 [29] for training and evaluating the performance of the models on the real images. We follow the standard training / testing split as used in [44] and [43], 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

[1]We noticed that in our original submission version of the paper, our experiments were conducted with evaluating the $64 \times 64 \times 64$ volume grids due to an error. We thus revised our results in Tab. 2.1 with the evaluation results returned from the $128 \times 128 \times 128$ volume grid, and attach our previous results in Tab. 2.3. Please refer to Sec. 2.5 for detailed explanation.

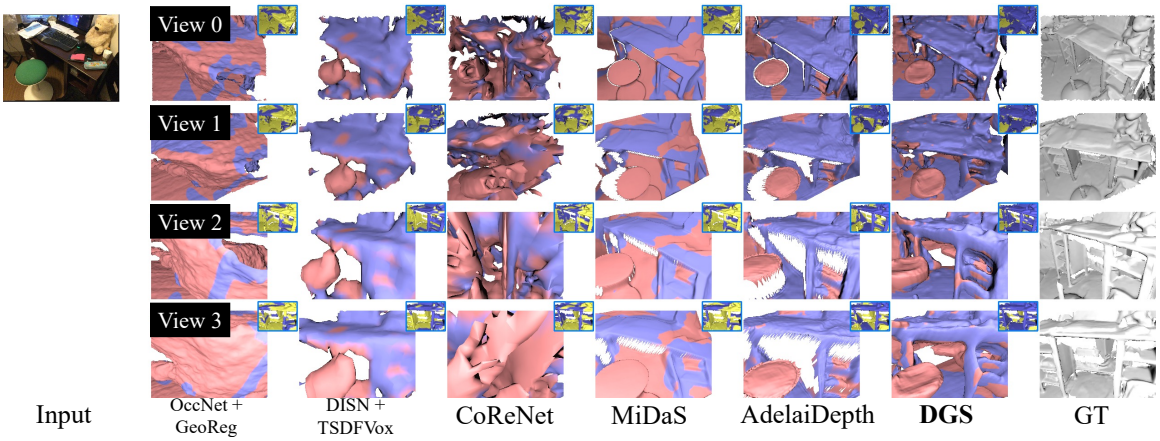


Figure 2.5: Qualitative comparisons on one challenging test case on ScannetV2. For each predicted surface with red and sky-blue colors, sky-blue indicates "positive precision" for that surface region, while red indicates "negative precision". The ground truth surface is shown on the top-right corner 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 the F1 score would be.

Table 2.2: 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**

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 [44] and [43]. 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 [44, 43]. In addition, we also only evaluate the geometries in front of the *amodal* depth for each pixel ray, where only the space in front of the wall, bounded within the ceiling and the floor, is evaluated. 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

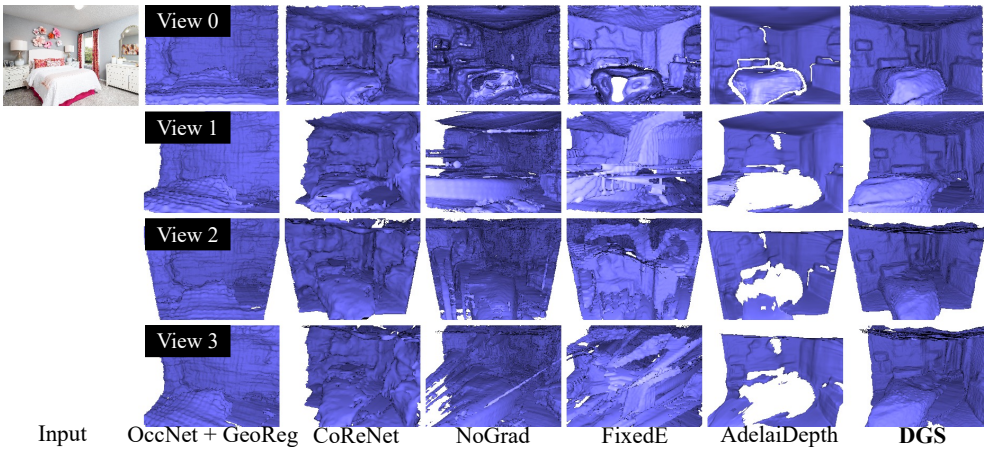


Figure 2.6: Qualitative comparisons on an unseen test image downloaded from the internet.

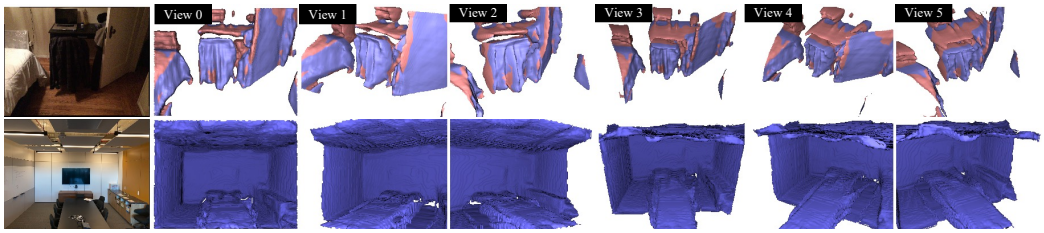


Figure 2.7: Two representative failure cases of our approach.

(1.05 in our case) multiplied with the amodal depth. This evaluation protocol would be equivalent to the "single-layered" protocol used in [44], within our single-view scenario.

Due to the inherent ambiguity of the scaling and shifting of the predicted 3D single view geometry [56, 57], we follow them by computing 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.

Please refer to Sec. 2.6 for details on baselines.

Results. We provide quantitative comparison in Tab. 2.2 and qualitative comparison in Fig. 2.5 respectively. Our model outperforms all state-of-the-art approaches as well as the ablation models. Compared to the synthetic data scenario in Tab. 2.1 and 2.4, our model demonstrates large 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 [58, 57], 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 [45, 44, 43].

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 run our model on Pix3d [59] as well as test images downloaded directly from the internet. We provide a detailed qualitative comparison in Fig. 2.6 and more results in Fig. 2.1. 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. 2.7. 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 small objects (e.g. chairs) and not predict the invisible partition of these objects.

2.4 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-efficient end-to-end training from images and demonstrated single view 3D surface reconstruction results on scenes for the first time, to our knowledge.

2.5 Additional Details and Results on ShapeNet

Additional Quantitative Results. We provide our original results on high-realism ShapeNet in Tab. 2.3. These results are evaluated with the $64 \times 64 \times 64$ volume grids due to an error in our prior code base. Based on the results in both Tab. 2.1 and Tab. 2.3, we found our proposed learning framework performs competitively with the oracle training setting, even though our approach does not utilize dense occupancy labels as used in the oracle state-of-the-art approaches (OccNet, DISN, CoReNet and DISN (ResNet50) + DVR). On the other hand, we observe that our model demonstrates slight improvements when evaluated with the resolution of $64 \times 64 \times 64$, while evaluating the same models with the resolution of $128 \times 128 \times 128$ would reverse the course. In particular, we found categories like "car" exhibited major performance decrease on all the approaches with the higher resolution evaluating setting. This might be due to the fact that these categories demonstrate empty ground truth inside the objects, and and at a higher resolution, the metrics are biased to penalize models

that predict "occupie" inside the object. However, the competitive performance in both resolution settings indicates that our learning with only the surface labels does not deteriorate the performance, and can be faithfully applied toward the real-scene settings where the dense occupancy labels are truly unavailable.

Table 2.3: Intersection over Union % (IoU ↑) benchmarking result on the high-realism ShapeNet with the resolution of $64 \times 64 \times 64$. Please refer to Tab. 2.1 (the resolution of $128 \times 128 \times 128$) for details.

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
DISN(Res50)+DVR	66.9	74.4	67.3	54.5	66.1	63.2	66.2	58.6	71.2	67.2	72.5	63.1	82.6	67.2
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. 2.1	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. 2.1	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
CoReNet w/ Eq. 2.1 (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
DISN(Res50)+DVR w/ Eq. 2.1 (DGS Best)	66.8	75.1	68.7	55.1	65.7	63.7	66.5	58.2	71.5	67.4	72.9	63.5	84.2	67.6

Ablation Study Details. We 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 do not use any 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. 2.1) 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. 2.1. Both experiments are conducted with the high-realism ShapeNet data. We comfortably outperform all the in-house ablation baselines, validating the essential roles of DGS in our learning framework.

Experiments on low-realism ShapeNet. For our low-realism evaluation set-

ting, 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 [2]. 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 [2].

Qualitative Results. We provide qualitative results in Fig. 2.9-2.10 as the additional illustration of the performance of all the approaches on ShapeNet.

Table 2.4: 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^{2}IM-Net) compared to the reported performance from the literatures.

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^{2}IM-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**

2.6 Additional Results on ScannetV2

Baselines. Since we demonstrate the first attempt to predict 3D implicit surface of scenes from a single image, very few exiting works provide a direct baseline performance to our task. For OccNet [14], since its image feature extraction is not local, and its gradient propagation does not require sampling, we use its direct application with [17] "OccNet + GeoReg" as one baseline. We train DISN [23] with the TSDF voxelization labels [43, 44, 45]. For CoReNet [2], 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 [56, 57] for comparison by finetuning their weights on ScannetV2. Lastly, we compare with the two ablation models *NoGrad* and *FixedE* as introduced in Sec. 2.3.1.

Implementation details for DGS. Since the DISN variant architectures [23, 39, 19] 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 [2], we build our model based on the former architectures. Similar to [39, 19], 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 [60] as in [57] 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. 2.5. 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. 2.11-2.13.

Table 2.5: 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

2.7 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. 2.2.2) 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. 2.6. Please find qualitative comparison in Fig. 2.9-2.10.

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. 2.8, 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. 2.6 that the closed-form variant (DGS) achieve better performance.

Table 2.6: Intersection over Union % (IoU ↑) benchmarking comparison between the closed-form solution and the numerical gradients on the high-realism ShapeNet. We report the performance comparison with both the resolution settings of $64 \times 64 \times 64$ and $128 \times 128 \times 128$ (Please refer to Sec. 2.5 for details).

Category	Craft	Rifle	Disp.	Lamp	Spk.	Box	Chair	Bench	Car	Plane	Sofa	Table	Phone	Mean
Numerical (64)	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 (64)	**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**
Numerical (128)	59.7	66.9	61.3	41.8	54.1	48.5	58.2	43.8	**59.8**	59.0	68.5	54.3	77.0	57.9
Closed-form (128)	**61.1**	**67.5**	**62.7**	**44.2**	**54.8**	**49.6**	**59.5**	**45.4**	59.4	**59.9**	**69.8**	**55.1**	**78.0**	**59.0**

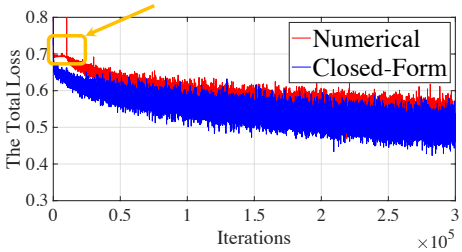


Figure 2.8: Convergence Analysis for the comparison between the closed-form (blue) and the numerical counterpart (red). Notably, the numerical counterpart does not observe loss drop in the first 10k iterations.

2.8 Additional Generalizability Qualitative Results

We provide more qualitative results on our model generalizing to unseen images downloaded from the Internet in Fig. 2.14. 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.

2.9 Evaluation of the Generalizability to Matterport3D dataset

To quantatively evaluate our generaizability to new datasets, we provide our results on the Matterport3D dataset [28] using our model trained from ScannetV2 as used in Sec. 2.3.2.

Table 2.7: 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 [28] 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. 2.7. We can see from the table that the results align with our evaluation in Tab. 2.2 that our model demonstrates clear advantages. We also noticed that the AdelaiDepth [57] 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.

2.10 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. 2.2.2.

In Eq. 2.4, we provided the backward gradient for differentiable sampling for Pixel A. The backward gradient for Pixel B, C and D are

$$\begin{aligned}\frac{\partial L}{\partial \phi_B} &= (1-\alpha)\beta \cdot \frac{\partial L}{\partial \phi(i,j)} \\ \frac{\partial L}{\partial \phi_C} &= \alpha(1-\beta) \cdot \frac{\partial L}{\partial \phi(i,j)} \\ \frac{\partial L}{\partial \phi_D} &= \alpha\beta \cdot \frac{\partial L}{\partial \phi(i,j)}\end{aligned} \tag{2.9}$$

In Eq. 2.5, we provided the forward gradient in the vertical pixel direction. The forward gradient of the horizontal pixel direction is

$$\frac{\partial \phi(i,j)}{\partial j} = \frac{(1-\alpha)(\phi_B - \phi_A) + \alpha(\phi_D - \phi_C)}{w} \tag{2.10}$$

In Eq. 2.6, we provided the backward gradient for only Pixel A. The backward gradient for Pixel B, C and D are

$$\begin{aligned}
\frac{\partial L}{\partial \phi_B} &= (1-\alpha)\beta \cdot \frac{\partial L}{\partial \phi(i,j)} + (-\frac{\beta}{h}) \cdot \frac{\partial L}{\partial \nabla_i \phi(i,j)} + \frac{1-\alpha}{w} \cdot \frac{\partial L}{\partial \nabla_j \phi(i,j)} \\
\frac{\partial L}{\partial \phi_C} &= \alpha(1-\beta) \cdot \frac{\partial L}{\partial \phi(i,j)} + \frac{1-\beta}{h} \cdot \frac{\partial L}{\partial \nabla_i \phi(i,j)} + (-\frac{\alpha}{w}) \cdot \frac{\partial L}{\partial \nabla_j \phi(i,j)} \\
\frac{\partial L}{\partial \phi_D} &= \alpha\beta \cdot \frac{\partial L}{\partial \phi(i,j)} + \frac{\beta}{h} \cdot \frac{\partial L}{\partial \nabla_i \phi(i,j)} + \frac{\alpha}{w} \cdot \frac{\partial L}{\partial \nabla_j \phi(i,j)}
\end{aligned} \tag{2.11}$$

In Eq. 2.8, we provided the backward gradient for only Pixel A in the 3D setting. The backward gradient for Pixel B, C and D are

$$\begin{aligned}
\frac{\partial L}{\partial \phi_B} =& \frac{\partial L}{\partial \phi(x,y,z)} \cdot (1-\alpha)\beta + \frac{\partial L}{\partial \nabla_x \phi(x,y,z)} \cdot \frac{1-\alpha}{w} \cdot \frac{f_0}{z} + \frac{\partial L}{\partial \nabla_y \phi(x,y,z)} \cdot (-\frac{\beta}{h}) \cdot \frac{f_0}{z} \\
&+ \frac{\partial L}{\partial \nabla_z \phi(x,y,z)} \cdot (\frac{\beta}{h} \cdot \frac{y f_0}{z^2} - \frac{1-\alpha}{w} \cdot \frac{x f_0}{z^2}) \\
\frac{\partial L}{\partial \phi_C} =& \frac{\partial L}{\partial \phi(x,y,z)} \cdot \alpha(1-\beta) + \frac{\partial L}{\partial \nabla_x \phi(x,y,z)} \cdot (-\frac{\alpha}{w}) \cdot \frac{f_0}{z} + \frac{\partial L}{\partial \nabla_y \phi(x,y,z)} \cdot \frac{1-\beta}{h} \cdot \frac{f_0}{z} \\
&+ \frac{\partial L}{\partial \nabla_z \phi(x,y,z)} \cdot (-\frac{1-\beta}{h} \cdot \frac{y f_0}{z^2} + \frac{\alpha}{w} \cdot \frac{x f_0}{z^2}) \\
\frac{\partial L}{\partial \phi_D} =& \frac{\partial L}{\partial \phi(x,y,z)} \cdot \alpha\beta + \frac{\partial L}{\partial \nabla_x \phi(x,y,z)} \cdot \frac{\alpha}{w} \cdot \frac{f_0}{z} + \frac{\partial L}{\partial \nabla_y \phi(x,y,z)} \cdot \frac{\beta}{h} \cdot \frac{f_0}{z} \\
&+ \frac{\partial L}{\partial \nabla_z \phi(x,y,z)} \cdot (-\frac{\beta}{h} \cdot \frac{y f_0}{z^2} - \frac{\alpha}{w} \cdot \frac{x f_0}{z^2})
\end{aligned} \tag{2.12}$$

As an illustration of the derivation for Eq. 2.8, we can obtain the backward gradient via

$$\frac{\partial L}{\partial \phi_A} = \frac{\partial L}{\partial \phi} \cdot \frac{\partial \phi}{\partial \phi_A} + \frac{\partial L}{\partial \nabla_x \phi} \cdot \frac{\partial \nabla_x \phi}{\partial \phi_A} + \frac{\partial L}{\partial \nabla_y \phi} \cdot \frac{\partial \nabla_y \phi}{\partial \phi_A} + \frac{\partial L}{\partial \nabla_z \phi} \cdot \frac{\partial \nabla_z \phi}{\partial \phi_A} \tag{2.13}$$

The four partial derivatives with respect to ϕ_A in Eq. 2.13 can be determined based on the sampling rules. For example, based on Eq. 2.3, we can quickly obtain the first derivative with respect to ϕ_A via

$$\frac{\partial \phi}{\partial \phi_A} = (1-\alpha)(1-\beta) \tag{2.14}$$

For the other three derivatives with respect to ϕ_A in Eq. 2.13, we can substitute Eq. 2.5, 2.10 into Eq. 2.7, and obtain them via

$$\begin{aligned}
\frac{\partial \nabla_x \phi}{\partial \phi_A} &= -\frac{f_0}{z} \cdot \frac{1-\alpha}{w} \\
\frac{\partial \nabla_y \phi}{\partial \phi_A} &= -\frac{f_0}{z} \cdot \frac{1-\beta}{h} \\
\frac{\partial \nabla_z \phi}{\partial \phi_A} &= \frac{y f_0}{z^2} \cdot \frac{1-\beta}{h} + \frac{x f_0}{z^2} \cdot \frac{1-\alpha}{w}
\end{aligned} \tag{2.15}$$

We can then obtain Eq. 2.8 via substituting Eq. 2.14, 2.15 into Eq. 2.13.

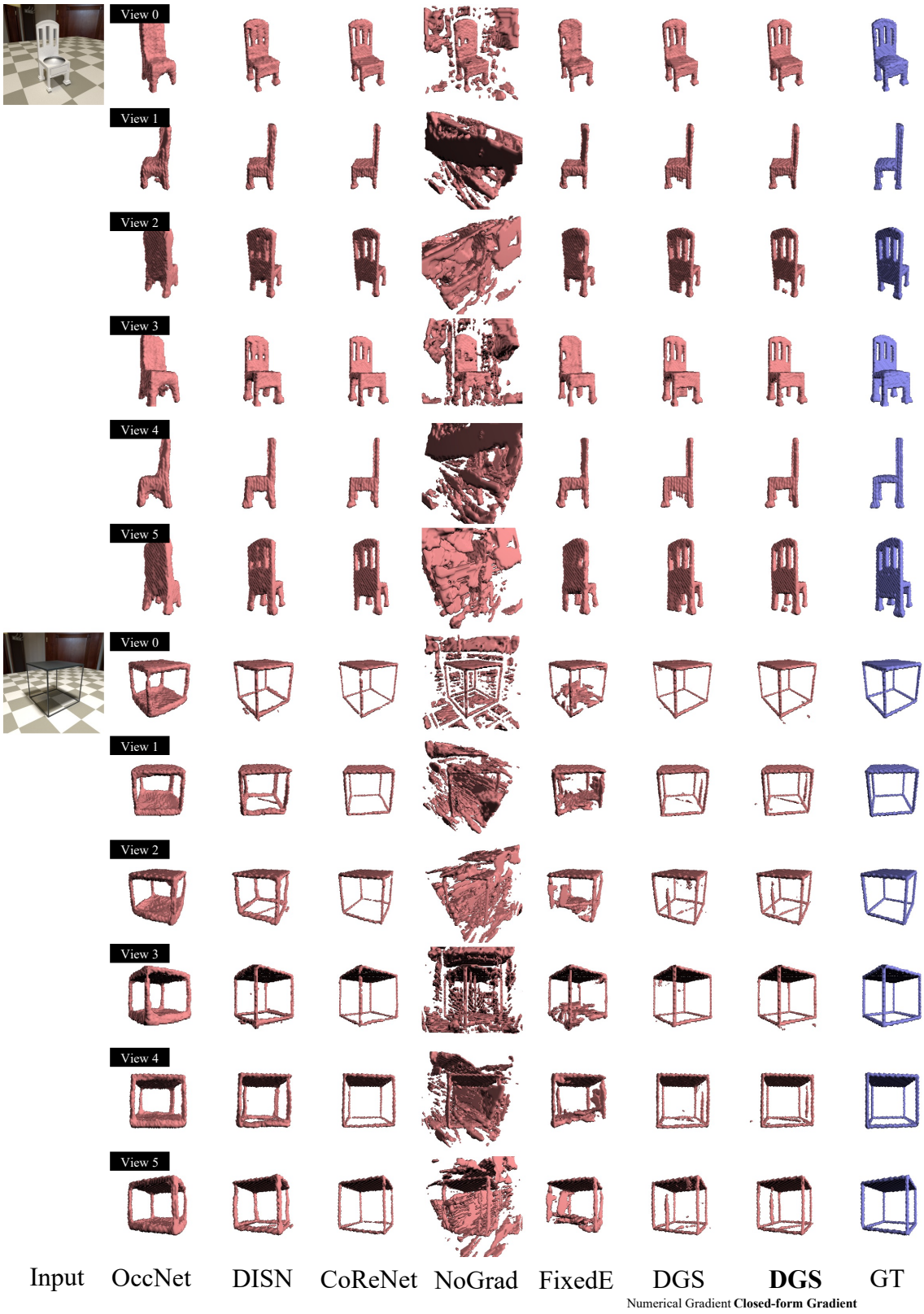


Figure 2.9: Quantitative comparison on the high-realism ShapeNet (without handpick: test case number 0 and 100). The reconstruction result of each approach is visualized in six different views, with the first view the same as the camera view, the first three views the same elevation as the camera view, and the last three view horizontal view.

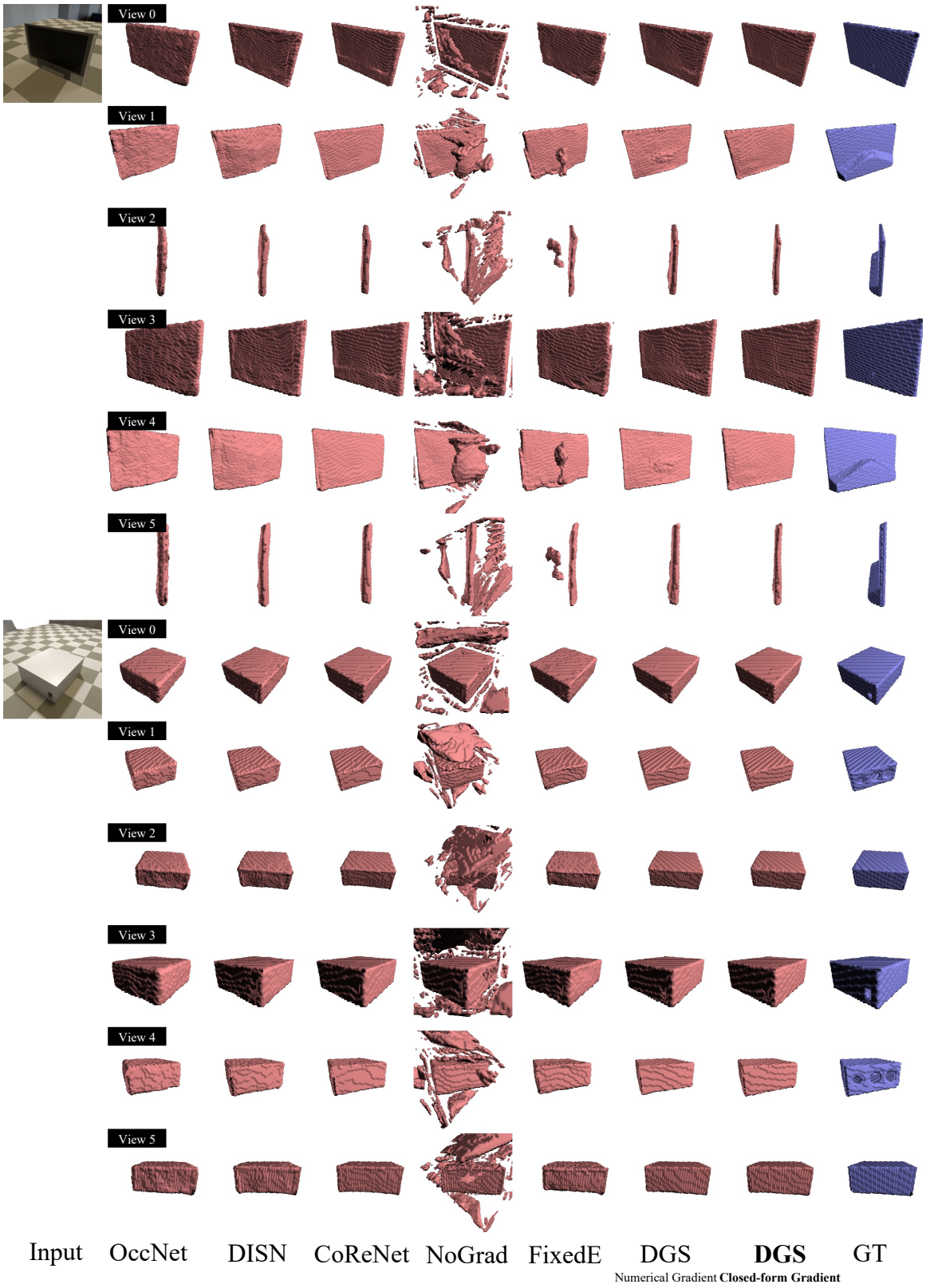


Figure 2.10: Quantitative comparison on the high-realism ShapeNet (without handpick: test case number 800 and 900). The reconstruction result of each approach is visualized in six different views, with the first view the same as the camera view, the first three views the same elevation as the camera view, and the last three view horizontal view.

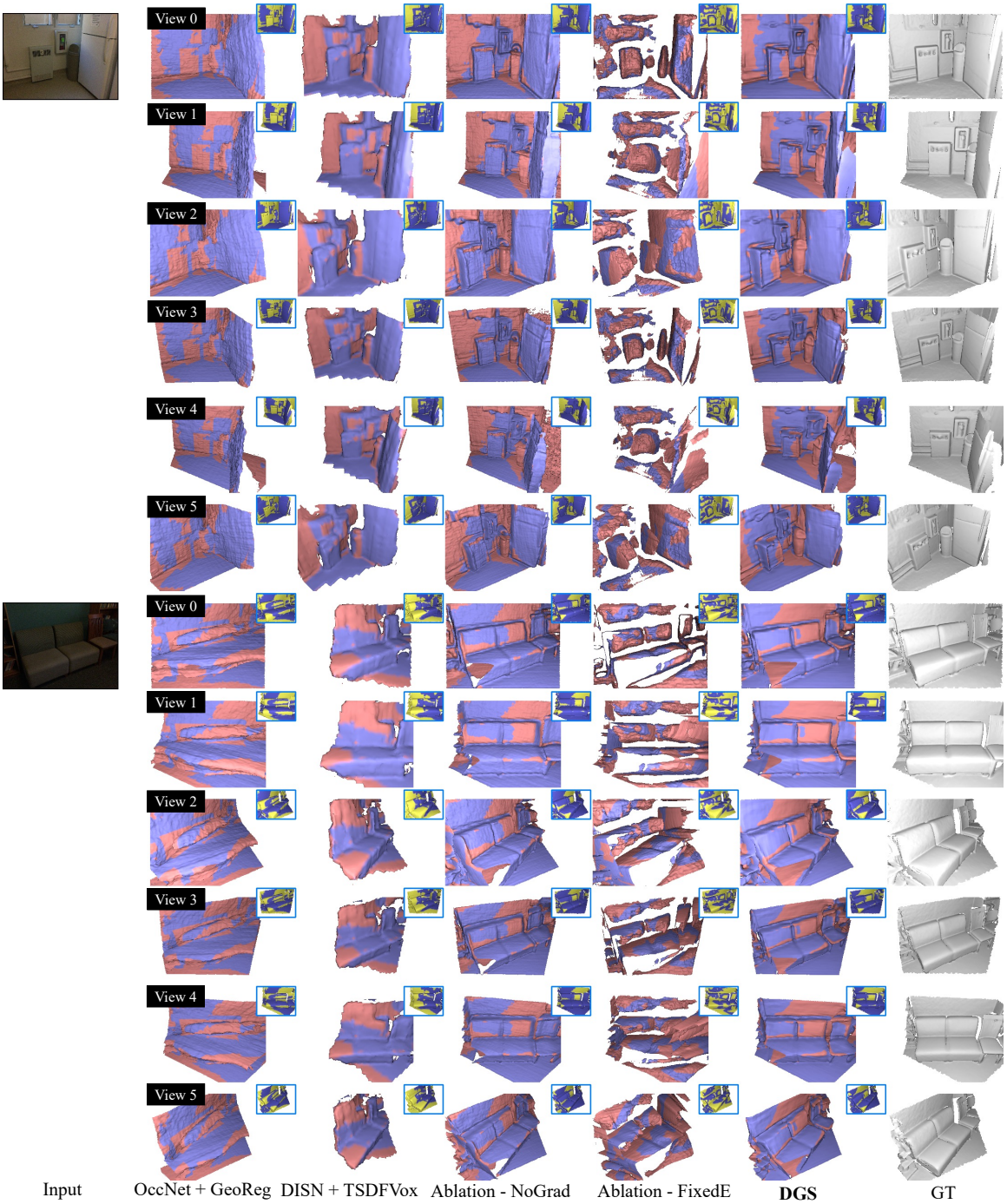


Figure 2.11: Quantitative comparison on the ScannetV2 (without handpick: the first frame of the 1st and 2nd test scene in ScannetV2). The reconstruction result of each approach is visualized in six different views, with the first view the same as the camera view, the first three views the same elevation as thecamera view, and the last three view elevated view.

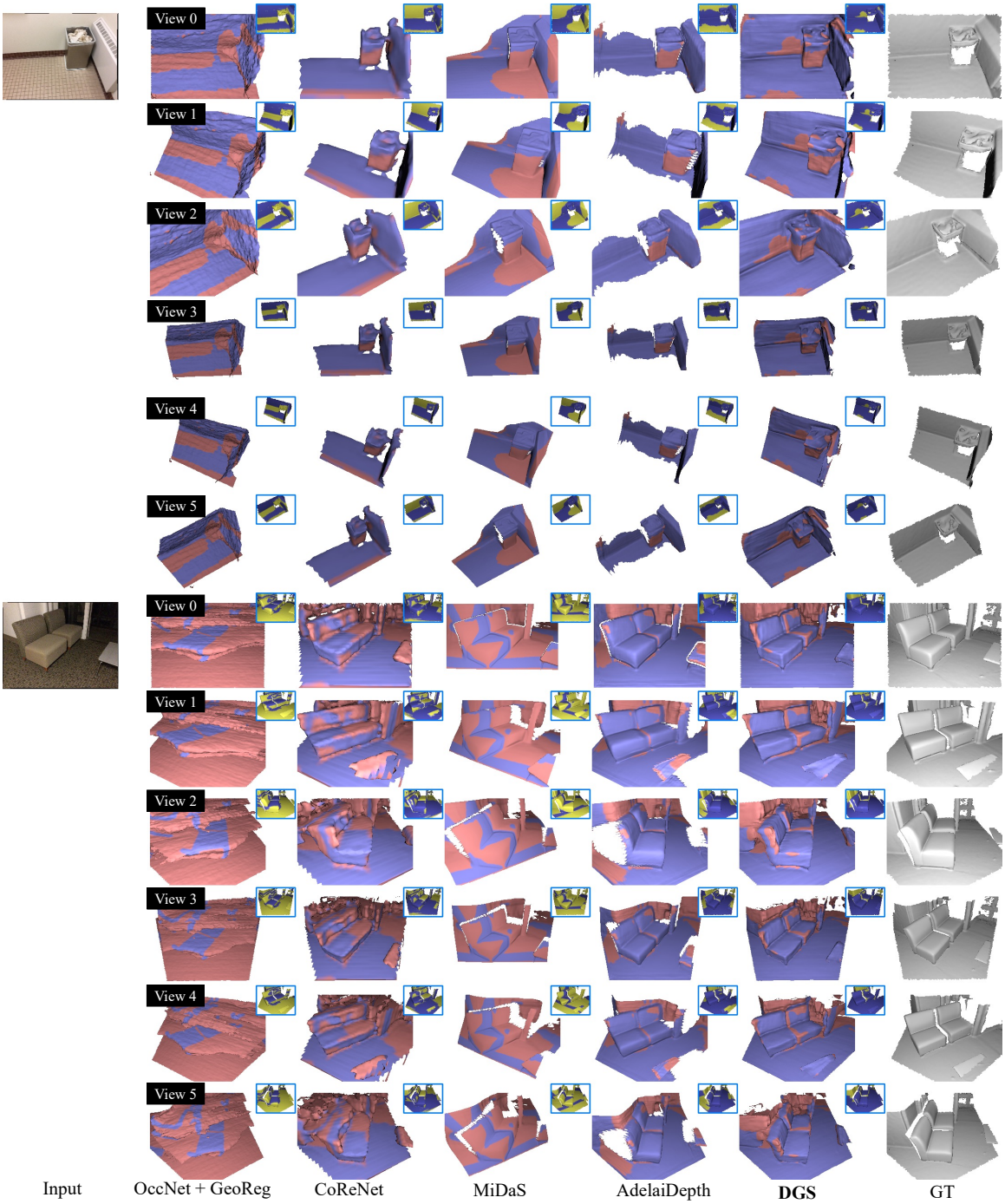


Figure 2.12: Quantitative comparison on the ScannetV2 (without handpick: the first frame of the 7th and 8th test scene in ScannetV2). The reconstruction result of each approach is visualized in six different views, with the first view the same as the camera view, the first three views the same elevation as thecamera view, and the last three view elevated view.

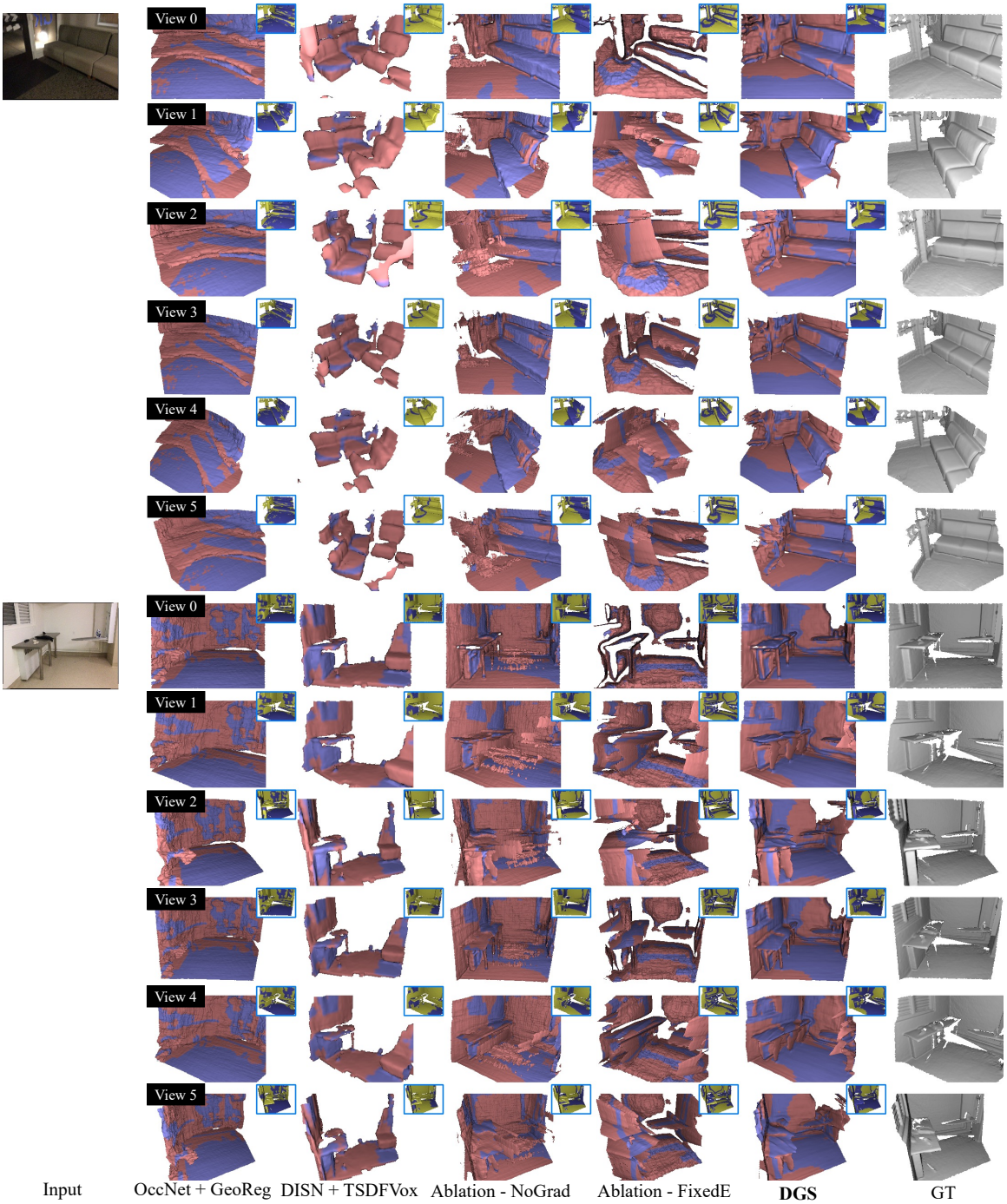


Figure 2.13: Quantitative comparison on the ScannetV2 (without handpick: the first frame of the 9th and 10th test scene in ScannetV2). The reconstruction result of each approach is visualized in six different views, with the first view the same as the camera view, the first three views the same elevation as thecamera view, and the last three view elevated view.

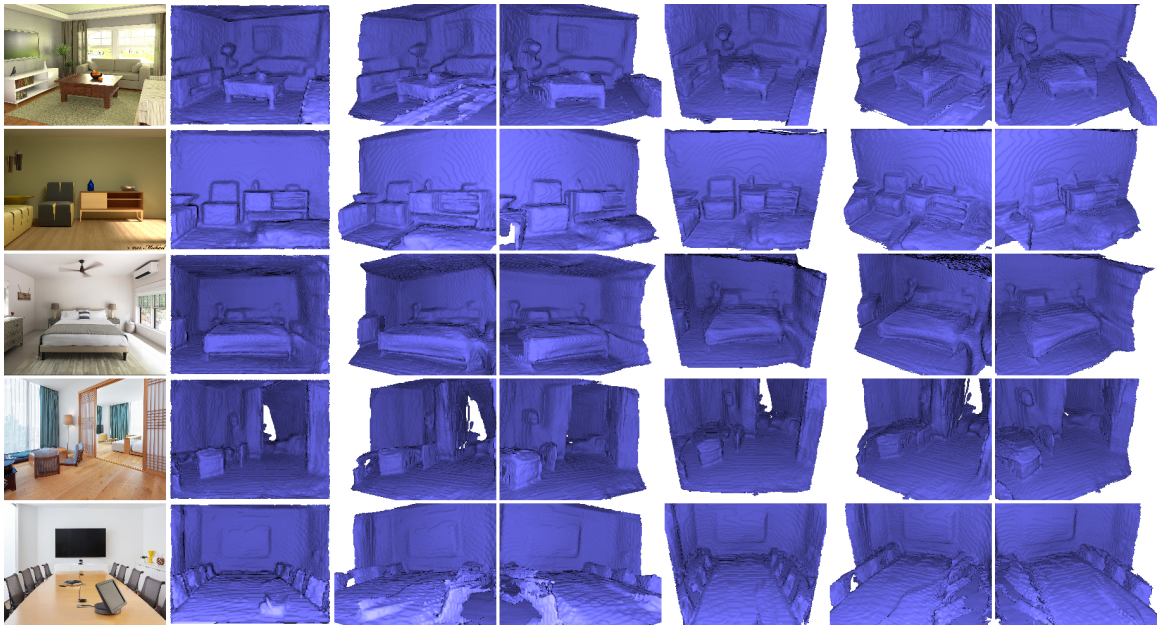


Figure 2.14: Additional Qualitative results of our model generalizing to unseen test images downloaded from the Internet.

Chapter 3

Voxel-informed Language Grounding

3.1 Introduction

Embodied robotic agents hold great potential for providing assistive technologies in home environments [61], and natural language provides an intuitive interface for users to interact with such systems [62]. For these systems to be effective, they must be able to reliably ground language in perception [63, 64].

Despite typically being paired with 2D images, natural language that is grounded in vision describes a fundamentally 3D world. For example, consider the grounding task in Figure 3.1, where the agent must select a target chair against a distractor given the description "the swivel chair with 6 wheels." Although the agent is provided with multiple images revealing all of the wheels on each chair, it must be able to properly aggregate information across images to successfully differentiate them, something that requires reasoning about their *3D geometry* at some level.

In this work, we show how language grounding performance may be improved by leveraging 3D prior knowledge. Our model, Voxel-informed Language Grounder (VLG), extracts 3D voxel maps using a pre-trained *volumetric reconstruction model*, which it fuses with multimodal features from a large-scale vision and language model in order to reason jointly over the visual and 3D geometric properties of objects.

We focus our investigation within the context of SNARE [11], an object reference game where an agent must ground natural language describing common household objects by their geometric and visual properties, showing that grounding accuracy significantly improves by incorporating information from predicted 3D volumes of objects. At the time of writing, VLG achieves SOTA performance on SNARE, attaining an absolute improvement of 2.0% over the next closest baseline. Code to replicate

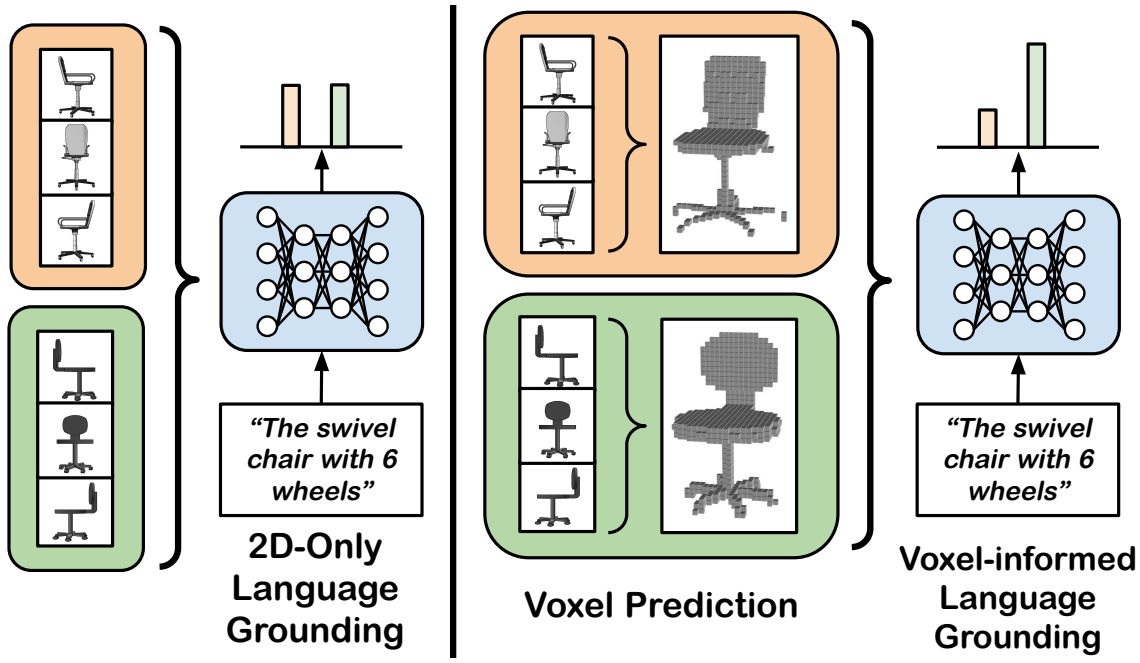


Figure 3.1: Voxel-informed Language Grounder. Our VLG model leverages explicit 3D information by inferring volumetric voxel maps from input images, allowing the agent to reason jointly over the geometric and visual properties of objects when grounding.

our results is publicly available.[1]

3.2 Related Work

Prior work has studied deriving structured representations from images to scaffold language grounding. However, a majority of systems use representations such as 2D regions of interest [65, 66] or symbolic graph-based representations [67, 68], which do not encode 3D properties of objects.

Most prior work tying language to 3D representations has largely focused on generating 3D structures conditioned on language, rather than using them as intermediate representations for language grounding as we do here. Specifically, prior work has performed language conditioned generation at the scene [69, 36], pose [70, 71], or object [72] level. More recently, a line of work has explored referring expression grounding in 3D by mapping referring expressions of objects to 3D bounding boxes localizing them in point clouds of indoor scenes [73, 74, 75, 76]. Standard approaches follow a two-tiered process where an object proposal system will first provide bounding boxes for candidate objects, and a scoring module will then compute a compatibility score between each box and the referring expression in order to ground it. At a more

[1] `https://github.com/rcorona/voxel_informed_language_grounding`

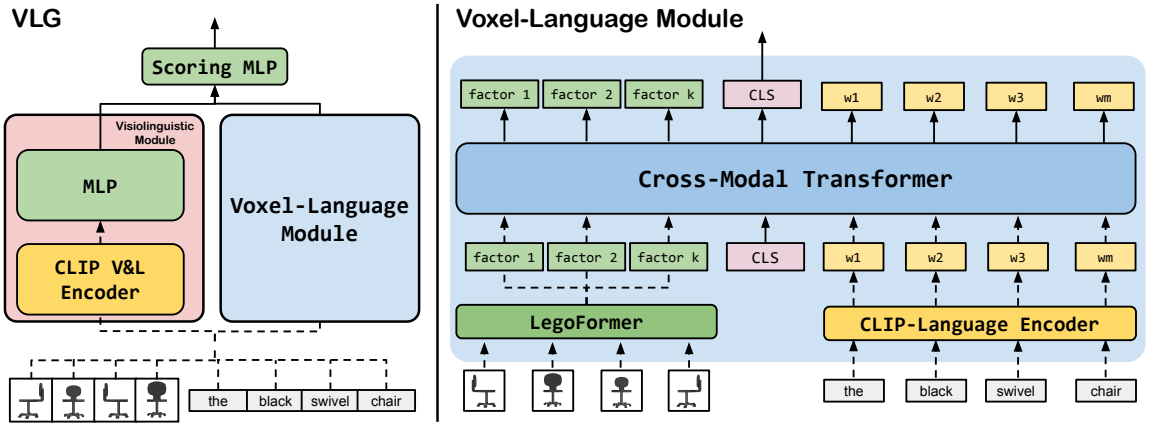


Figure 3.2: VLG Architecture. (Left) Our VLG model consists of a visiolinguistic module which produces a joint embedding for text and images using CLIP [3] and a voxel-language module for jointly embedding language and volumetric maps. (Right) The voxel-language module uses a cross modal transformer to fuse word embeddings from CLIP with voxel map factors extracted from LegoFormer [4]. During training, gradients only flow through solid lines.

granular level, [77] learn alignments from language to object parts by training agents on a reference game over point cloud representations of objects.

In contrast, in this work we focus on augmenting language grounding over 2D RGB images using structured 3D representations derived from them. For the task of visual language navigation, prior work has shown how a persistent 3D semantic map may be used as an intermediate representation to aid in selecting navigational waypoints [78, 79]. The semantic maps, however, represent entire scenes with individual voxels representing object categories, rather than their geometry. In this work, we show how a more granular occupancy map representing objects' geometry can improve language grounding performance.

Closest to our work is that of [80], which presents a method for mapping language to 3D features within scenes from the CLEVR [81] dataset. Their system generates 3D feature maps inferred from images and then grounds language directly to 3D bounding boxes or coordinates. Their method assumes, however, that dependency parse trees are provided for the natural language inputs, and it is trained with supervised alignments between noun phrases and the 3D representations, which VLG does not require.

3.3 Voxel-informed Language Grounder

We consider a task where an agent must correctly predict a target object v^t against a distractor v^c given a natural language description $w^t = \{w_1, ..., w_m\}$ of the target. For each object, the agent is provided with n 2D views $v = \{x_1, ..., x_n\}$, $x_i \in \mathbb{R}^{3\times W\times H}$.

An agent for this task is represented by a scoring function $s(v, w) \in [0, 1]$, computing the compatibility between the target description and the 2D views of each object, and is used to select the maximally scoring candidate. We first use unimodal encoders to encode the language description into $e_w = h(w)$ and the object view images into a single aggregate visual embedding $e_v = g(v)$ before fusing them with a visiolinguistic module $e_{vw} = f_{vw}([e_v; e_w])$. Prior approaches to this problem [11] directly input this fused representation to a scoring module to produce a score $s(e_{vw})$. They do not explicitly reason about the 3D properties of the observed objects, requiring the models to learn them implicitly.

In contrast, our Voxel-informed Language Grounder augments the scoring function s with explicit 3D volumetric information $e_o = o(v)$ extracted from a pre-trained multiview reconstruction model. The volumetric information (in the form of a factorization of a voxel occupancy map in $\mathbb{R}^{W \times H \times D}$) is first fused into a joint representation with the language using a multimodal voxel-language module $e_{ow} = f_{ow}([e_o; e_w])$. The scoring function then produces a score based on all three modalities $s([e_{vw}; e_{ow}])$.

3.3.1 Model Architecture

VLG (Figure 3.2) consists of two branches: a visiolinguistic module for fusing language and 2D RGB features, and a voxel-language module for fusing language with 3D volumetric features. A scoring function is then used to reason jointly over the output of the two branches, producing a compatibility score.

Visiolinguistic Module. The architecture of our visiolinguistic module f_{vw} (left panel, Figure 3.2) largely mirrors the architecture of MATCH from [11]. A pre-trained CLIP-ViT [3] model is used to encode the language description and view images into vectors in $\mathbb{R}^{512}$. The image embeddings are max-pooled and concatenated to the description embedding before being passed into an MLP which generates a fused representation.

Voxel-Language Module. We use representations extracted from a ShapeNet [82, 83] pre-trained LegoFormerM [4], a multi-view 3D volumetric reconstruction model, as input to our voxel-language module f_{ow}. LegoFormer is a transformer [84] based model whose decoder generates volumetric maps factorized into 12 parts. Each object factor is represented by a set of three vectors $x, y, z \in \mathbb{R}^{32}$, which we concatenate to use as input tokens for our voxel-language module. A triple cross-product over x, y, z may be used to recover a 3D volume $\mathcal{V} \in \mathbb{R}^{32 \times 32 \times 32}$ for each factor. The full volume for the object is generated by aggregating the factor volumes through a sum operation. For more details on LegoFormer, we refer the reader to [4]. We use a cross-modal transformer [84] encoder to fuse the language and object factors (Figure 3.2, right). The cross-modal transformer takes as input language tokens, in the form of CLIP word embeddings, and the 12 object factors output by the LegoFormer

decoder, which contain the inferred geometric occupancy information of the object. We use a CLS token as an aggregate representation of the language and object factors.

Scoring Function. The scoring function is represented by an MLP which takes as input the concatenation of the visiolinguistic module output and the cross-modal transformer's CLS token.

3.4 Language Grounding Evaluation

	VALIDATION			TEST		
Model	Visual	Blind	All	Visual	Blind	All
ViLBERT	89.5	76.6	83.1	80.2	**73.0**	76.6
MATCH	89.2 (0.9)	75.2 (0.7)	82.2 (0.4)	83.9 (0.5)	68.7 (0.9)	76.5 (0.5)
MATCH*	90.6 (0.4)	75.7 (1.2)	83.2 (0.8)	-	-	-
LAGOR	89.8 (0.4)	75.3 (0.7)	82.6 (0.4)	84.3 (0.4)	69.4 (0.5)	77.0 (0.5)
LAGOR*	89.8 (0.5)	75.0 (0.4)	82.5 (0.1)	-	-	-
VLG (Ours)	**91.2** (0.4)	**78.4**†(0.7)	**84.9**†(0.3)	**86.0**	71.7	**79.0**

Table 3.1: SNARE Benchmark Performance. Object reference game accuracy on the SNARE task across validation and test sets. Performance on models with an asterisk are our replications of the baselines in [11]. Standard deviations over 3 seeds are shown in parentheses. MATCH corresponds to the max-pool variant from [11] since no test set results are provided for the mean-pool variant. Our VLG model achieves the best overall performance. Due to leaderboard submission restrictions, we were not able to get test set results for the MATCH* and LAGOR* replications. † denotes statistical significance over replicated models with $p < 0.1$.

Evaluation. We test our method on the SNARE benchmark [11]. SNARE is a language grounding dataset which augments ACRONYM [85], a grasping dataset built off of ShapeNetSem [86, 36], with natural language annotations of objects.

SNARE presents an object reference game where an agent must correctly guess a target object against a distractor. In each instance of the game, the agent is provided with a language description of the target as well as multiple 2D views of each object. SNARE differentiates between **visual** and **blindfolded** object descriptions. Visual descriptions primarily include attributes such as *name*, *shape*, and *color* (e.g. "classic armchair with white seat"). In contrast, blindfolded descriptions include attributes such as *shape* and *parts* (e.g. "oval back and vertical legs"). The train/validation/test sets were generated by splitting over (207 / 7 / 48) ShapeNetSem object categories, respectively containing (6,153 / 371 / 1,357) unique object instances and (39,104 / 2,304 / 8,751) object pairings with referring expressions. Renderings are provided for each object instance over 8 canonical viewing angles.

Because ShapeNet and ShapeNetSem represent different splits of the broader ShapeNet database, we pre-train the LegoFormerM model on a modified dataset

to avoid dataset leakage. Specifically, any objects which appear in both datasets are re-assigned within the pre-training dataset used to train LegoFormerM to match its split assignment from SNARE.

ShapeNetSem images are resized to 224×224 when inputting them to LegoFormerM in order to match its ShapeNet pre-training conditions.

Baselines. We compare VLG against the set of models provided with SNARE.[2] All SNARE baselines except **ViLBERT** use a CLIP ViT-B/32 [3] backbone for encoding both images and language descriptions:

> **MATCH** first uses CLIP-ViT to embed the language description as well as each of the 8 view images. Next, the view embeddings are mean-pooled and concatenated to the description embedding. Finally, a learned MLP is used over the concatenated feature vector in order to produce a final compatibility score.
>
> **ViLBERT** fine-tunes a 12-in-1 [87] pre-trained ViLBERT[88] as the backbone for MATCH instead of using CLIP-ViT. Each object is presented to ViLBERT in the form of a single tiled image containing all 14 views from ShapeNetSem, instead of just the canonical 8 presented in the standard task. ViLBERT tokenizes images by extracting features from image regions, with the ground truth bounding boxes for each region (i.e. view) being provided. Because this baseline is not open-source, we report the original numbers from [11].
>
> **LAGOR** (**La**nguage **G**rounding through **O**bject **R**otation) fine-tunes a pre-trained MATCH module and is additionally regularized through the auxiliary task of predicting the canonical viewing angle of individual view images, which it predicts using an added output MLP head. Following [11], the LAGOR baseline is only provided with 2 random views of each object both during training and inference.

For more details on the baseline models, we refer the reader to [11].

Training Details. Apart from the dataset split re-assignments mentioned in Section 3.4, we use the code[3] and hyperparameters presented by [4] to train LegoFormerM.

For training on SNARE, we follow [11] and train all models with a smoothed binary cross-entropy loss [89].

We train each model for 75 epochs, reporting performance of the best performing checkpoint on the validation set. For our replication of the SNARE MATCH and LAGOR baselines, we use the code and hyperparameters provided by [11]. For all variants of our VLG model we use the AdamW [90] optimizer with a learning rate of 1e-3 and a linear learning rate warmup of 10K steps.

[2] `https://github.com/snaredataset/snare`

[3] `https://github.com/faridyagubbayli/LegoFormer`

Model	Visual	Blind	All
VGG16	**91.4** (0.5)	76.5 (0.9)	84.0 (0.2)
MLP	91.1 (0.8)	77.9 (0.9)	84.6 (0.1)
no-CLIP	71.0 (0.6)	65.8 (0.7)	68.4 (0.1)
VLG	91.2 (0.4)	**78.4** (0.7)	**84.9** (0.3)

Table 3.2: Ablation Study. SNARE reference game accuracy across ablations of our model on the validation set. We show performance when replacing LegoformerM object factors with **VGG16** features, replacing the cross-modal transformer with an **MLP**, and when foregoing the use of CLIP features (**no-CLIP**).

3.5 Results

We present test set performance for VLG and the SNARE baselines reported by [11]. We also present average performance for trained models over 3 seeds with standard deviations on the validation set.

3.5.1 Comparison to SOTA

In Table 3.1 we can observe reference game performance for all models. VLG achieves SOTA performance with an absolute improvement on the test set of 2.0% over LAGOR, the next best leaderboard model. Although there is a general improvement of 1.7% in **visual** reference grounding, there is an improvement of 2.3% in **blindfolded** (denoted as **Blind** in tables to conserve space) reference grounding. This suggests that the injected 3D information provides a greater boost for disambiguating between examples referring to geometric properties of target objects. VLG generally improves over all baselines and conditions for blindfolded examples, with the exception of ViLBERT, which may be due to the additional information ViLBERT receives in the form of 14 viewing angles of each object instead of 8. Improvements on the Blind and All conditions of the validation set are statistically significant over replicated models with $p < 0.1$ under a Welch's two-tailed t-test.

3.5.2 Ablation Study

We present a variety of ablations on the validation set to investigate the contributions of each piece of our model. All results can be observed in Table 3.2.

VGG16 Embeddings. LegoFormer uses an ImageNet [91] pre-trained VGG16 [54] as a backbone for extracting visual representations, which is a different dataset and pre-training task than what the CLIP-ViT image encoder is trained on. This presents a confounding factor which we ablate by performing an experiment feeding our model's scoring function VGG16 features directly instead of LegoFormer object factors (VGG16 in Table 3.2). Despite getting comparable results to VGG16 on visual

reference grounding, VLG provides a clear improvement in blindfolded (and therefore overall) reference performance, suggesting that the extracted 3D information is useful for grounding more geometrically based language descriptions, with the VGG16 features being largely redundant in terms of visual signal.

Architecture. We ablate the contribution of our cross-modal transformer branch by comparing it against an MLP mirroring the structure of the SNARE MATCH baseline. This model (MLP in Table 3.2) max-pools the LegoFormer object factors and concatenates the result to the CLIP visual and language features before passing them to an MLP scoring function. The MLP model overall outperforms the SNARE baselines from Table 3.1, highlighting the usefulness of the 3D information for grounding, but does not result in as large an improvement as the cross-modal transformer. This suggests that the transformer is better able at integrating information from the multi-view input.

CLIP Visual Embeddings. Finally, we evaluate the contribution of the visiolinguistic branch of the model by removing it and only using the cross-modal transformer over language and object factors. As may be observed, there is a large drop in performance (16.5% overall), particularly for visual references (20.2%). These results suggest that maintaining visual information such as color and texture is critical for performing well on this task, since the LegoFormer outputs contain only volumetric occupancy information.

3.6 Discussion

We have presented the Voxel-informed Language Grounder (VLG), a model which leverages explicit 3D information from predicted volumetric voxel maps to improve language grounding performance. VLG achieves SOTA results on SNARE, and ablations demonstrate the effectiveness of using this 3D information for grounding. We hope this paper may inspire future work on integrating structured 3D representations into language grounding tasks.

Chapter 4

Neural Relighting with Subsurface Scattering Effects

4.1 Introduction

The ability to relight objects and scenes under varying lighting conditions is crucial in many areas, such as virtual reality, gaming, visual effects, and architecture. It enables artists, designers, and engineers to experiment with many lighting setups without having to physically recreate a scene. It also allows for the creation of more realistic and immersive experiences by accurately simulating the lighting conditions in a virtual environment.

However, relighting remains a challenging task due to the complex interaction between the light and the materials in a scene. Traditional approaches have sought to decompose rendering into geometry, material, and lighting to simplify the problem. For example, opaque materials are represented in Physically Based Rendering (PBR) by the Bidirectional Reflectance Distribution Function (BRDF), which describes how light interacts with a material's surface [92, 93, 94, 95, 96]. Similarly, many relighting methods such as [97, 98] rely on decomposing light into its components, such as direct lighting and indirect lighting, to allow for more fine-grained control.

While these approaches have been successful in many cases, they are limited in their ability to handle objects with translucency or subsurface scattering (SSS). This is because these materials are not well approximated by a simple BRDF function and require more complex models, such as the Bidirectional Surface Scattering Reflectance Distribution Function (BSSRDF). However, modeling BSSRDF are computationally expensive and slow to evaluate, neglecting textures beneath the surface (Fig. 4.3), limiting their practicality for inverse rendering with complex geometry.

Recent works on neural radiance transfer fields [6] have incorporated the idea

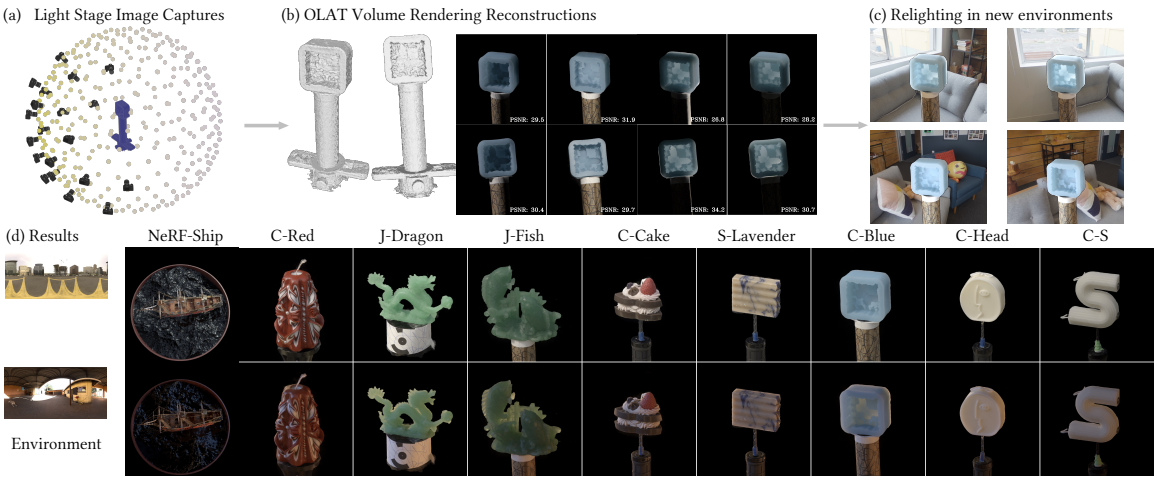


Figure 4.1: Our approach reconstructs objects with significant subsurface scattering effects with high fidelity and inserts models into arbitrary environments for relighting. It is fully data-driven and does not assume a particular material representation (such as BRDF or BSSRDF), and can faithfully render high quality appearance under varying lighting conditions and view points. Please see our supplementary video for comprehensive visualizations and comparisons.

of pre-computed radiance transfer (PRT) into the neural radiance fields (NeRF) literature, providing promising results for relighting with global illumination effects. However, these approaches rely on a pre-estimated surface, which is nontrivial to reconstruct for objects with SSS or with translucency. Additionally, the separated geometry and appearance optimization is suboptimal, leading to artifacts and unrealistic results.

In this work, we propose a novel framework for relighting that incorporates the optimization of shape and radiance transfer using a volume rendering approach (Fig. 1). Our framework extends the relighting capability to a wider range of materials, including translucent objects with strong SSS effects and textures beneath the surface. Specifically, we use a volume rendering approach to estimate the transfer field and utilize appearance cues to refine the geometry in an end-to-end fashion.

To evaluate our approach, we have recorded real-world objects featuring subsurface scattering effects in a light stage and show that our method produces high quality visual results in recorded and novel lighting conditions. Quantitatively, our approach compares favorably with the current state of the art with a 5 points higher PSNR on average across three datasets.

In summary, we propose a novel framework for neural radiance transfer fields using volume rendering, optimizing appearance and geometry in an end-to-end fashion, which to the best of our knowledge has not been achieved before for optically-thick translucent materials. Additionally, we collected and will release a dataset of objects that exhibit prominent subsurface scattering effects for training and evaluation purposes. These objects have been recorded with high fidelity featuring rich, high

frequency spatially-varying details, resulting in 15TiB of data, which is 3000 times larger and notably more detailed than the current highest quality data for research in this area [9].

4.2 Related Work

Relighting and Surface Representations. The problem of relighting an object or a scene under novel lighting conditions has been extensively studied. Usually, the problem is tackled via decomposing the appearance into the lighting and the surface material properties. Early works estimate material given known illumination such as a single light source [99, 100] or spherical gradient illumination [101, 102] with known geometry. [94] directly model light transports with known illuminations and know geometry. More recently, neural scene representations [103] and differentiable rendering [104] allow us to jointly optimize BRDF and geometry. Some methods apply inverse rendering using implicit surface to obtain materials [105, 8, 93]. Other approaches utilize volumetric representations with opacity fields [106, 107, 92, 94, 96]. The required illumination setup can be reduced to a co-located light [106, 107], and unknown illuminations [105, 8, 92, 94, 96]. To reduce the ambiguity in BRDF, the aforementioned methods use parametric BRDFs such as a microfacet model [108, 109]. However, these parametric models do not support subsurface scattering as they only consider reflectance. In contrast, our approach deals with global light transport effects including subsurface scattering.

Subsurface Scattering. Subsurface scattering refers to light transport inside of a solid substance. It happens with some particular types of materials (such as wax, jade, tiny furs or various fruits), and is quite common in the real world. Since the light might leave the object surface at a different point from where it enters, surface representations (e.g. various BRDFs) cannot represent this type of light transmission. While subsurface scattering can be accurately modeled by volumetric path tracing algorithms [110], their run time is typically prohibitive in certain applications, despite efforts to accelerate brute-force computation, e.g. through a shape adaptive learned SSS model [111] that relies on a conditional variational auto encoder that learns to sample from a distribution of exit points on the object surface. Some other works have focused on estimating the scattering parameters from images of translucent objects. Inverse Transport Networks [112] infer the optical properties that control subsurface scattering inside translucent objects of any shape under any illumination. They rely on an encoder decoder where the latter is replaced by a physically-based differentiable path tracer, trained with synthetic images. Prior to that, another approach based on stochastic gradient descent, combined with Monte Carlo rendering and a material dictionary was capable of estimating the scattering materials, inverting the radiative transfer parameters [113]. Nevertheless, since volumetric path tracing can be costly, applying a BSSRDF can be a faster alternative [9]. Compared to BRDF-based rep-

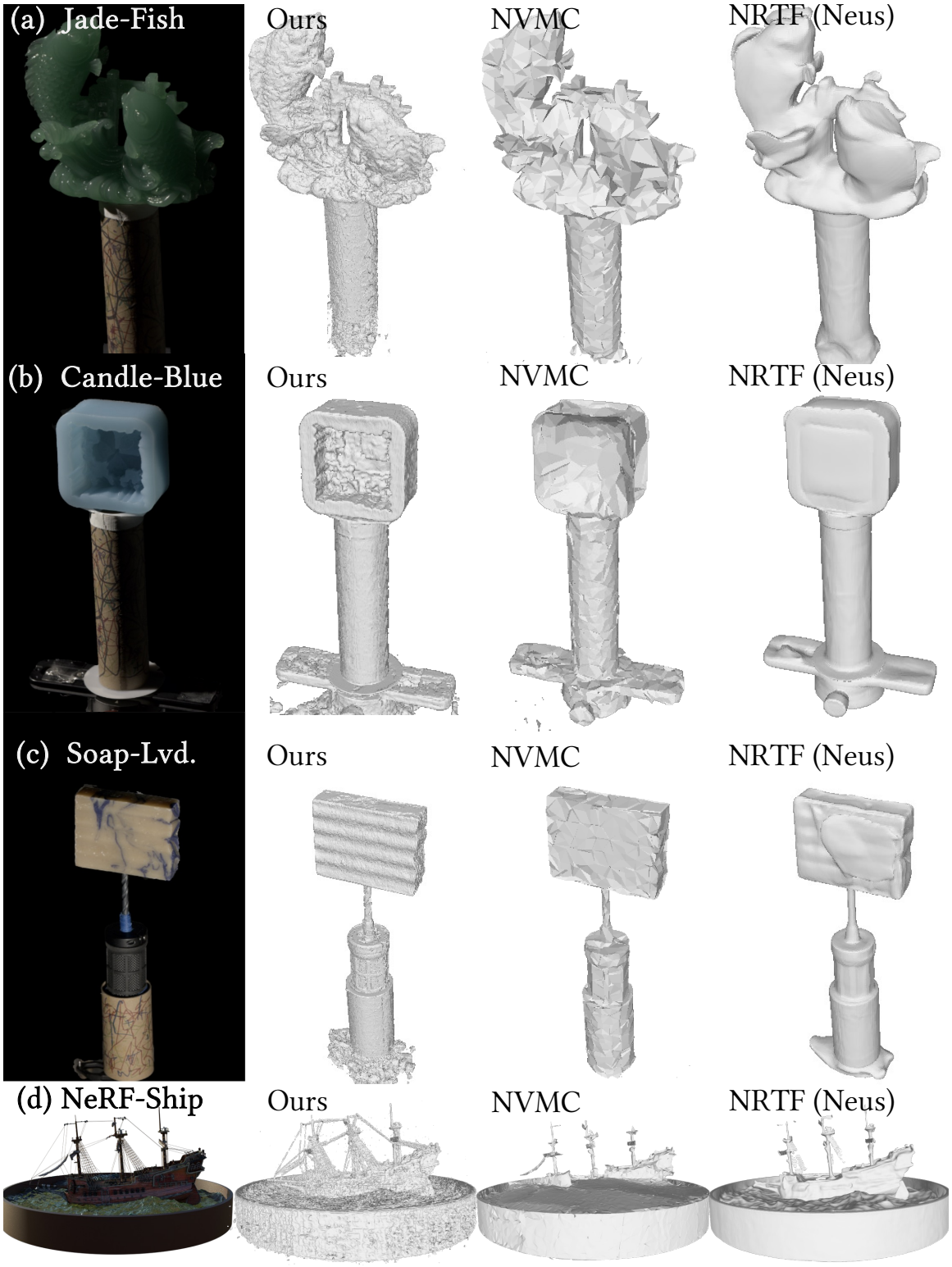


Figure 4.2: Despite presented with significant subsurface scattering and translucency in the scene, our approach provides the highest geometric reconstruction quality compared to other approaches (NVMC [5]; NRTF [6] via Neus [7]). For our approach, we show the extracted mesh using marching cubes from the density in the 512^3 resolution. The high quality geometry is one of the key advantages of our method.

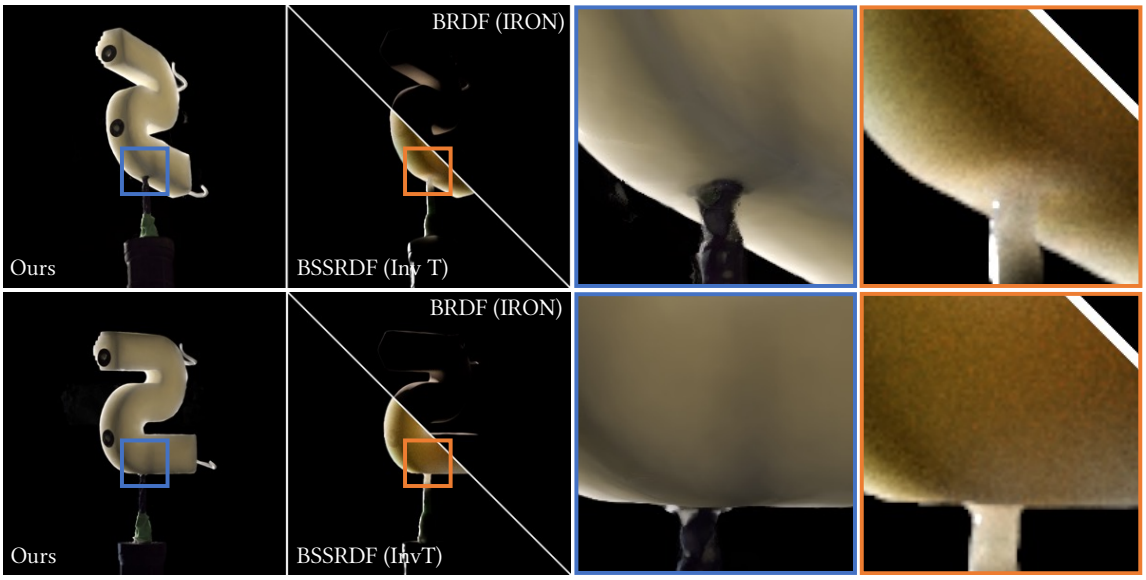


Figure 4.3: For relighting objects with subsurface scattering effects (e.g., the translucent soap shown in this figure), the BRDF-based approach [8] renders the object with full opacity when the light comes from the opposite directions, while the BSSRDF-based approach [9] cannot capture the texture details and structures beneath the surface (highlighted in the orange squares). In contrast, our approach can faithfully render the right opacity of the object and retain appearance even given the subsurface structure of the drill inside the candle (highlighted in the blue squares).

resentations, a higher dimension of inputs (usually 6D for homogeneous materials) is fed to query the outgoing radiance. A relighting algorithm can thus seek to optimize the BSSRDF function with the inverse rendering process so that the resulting material can be relit in conventional rendering engines. Our work follows a different path - we learn our relighting model in a fully data driven fashion, and learn the cached outgoing radiance for each point using a deep neural network, where we bypass the expensive BSSRDF computation in our optimization iteration.

Neural Radiance Fields and Precomputed Radiance Transfer. Neural Radiance Fields (NeRF) [15, 114, 115] optimize a parameterized volume rendering model from multiple views of the scene so that at test time, novel views can be synthesized from the learned model. Despite its superior rendering quality, NeRF bakes all the lighting and reflective surface information into the RGBs without modeling the interaction of the light and the material. Recent studies [6] have shown promising results for relightable models via incorporating the idea of "precomputed radiance transfer" (PRT) [116] from the real time rendering community. Instead of precomputing and caching the intermediate representation per location, they seek to optimize a cached intermediate representation in the reconstruction process. Notably, [6] relies on a fairly accurate pre-computed surface [7], and keeps the lighting appearance optimization separate from the geometry acquisition process. Focused on synthetic images with varying but known illumination, a NeRF extension [117]

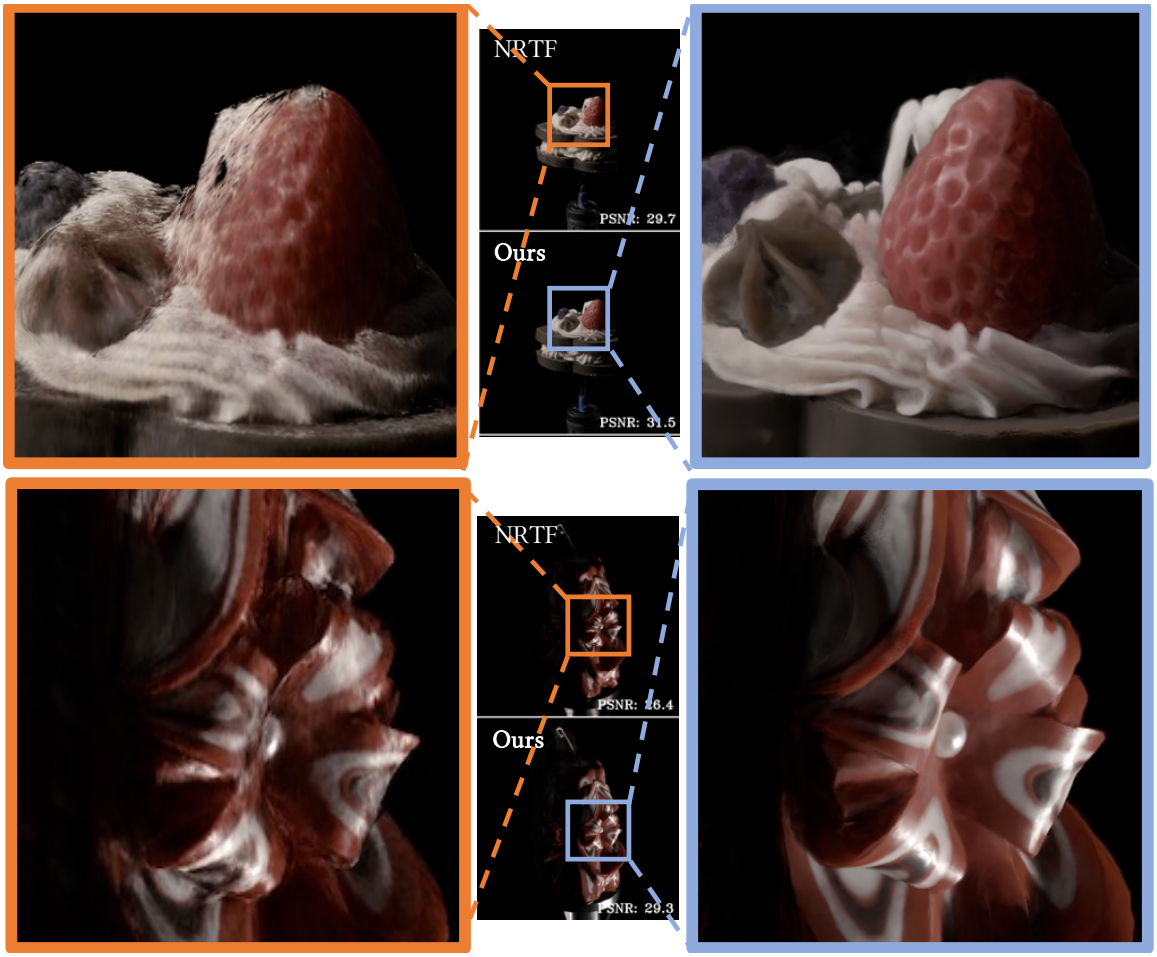


Figure 4.4: Volume rendering leads to cleaner surface reconstructions and higher rendering quality compared to NRTF [6].

was presented to reconstruct participating media with full global illumination effects, achieving good results on synthetic data. In contrast, our novel volume rendering framework not only enables optimizing the geometry details with appearance cues, but also works on scenes with partially opaque mass (e.g. thin rope or furs) and demonstrates high quality results on synthetic and real data. It is worth mentioning that a recent concurrent work [118] also addressed relighting with translucent objects using scattering functions. In addition to distant point lights, our approach efficiently relights the captured scenes with environment maps with the help of the Median-cut algorithm. Further, we will release high-resolution and large scale light stage dataset with rich lighting effects, such as translucency coupled with specular highlights and translucent shadowing, facilitating future research.

4.3 Methodology

4.3.1 Notation

Our goal is to optimize a relightable neural model from a collection of photos of the object, captured from different camera view points and under varying lighting conditions, that is able to accurately represent strong subsurface scattering effects. Our input includes the set of the input images $\mathcal{I} = \{\mathbf{I}_{c,l}\}$, where $\mathbf{I}_{c,l} \in (\mathbb{R}^+)^{M\times N\times 3}$ are high dynamic range (HDR) images, and c and l represent the camera viewpoint and lighting condition, respectively. We assume the camera poses are known (e.g., computed using photogrammetry software such as Agisoft Metashape), and denote them as $\mathcal{C} = \{\mathbf{K}_c, \mathbf{R}_c, \mathbf{t}_c\}$ (camera intrinsic parameters, rotations and translations, respectively). We capture one-light-at-a-time (OLAT) images for training, and denote an OLAT lighting condition as $\mathcal{L} = \{\omega_l\}$, where $\omega_l \in \mathbb{R}^3$ is the ℓ_2 normalized vector representing the incident point light direction relative to the scene center. Since our data capture system uses white light, we parameterize the light using a single channel throughout this paper. During testing, we apply an environment map (*envmap*) $\mathbf{E}_l \in \mathbb{R}^{M_E\times N_E}$, where each pixel of the envmap can be considered as one light source.

We want our relightable model to render the scene under varying *unseen* viewpoints ($\{\mathbf{K}_{\text{query}}, \mathbf{R}_{\text{query}}, \mathbf{t}_{\text{query}}\}$) and lighting conditions (ω_{query} or $\mathbf{E}_{\text{query}}$). Our framework optimizes the geometry as well as the lighting- and viewpoint-varying appearance of the scene in an end-to-end fashion. More precisely, we use the function $f_\Theta(\cdot)$ to denote our model (parameterized by Θ), and denote our model prediction as $\hat{\mathbf{I}}(u, v; \omega \text{ or } \mathbf{E}) = f_\Theta(\mathbf{r}; \omega \text{ or } \mathbf{E}) \in (\mathbb{R}^+)^3$, where $\mathbf{r}$ represents a pixel ray in the space, and (u, v) represents its related pixel coordinates on the image plane under the given camera pose $\{\mathbf{K}, \mathbf{R}, \mathbf{t}\}$. We provide an overview of our approach in Sec. 4.3.2, and provide details of our volume rendering scheme as well as model details in Sec. 4.3.3 and Sec. 4.3.4.

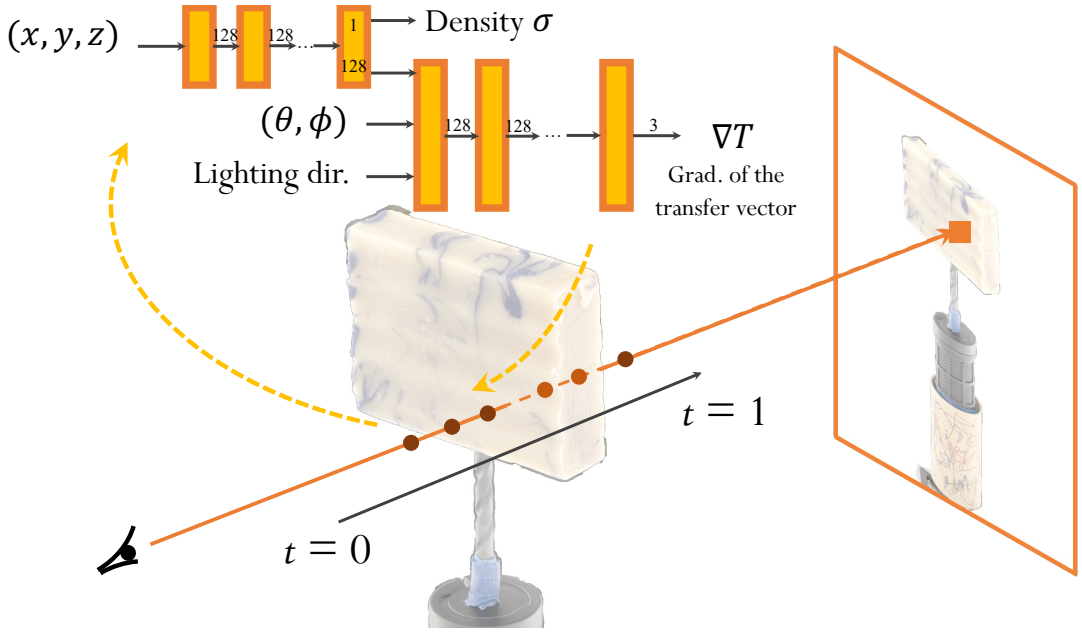


Figure 4.5: Illustration of the proposed relighting framework. We devise two MLPs to predict the gradient of the transfer vector for accumulating the HDR value of each ray. See Sec. 4.3 for details.

4.3.2 Method Overview

We devise a volume rendering based neural relightable model that is optimized directly from the image collections of varying camera views and lighting conditions (Fig. 4.5). The core of our learning framework consists of a volume renderer enabling an end-to-end optimization (Sec. 4.3.3) and the density-based neural transfer field networks (Sec. 4.3.4). There are several key differences compared to the existing (neural) relighting approaches. On one hand, unlike [6], our model can be trained from scratch, with no dependency on known estimated surface or other explicit geometry cues whose geometric details are difficult to obtain especially for materials with strong subsurface scattering effects (Fig. 4.2). Furthermore, training images captured under varying lighting conditions contain rich geometric cues via local micro shadowing or micro reflections, where a direct geometric optimization via an appearance loss is deemed necessary. On the other hand, thanks to our fully data-driven learning scheme, our model does not make any explicit assumptions about material (such as specifying a varying BRDF or BSSRDF) [8, 9], making it applicable to a wide range of materials, enabling global illuminations and subsurface scattering effects.

4.3.3 Volume Integration of the Transfer Gradient

The color of each pixel ray is computed using volume rendering. We denote the points sampled along the ray $\mathbf{r}$ as $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, where N is the total number of points. The model predicts $\hat{\sigma}(\mathbf{x}_i) \in \mathbb{R}^+$ and $\hat{\mathbf{h}}(\mathbf{x}_i; \mathbf{r}; \omega) \in (\mathbb{R}^+)^3$ for every sample

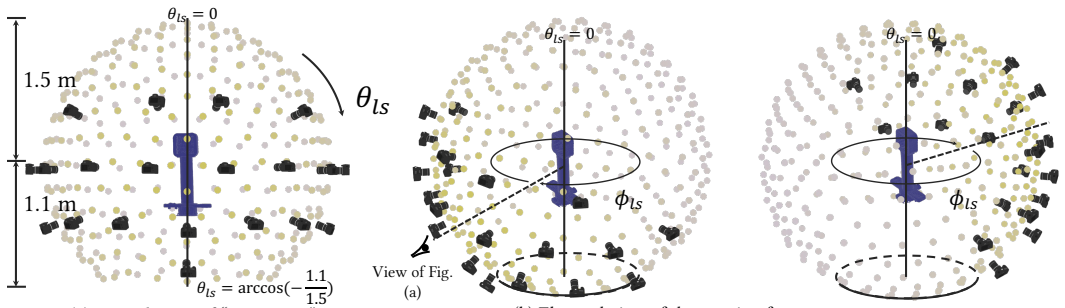
point, representing the density and the gradient of the pre-computed transfer vector, respectively. It is worth pointing out that instead of predicting the transfer vector directly as in [6], we predict the transfer vector gradient prediction, which represents the HDR contribution of a particular segment along a particular light transmission direction. It is clear that no HDR delta would incur at a density-free location, and among the non-zero density locations, the HDR contribution at a segment can only be non-negative if a location is visible, i.e. when its volume accumulation weight ($\hat{w}(\mathbf{x})$ in Eq. 4.1) is positive. This intuition aligns well with our volume accumulation and rendering scheme. We follow the volume integration from [15] and obtain the accumulated transfer vector prediction as:

$$\begin{aligned} \hat{\mathbf{I}}(u, v; \omega) &= \sum_{i=1}^{N} \hat{w}(\mathbf{x}_i)\hat{\mathbf{h}}(\mathbf{x}_i; \mathbf{r}; \omega) \\ \text{where} \qquad \hat{w}(\mathbf{x}_i) &= \hat{T}_i(1 - \exp(-\hat{\sigma}(\mathbf{x}_i)\delta_i)) \\ \hat{T}_i &= \exp(-\sum_{j=1}^{i-1} \hat{\sigma}(\mathbf{x}_j)\delta_j) \\ \delta_i &= t_{i+1} - t_i. \end{aligned} \tag{4.1}$$

Our volume rendering scheme demonstrates several key benefits over a surface representation [6] (Fig. 4.4). First and foremost, obtaining a fairly accurate pre-estimated surface for materials featuring subsurface scattering with detailed geometry is non-trivial. Our model bypasses the difficulties of pre-estimating the surface geometry by applying volume rendering and optimizing the surface density together with appearance. In this case, all local shadowing and reflection effects captured under different lighting conditions are taken into account for geometry estimation, providing stronger cues compared to surface estimation under a single lighting condition. Second, volume rendering enables accurate appearance modeling of semi-opaque materials (e.g. fur) with their subsurface scattering effects, which cannot be trivially achieved using a surface-based rendering framework. Third, similar to other volume rendering-based models, our model is able to optimize the geometry as well as the relightable appearance end-to-end under varying lighting conditions. We do not require model design changes to back propagate the loss gradient back to the geometry prediction [93]. Our results show that this rendering strongly result in higher fidelity compared to previous surface-based rendering [6].

4.3.4 End-to-end Learning of Neural Relighting

Architectures. We follow [15] and use an MLP to predict the density as well as the transfer vector gradient for each sample interval. The MLP consists of 8 fully-connected layers (with a width of 256, and a skip connection in the fourth layer) each for the density as well as the transfer vector gradient prediction respectively. We



(a) Frontal View of "Group 000" (b) Elevated view of the rotation from group 000 to group 240.

Figure 4.6: Illustration of our light stage capture system. A full capture consists of 9 capture groups, with each group labelled as "000", "040", "080", ..., "320", with their number denoting the 40 degree-stepped yaw rotation (see "ϕ_{ls}" in (b)). Lights are visualized as dots and cameras with camera icons. All lights are of the same white color—the visualized dot colors merely refer to the light bulb instance, highlighting that the lights are locked with the cameras when rotating between groups. **(a)** Frontal view of the system (group "000"). The radius of the light stage is 1.5 meters, with its center at 1.1m height—the layout is a bottom-truncated sphere. The light stage illustration in Fig. 3.1(a) is the elevated view of group "040". **(b)** Rotating from group "000" (b-left) to group "240" (b-right) according to the "ϕ_{ls}" rotation. (b-left) and (b-right) are visualized at an elevated angle. (a) is viewed from the dashed line direction in (b-left).

devise two MLPs tackling the coarse level and fine level of accumulation respectively. To ensure the predicted transfer vector gradient to be non-negative, we use the exponential function as the activation function, following [119]. It is worth pointing out that MLPs are just one option for modeling the predictions of each point, we expect that more efficient models [31, 120, 121, 122] can be used as well.

Loss functions. We utilize the weighted L2 tonemapped loss [119] to supervise the predicted HDR values of each pixel. We also impose an auxiliary mask loss, where we pose L2 regularizers on the density of all points that are sampled on a background ray. Our main focus is the modeling of the foreground objects, and due to the inconsistency of the background appearance (rotation nature of the camera groups, see Sec. 4.4), we set all the ground truth HDR values in the background to be 0. We sample the rays (from both the foreground as well as the background) with importance sampling, where in each training batch, 1/2 of all rays are from the foreground, 3/8 of all rays are from the near-silhouette area, and 1/8 of total rays are from arbitrary locations in the background. In our real light stage data, since the aspect ratio of the captured images is pretty large, we extend the background area that is outside of the pixel map. We pad them with 0 as their ground truth HDR values. We found this to be useful for clean free-space density predictions.

Using environment map conditioning. We follow [6] to obtain the envmap relighting prediction via accumulating the OLAT HDR prediction. More precisely, we treat each pixel on the envmap as an OLAT point light. Practically, we apply the median cut algorithm [123] to accelerate inference. During accumulation, we reweight the predicted HDR value from each OLAT location by the cosine value of the latitude

angle on the envmap to account for the area of lights on the envmap sphere. The aggregated predicted rendering serves as our final prediction of the relighting given the query envmap.

4.4 Light Stage Data Acquisition

To facilitate studies on the light-dependent appearance modeling of objects and scenes under significant subsurface scattering effects, it is critical to acquire real-world objects featuring such effects. While existing datasets (e.g., [9]) includes captures of two translucent objects, they are often limited by resolution and fidelity of the acquired images, causing local micro geometry details to not be fully captured. To reconstruct a relightable model in a data-driven fashion, we aim to have real-world captures with densely sampled camera viewpoints, complete incident light direction coverage and high-resolution images retaining as much detail as possible. Consequently, we propose a new dataset, consisting of 8 scenes with significant subsurface scattering effects. Our captured data demonstrates high fidelity, preserving rich appearance details, and represents a total of 15TB (3000 times larger than the currently highest quality dataset with similar goals to our knowledge, [9]).

As shown in Fig. 4.6, we place the cameras and the light sources on the spherical light stage cage, while the objects to be captured are placed on a holding table in the center with a height of roughly 1.1 meter. In particular, when capturing the data, our cameras and the light bulbs are fixed on the sphere, while a turntable in the middle can be freely rotated. Ignoring background pixels, this is equivalent to keeping the object scene static to satisfy the consistency of the scene among views, while rotating the cameras and the light bulbs altogether. Throughout the text, we assume that the light stage is configured in the latter case for notational convenience. Our camera/light-bulb sphere radius is roughly 1.5 meter from the surface of the holding table in the middle).[1] The rotations of the sphere put the whole captured frames into 9 groups, with each group corresponding to one particular rotated setting of the camera-light sphere. On the sphere, we have a total of 20 cameras as well as 331 lighting bulbs (serving as 331 OLAT point lights).[2] Consequently, in each group we captured a $20 \times 331 = 6620$ frames, and for the total 9 groups, we captured a total of 59580 frames for one scene. Our camera captures high dynamic range value for the RGBs, with the cutoff threshold at 4.4019. The original captured frames come with a resolution of 8192×5464. We found a 4 times down-scaling retains most of the texture details and hence we conduct all our experiments on the down-scaled version

[1] Our light bulbs only span roughly between $[0, \frac{3}{2}\pi)$ for θ_{ls}, hence no light bulb has a negative altitude even if the sphere radius is larger than the height of the center—the holding table.

[2] Notably, since the point light locations are locked with the camera during rotation, the OLAT location in different groups are different from each other. In other words, in our whole dataset, there are only up to 20 images that have been recorded with the same lighting.

(2048×1366). Notably, all the captures at the resolution of 2048×1366 still span 15TB of storage. During the capture, the cameras always face toward the objects on the holding table, and we tune the focal length of the camera to best suit the size of the particular objects. We obtain the extrinsic camera poses via an off-the-shelf software with manual corrections. Since the light bulbs shining in the opposite direction of the camera incur significant noise to the reconstruction process (especially considering that the rotation between the group would make the background inconsistent), we introduced several heuristics, including RGB variations and saturation to segment out the background. All the camera poses, light locations as well as the masking information are used by all the approaches in our evaluation sections (Sec. 4.5), and we shall make all the details about the data publicly available to facilitate future research.

4.5 Experiments

We use both the synthetic data (8 scenes) and the real data we captured (8 scenes) for evaluation and comparisons. All the details on data, training and benchmarking protocols will be released.

Synthetic Data. We use the 8 scenes from the NeRF Blender dataset [15] and evaluate them with both their original materials (*Synthetic-Original*) as well as their modified materials with the subsurface scattering shader in Blender [124] (*Synthetic-SSS*). During training, we use the same 100 camera views given in the training set for each scene as provided by the originally released data [15]. To simulate OLAT lighting, we evenly sample 112 incident lighting directions on the upper hemisphere. More precisely, we sample evenly with 7 latitudes in the upper hemisphere, evenly sampled 32 longitudes for each latitude, and left out every other light (to be used during evaluation). The 7×32 OLAT directions exactly correspond to Row 2 through Row 8 of the 16×32 envmap as used in NRTF [6]. We exclude the lower hemisphere for OLAT sampling, mainly due to the fact that most of the scenes in the NeRF blender dataset are rendered as top views, and the OLAT lighting from the bottom produces overall dark renderings. This training setting gives us a total of 11200 training images per scene. To mimic the light stage setting used for real-world data capture, we use only white lights, and use the point light instead of the envmap for rendering the ground truth. More precisely, the point lights are placed roughly 100 units away from the scene center (with about 4 units being the approximate size of each scene). During testing, we use unseen lighting directions as well as unseen camera poses for each test sample. For quantitative evaluation, we stick to the OLAT protocol where there is only one light at a time. For saving evaluation time, we only test 10 out of the unseen 112 lights. We also provide qualitative samples by rendering results with several envmaps downloaded from PolyHaven (e.g., Fig. 4.10). Since our point lights are single-colored (white), we do the inference with the independent-

	Real-SSS	C-Red	J-Dragon	J-Fish	C-Cake	S-Lvd.	C-Blue	C-Head	C-S	Average
PSNR(↑)	IRON [8]	21.6	17.5	20.7	22.2	22.4	19.1	21.4	23.3	21.0
	InverseTranslucent [9]	23.3	21.6	23.6	22.9	25.1	21.8	25.0	26.8	23.8
	NRTF [6]	27.5	28.5	28.4	29.7	30.7	29.0	30.7	32.0	29.6
	Ours	**30.9**	**29.0**	**30.3**	**32.3**	**33.2**	**31.2**	**34.1**	**36.3**	**32.2**
SSIM(↑)	IRON [8]	85.5	85.7	82.8	88.6	89.2	82.7	90.0	90.8	86.9
	InverseTranslucent [9]	86.2	89.6	84.6	89.7	90.7	86.3	92.3	93.3	89.1
	NRTF [6]	92.3	94.0	92.5	94.1	94.7	92.8	95.8	96.5	94.1
	Ours	**93.4**	**94.7**	**93.3**	**94.8**	**95.7**	**93.8**	**96.9**	**97.6**	**95.0**
LPIPS(↓)	IRON [8]	0.131	0.143	0.173	0.108	0.109	0.179	0.109	0.106	0.132
	InverseTranslucent [9]	0.139	0.132	0.165	0.119	0.110	0.186	0.104	0.104	0.132
	NRTF [6]	0.110	0.095	0.125	0.088	0.088	0.139	0.082	0.080	0.101
	Ours	**0.099**	**0.089**	**0.123**	**0.078**	**0.077**	**0.132**	**0.071**	**0.067**	**0.092**

Table 4.1: Comparison with several state-of-the-art methods on the "Real-SSS" data (8 scenes). Despite optimized on the same data, our results consistently outperform the existing approaches on all scenes and all evaluation metrics. Material abbreviations: "C-" stands for "Candle", "J-" stands for "Jade", and "S-" stands for "Soap".

RGB-channel assumption when relighting under a colored envmap. Following [6] we cast them into a 32×16 envmap to serve as the input. For test time camera poses, we apply the camera views from the test views given in the NeRF blender dataset. For saving evaluation time, we only test 10 out of the unseen 200 test views. This test setup gives us 100 test cases in total for each scene.

Light Stage Data. As introduced in Sec. 4.4, the proposed light stage data contains 9 groups and 20 cameras per scene (a total of 180 views), with each view consisting of 331 OLAT renderings, thus leading to a total of 59580 HDR images per scene. During training, we use the first 18 cameras in each group, and use 75 out of the 331 OLATs for training, leading to a total of 12150 training images per scene. Testing on real data also only includes samples with both, unseen lighting directions and unseen views. For quantitative evaluation, we use the remaining 2 cameras from each group (a total of 18 views) and 10 unseen OLATs to form our test set (180 images per scene). For qualitative evaluation, we use the same input lighting envmaps as used in the synthetic data benchmark. Since most of our real captures exhibit subsurface scattering, we denote this data with *Real-SSS*.

Evaluation Metrics. We evaluate the predicted pixel map following the standard metric protocol [6, 15], including PSNR, SSIM and LPIPS [125]. While our main focus is to evaluate the objects of the scene, we follow existing protocols [15] to include all the pixels for evaluation. Following most of the recent evaluation conventions (e.g. [15]), we evaluate every pixel on the predicted pixel maps (including the background regions). This also includes the areas where the stand holds the captured objects.

Baseline approaches. We compare with several most representative state-of-the-art approaches to highlight the strengths of our neural relightable model. All models are trained with exactly the same data.

• **IRON** [8] is a recent representative BRDF-based relighting approach and achieves

Nerf-Blender (Original)		Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Average
PSNR(↑)	IRON [8]	26.4	26.2	27.2	22.9	22.7	22.3	25.9	21.7	24.4
	InverseTranslucent [9]	24.1	24.8	27.8	24.4	21.8	23.2	25.2	19.0	23.8
	NRTF [6]	28.7	33.1	32.3	29.6	23.9	31.2	29.3	23.7	29.0
	Ours	**32.6**	**35.4**	**35.0**	**35.3**	**31.3**	**35.8**	**35.7**	**25.4**	**33.3**
SSIM(↑)	IRON [8]	93.0	90.3	89.0	91.2	86.1	83.5	94.0	78.0	88.1
	InverseTranslucent [9]	87.4	83.5	88.3	86.3	81.0	82.7	90.1	68.6	83.5
	NRTF [6]	93.4	**95.7**	93.6	94.1	87.4	93.9	96.1	**85.8**	92.5
	Ours	**96.2**	94.1	**95.3**	**96.6**	**94.3**	**96.2**	**98.3**	84.8	**94.5**
LPIPS(↓)	IRON [8]	0.081	0.144	0.126	0.103	0.156	0.182	0.077	0.204	0.134
	InverseTranslucent [9]	0.128	0.194	0.125	0.172	0.204	0.175	0.104	0.278	0.173
	NRTF [6]	0.077	**0.091**	0.079	0.077	0.143	0.097	0.058	**0.150**	0.097
	Ours	**0.065**	0.135	**0.061**	**0.062**	**0.084**	**0.072**	**0.036**	0.196	**0.089**

Table 4.2: Detailed comparison with the state-of-the-art baselines on the "Synthetic-Original" data (8 scenes). It is worth pointing out that InverseTranslucent [9] was not proposed to handle the type of data used in this benchmark.

Nerf-Blender (SSS)		Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Average
PSNR(↑)	IRON [8]	24.0	22.0	25.8	22.2	24.0	19.3	28.0	19.5	23.1
	InverseTranslucent [9]	28.6	23.1	27.5	30.3	24.9	30.0	30.5	20.2	26.9
	NRTF [6]	32.2	30.5	32.0	32.8	25.6	33.2	33.4	29.2	31.1
	Ours	**40.3**	**36.2**	**37.1**	**45.3**	**34.4**	**44.0**	**42.5**	**34.3**	**39.3**
SSIM(↑)	IRON [8]	90.5	81.5	87.8	90.2	84.1	74.7	91.8	77.7	84.8
	InverseTranslucent [9]	90.9	83.7	86.6	93.1	81.3	91.8	94.1	75.8	87.2
	NRTF [6]	93.8	93.9	93.8	95.4	87.4	93.9	96.0	88.6	92.9
	Ours	**98.5**	**97.4**	**97.5**	**99.1**	**96.4**	**98.6**	**99.1**	**92.9**	**97.4**
LPIPS(↓)	IRON [8]	0.102	0.191	0.129	0.125	0.155	0.239	0.091	0.305	0.167
	InverseTranslucent [9]	0.120	0.197	0.149	0.149	0.204	0.097	0.078	0.276	0.159
	NRTF [6]	0.078	0.095	0.075	0.087	0.150	0.108	0.061	0.189	0.105
	Ours	**0.027**	**0.051**	**0.031**	**0.024**	**0.048**	**0.077**	**0.015**	**0.165**	**0.055**

Table 4.3: Detailed comparison with the state-of-the-art baselines on the "Synthetic-SSS" data (8 scenes). It is worth pointing out that IRON [8] was not proposed to handle the type of data used in this benchmark.

state-of-the-art performance with the collocated GGX shader. We underwent major efforts to generalize it to the general setting where the incident light direction, viewing direction and bi-sector direction are no longer identical. Notably, while the GGX shader cannot handle subsurface scattering, optimization in scenarios where lighting is coming from the opposite side of the camera is essential, especially when translucency is present. We used Mitsuba 3 [126] to render the trained textured models and fit the best HDR scaling with the ground truth before computing PSNR.

• **InverseTranslucent** [9] is a recent representative state-of-the-art BSSRDF relighting approach. We train the models using spatially varying albedo, sigma (controlling light transmission underneath the surface) and roughness all in the resolution of 256 $\times$ 256. We found [9] is sensitive to the geometry initialization, and thus we provide the baseline with the optimized Neus reconstruction using their original implementation [7].

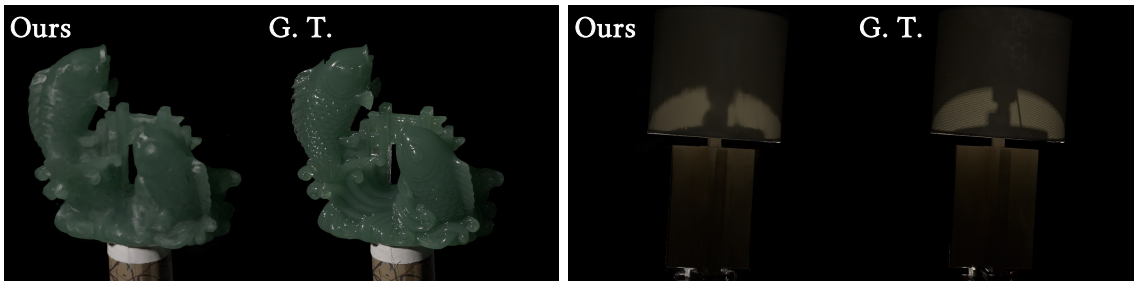


Figure 4.7: Failure cases on specular highlights (left) and translucent shadowing (right). The proposed method does not explicitly model specularities and shadowing.

• **NRTF** [6] is a recent state-of-the-art fully data-driven approach that is designed to handle global illumination and potentially subsurface scattering.

For [8, 6], we use the provided Neus implementation rather than the original version to obtain object surfaces.

It is worth pointing out that [8] was originally proposed to handle only the PBR based materials with the assumptions that all the objects are fully opaque, and hence it was not proposed to handle our evaluation data of *Synthetic-SSS* and *Real-SSS* (our proposed light stage data). Meanwhile, [9] was originally proposed to handle specifically objects with translucency, but not necessarily opaque objects as present in our evaluation data *Synthetic-Original*. We still include all results in the experiments for reference purposes since our approach is able to handle all the types of the materials, further showcasing the wide applicability of the method.

Results. As shown in Tab. 4.1-4.3 and Fig. 4.8-4.10, our results demonstrate clear advantages compared to all aforementioned methods. Notably, we achieve 5 points overall average PSNR gain (averaging over all the synthetic and real data) over the best-performing existing method thanks to our end-to-end learning framework. We conclude that our relighting approach can not only handle a wider range of material types (in particular objects with subsurface scattering effects) with significantly improved fidelity, but also stays flexible representing vivid and rich geometric structures, such as the thin ropes that are generally not easy to represent using meshes. In contrast to other approaches [8, 9] that were designed to handle a relatively narrow range of material types, our approach is able to handle the full variety of materials present in the datasets. This underscores the general applicability of our approach regarding material representations. Please refer to our supplementary materials for additional results.

Limitations. Our approach exhibits two main types of failure modes. First, the proposed method may return blurry results for specular highlights (c.f., Fig. 4.7—left) since the model does not take specularities into account in a dedicated way. Similarly, our approach does not contain a dedicated model for shadows. In particular, when shadows "penatrate" a thin layer of translucent material (e.g., Fig. 4.7-right) our model creates blurry boundaries on otherwise hard shadow borders.

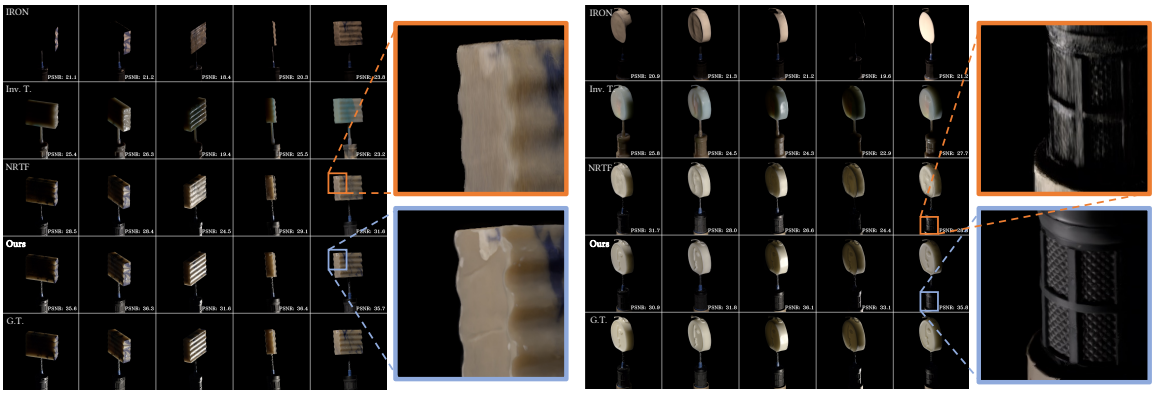


Figure 4.8: Detailed comparison for Soap-Lavender (**left**) and Candle-Head (**right**) between our results (Row 4) with other state-of-the-art approaches (IRON [8] in Row 1, InverseTranslucent [9] in Row 2, and NRTF [6] in Row 3). Recordings can be found in the last row; all images are held out positions for lights and cameras. Our results show a clear advantage in terms of visual fidelity and geometric accuracy.

Another avenue for future improvement is rendering speed: the proposed model does not yet meet the demand of real-time applications. Further, our relighting algorithm is relying on a light stage capture system and is not yet suited for in-the-wild use.

4.6 Translucency-Reflection Modeling

Image-based lighting encodes the illumination at a point in space from all directions in an image, the environment map, and uses this information to realistically light an object. It is a widely used technique in industry due to its flexibility: natural illumination from the physical world can swiftly be recorded using a 360 degree camera to create a light probe[3] and then used to place virtual objects in real scenes seamlessly. Depending on its resolution, such a light probe is sufficient to describe the entire far field environmental light, and can be used to render all kinds of materials. Rough and glossy surfaces can be reproduced correctly and may contain a high resolution reflection of the environment, for example; for translucent objects, the entire light transport can be computed from all directions. However, that comes at a cost: if the light probe is used directly, even for a low resolution of 16×32, 512 passes would have to be performed if no simplifying assumptions are being made. Even for highly optimized rendering frameworks this makes rendering a single frame too slow for real-time applications, unless simplifying assumptions about the materials and lighting are made (e.g. pre-convolving the environment for high-roughness materials,

[3] `https://www.pauldebevec.com/Probes/`

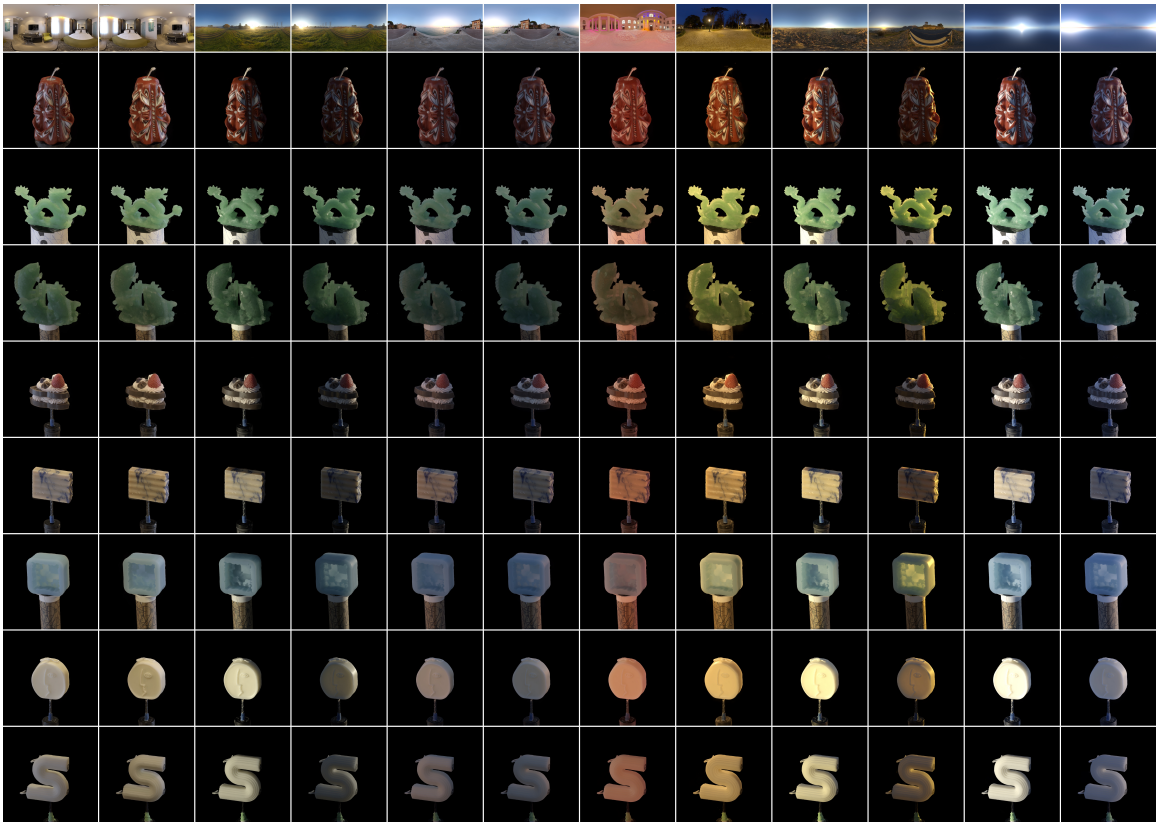


Figure 4.9: Envmap relighting results on our real-SSS dataset (light stage captures). The results in each row are from the same scene, while the results in each column are relit using the same environment map.

Figure 4.10: Relighting results for various environment maps for the original as well as the translucent version of the synthetic scenes from the Nerf-Blender synthetic datasets (*Synthetic-Original* and *Synthetic-SSS*).

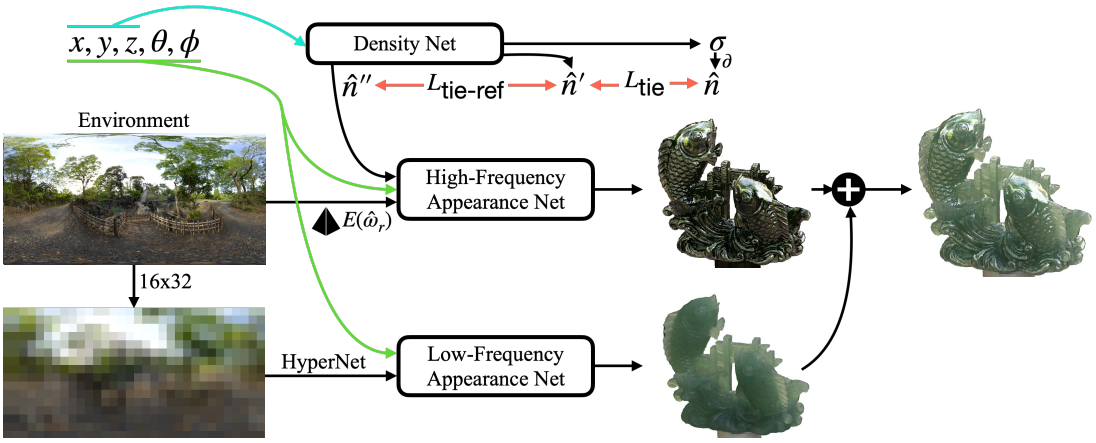


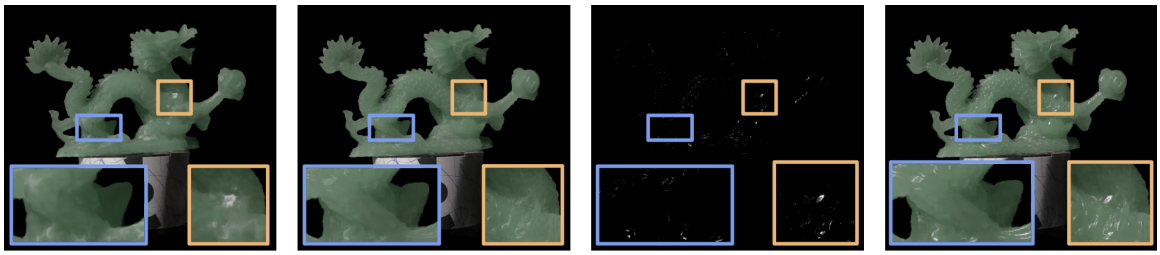
Figure 4.11: Overview of the proposed method. The inputs for our volumetric model are sampling position x, y, z, view direction θ, ϕ and an environment map (top left). A density model predicts material density σ. The appearance is predicted in two components: high- and low-frequency. For the high frequency component, we compute a reflection hint pyramid (see Sec. 4.6.4). This enables the model to make detailed prediction about specularity while taking ambient occlusion into account. For the low frequency prediction, we use a downsampled version of the environment map in combination with a HyperNet. Overall, only a single rendering pass is necessary for full, image-based illumination. Careful optimization of the normals is necessary during reconstruction: we use three normal estimates in the process and two tie losses; for details, see Sec. 4.6.4.

often used in real time rendering). Yet, combining IBL with radiance field reconstruction techniques and pre-computed radiance transfer (PRT) is highly attractive. The combination of these techniques covers the radiance transfer through an object in its entirety without any simplifying assumptions and allows rendering of even the most complex materials faithfully.

In this part, we propose a method to achieve this at 5 frames per second for 800×800 resolution images. It accounts for hard shadows and specular highlights on glossy surfaces—a particularly hard case for high resolution environment maps. Our model avoids repeatedly querying the environment for accumulation and instead uses a neural model to 'summarize' them instead. This works well for low frequency illumination and has been explored in the past with HyperNets. However, this approach does not fare well for high frequency details: for this scenario, we propose a separate model stream that uses a reflection hint pyramid that can be precomputed and is queried only once given the computed normal direction. This means, our model achieves IBL for low- and high-frequency lighting including reflections in a single pass. We name the full model model TRHM (Translucent-Reflection Hybrid Modeling).

4.6.1 Overview

Our approach relights a scene based on the given query lighting input - an environment map that is denoted as $\mathbf{E} \in \mathbb{R}^{M_E \times N_E \times 3}$, where M_E and N_E denotes the



(a) Without specular highlight modeling. (b) Low-frequency prediction. (c) High-frequency prediction. (d) Full result.

Figure 4.12: Separation into low-frequency and high-frequency enables the proposed framework to render complex materials with high fidelity. In this example, the jade structure is present in the low frequency rendering, but does not exhibit specular highlights. These are captured well in the high frequency component, leading to a faithful rendering. Without the separation, reconstruction of this material is not possible.

height and width of $\mathbf{E}$. During training, our model is optimized from a collection of high dynamic range images $\{\mathbf{I}|\mathbf{I} \in \mathbb{R}^{H\times W\times 3}\}$ (H for the height and W for the width), with each image associated with the camera poses as well as the lighting information (*e.g.*, $\mathbf{E}$). We denote each pixel ray as $\mathbf{r} \in \mathbb{R}^3$, with its radiance from the image capture as $L(\mathbf{r}) = I(x, y)$, where (x, y) are the pixel coordinates of the ray $\mathbf{r}$ on the image plane. We thus denote our prediction problem as $\hat{L}(\mathbf{r}; \mathbf{E})$. Part of our training data also involves point lights as the lighting inputs. We denote the incident direction of the point light as $\omega_l \in \mathbb{R}^3$, and the prediction can thus be conditioned on the point light as $\hat{L}(\mathbf{r}; \omega_l)$. Throughout the methodology section, all the predicted variables are denoted with " ˆ ".

Assumptions. We assume the given image-based lighting is not a near-field lighting as a result of using environment maps as inputs. We assume all the subsurface scattering effects are isotropic. The hard shadow effect is not explicitly modeled in our approach and is beyond the scope of this work.

Modeling. We propose our fast neural relighting algorithm for image based lighting that faithfully captures lighting effects, such as translucency, subsurface scattering as well as hard shadows, specular highlights and glossy reflections. We term our model as translucent-reflection hybrid modeling (TRHM), that stands for two representative lighting effects. With an environment map as the input, we do not do expensive enumeration of every pixel as a point light to aggregate the rendering results [6]. Instead, we propose a set of techniques for enabling image-based lighting without querying the same lighting network numerous times, including *i)* a hypernet-based [127, 128, 129] distillation of the neural radiance transfer field for low-frequency effects such as diffuse component and subsurface scattering (Sec. 4.6.2); *ii)* learning of the reflection effects among various surface roughnesses with an incident pyramid hint (Sec. 4.6.3); and *iii)* learning of the local micro-geometry with the help of the specular highlight cues

(Sec. 4.6.4). We also showcase the flexibility of our model for moving from point-light based data training (*e.g.*, a light stage) into image-based lighting at render time 4.6.5. Figure 4.11 illustrates the proposed TRHM framework.

4.6.2 Hypernet-based Radiance Transfer Fields

Background - the Point Light Scenario. By incorporating the radiance transfer modeling into the neural relighting framework [6], one can enable global illumination and subsurface scattering with orthogonal basis lighting representations. For point lights, adding the additional dimensions that encode the incident lighting direction ω_l [6, 118, 130] would achieve global illumination effects, *i.e.*, for each query point $\mathbf{x} \in \mathbb{R}^3$ and the associated viewing direction $\mathbf{d} \in \mathbb{R}^3$, the color prediction becomes $c(\mathbf{x}; \mathbf{d}; \omega_l)$, with the additional input ω_l of the point light information.

HyperNet for Image-Based Lighting. To address the more challenging problem for relighting with image-based lighting, instead of directly concatenating the envmap $\mathbf{E}$ with $(\mathbf{x}, \mathbf{d})$, we propose to use a hyper network [127] to generate the parameters of the color branch of the NeRF network for better representational power, motivated by recent related approaches [128, 129]. Our hypernet-based prediction thus becomes

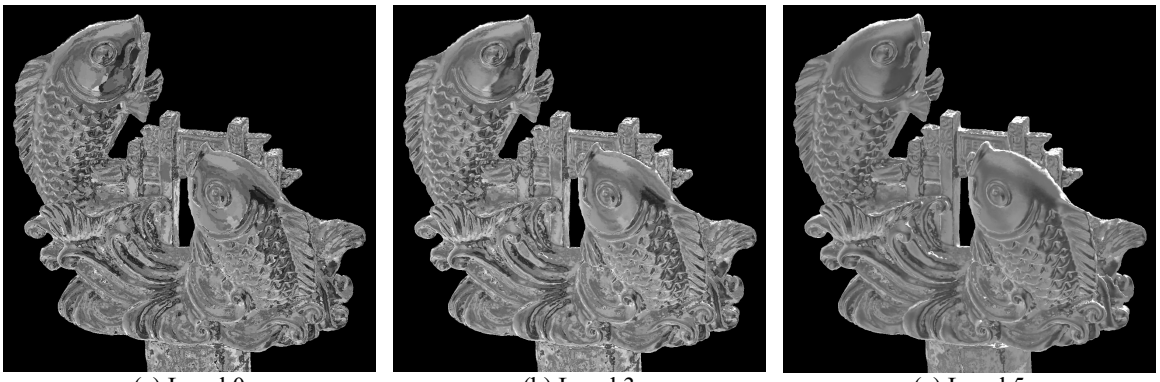
$$\begin{aligned} \Theta_{\text{color}} &= \mathcal{H}(\mathbf{E}), \\ \hat{L}_{\text{low-freq}} &= f_\Theta(\mathbf{x}; \mathbf{d}). \end{aligned} \tag{4.2}$$

Given that a vanilla radiance transfer model in the neural relighting framework generally lacks in high-frequency situations [10], we use a downsampled low resolution version (16×32) of $\mathbf{E}$ as input to the hyper network to improve rendering performance. We empirically found that a 3-layer MLP is sufficient for including the lighting information to the NeRF network.

The hyper network is expected to handle only the low-frequency part of the lighting effects (the diffuse component and subsurface scattering). We next proceed to introducing our algorithm for handling high frequency lighting effects such as specular reflections and highlights.

4.6.3 Reflection Hints for Image-Based Lighting

We noticed that reflections on glossy surfaces are one of the most prominent high-frequency effects that hypernets (Sec. 4.6.2) or the transfer field [6, 130, 118] fail to address. As illustrated in Fig. 4.12, the high response of the specular highlights on the glossy surface as well as their high correlation with the surface normal poses major difficulties for existing works [6, 130, 118]. Incorporating the neural hints [10] for predicting lighting appearance has demonstrated promising results for reflections. Under the point light condition, Zeng et al. [10] proposed to incorporate the reflection



(a) Level 0 (b) Level 3 (c) Level 5

Figure 4.13: Reflection hint pyramid visualization. The reflection hints model reflected irradiance and take self occlusion into account. As the pyramid level increases, roughness increases and the reflection becomes more blurry.

hints, a GGX micro-facet representation with four pre-defined roughnesses, for predicting specular highlights. When using an environment map, the lighting becomes a large collection of light sources, and it is hard to efficiently fit to the above-mentioned representation without the enumerate-and-accumulate inference process. We observe that the reflection of a point on the surface is highly dependent on the envmap intensities int the reflection direction as well as its adjacent directions, supposing no occluder is present on the reflection ray. With a lower roughness value, the reflection is impacted by a narrower range of envmap directions, while a higher roughness value means a wider range of envmap directions impacts reflection value. We propose to utilize a pyramid of envmaps, denoted as $\{\mathbf{E}_j\}_{j=0,1,2,...}$, with each level l processed by a Gaussian filter of a particular kernel size and standard deviation, to approximate the reflection hints for various surface roughnesses. We simply query the envmap HDR values $\mathbf{E}_j(\hat{\omega}_r)$ in the predicted reflection direction $\hat{\omega}_r$ [131] at each level j, and concatenate them to obtain our 'reflection hints' under the image-based lighting condition:

$$H_{\text{ref(envmap)}} = \{\mathbf{E}_j(\hat{\omega}_r)\}_{j=0,1,2,...}. \tag{4.3}$$

Fig. 4.13 illustrates the reflection hints based on the given environment map lighting, where we simply cast the object material as the mirror for visualization purpose. The hints $H_{\text{ref(envmap)}}$ retrieved from different levels of the environment map demonstrate different levels of blurriness of the environment map, serving for hints for different levels of roughness of the materials of the surface reflection. The hint serves a similar functionality as the point-light reflection hint [10], with the difference that the point-light reflection hint [10] directs the model to interpolate between the GGX reflection response under varying surface roughness, while our proposed image-based reflection hint instead lets the model to interpolate between the varying levels of the envmap HDR response. On one hand, our hint relaxes the constraint of using only

a single point-light, and can be seamlessly integrated into the image-based lighting prediction framework. On the other hand, the choice of this hint further enables the flexibility of transferring the learning from the point-light training data (e.g. a light stage) to the image-based lighting prediction scenario (Sec. 4.6.5), where different levels of material roughness shall be able to receive point light signal if the reflection direction $\hat{\omega}_r$ is close to, but no identical to, the point light direction ω_l.

Handling of self-occlusions. To accommodate self-occlusions along the reflection direction as well as approximating the prediction of the reflected light for more than one surface bounce, we further incorporate the self-occlusion cue as part of our reflection hint. More precisely, we further incorporate the opacity prediction as well as the incident radiance as part of our reflection hint. We predict these hints via putting a virtual eye on the predicted surface point, looking toward the predicted reflection direction of the ray ($\omega_r(\mathbf{r})$). We denote the obtained opacities as well as the radiance color along the reflection direction as $H_{\text{ref(opacities)}} \in \mathbb{R}$ and $H_{\text{ref(incident)}} \in \mathbb{R}^3$ respectively. Note that when predicting $H_{\text{ref(incident)}}$, it further requires to trace to the next reflection bounce, resulting in a recursive problem. We found that setting $H_{\text{ref(incident)}}$ to zero when computing the hints bypasses this issue while maintaining promising visual results. The full version of our proposed reflection hint could be written as

$$H_{\text{ref}} = \{H_{\text{ref(envmap)}}, H_{\text{ref(opacities)}}, H_{\text{ref(incident)}}\}. \tag{4.4}$$

Based on this hint, we devised a 3-layer MLP to predict the high-frequency prediction $\hat{L}_{\text{high-freq}}(\mathbf{r})$, and compute the full prediction via

$$\hat{L}(\mathbf{r}) = \hat{L}_{\text{low-freq}}(\mathbf{r}) + \hat{L}_{\text{high-freq}}(\mathbf{r}). \tag{4.5}$$

4.6.4 Modeling Local Micro Geometries

We noticed that the quality of modeling of the specular highlights as well as glossy reflections is highly sensitive to the accurate modeling of the complicated local micro geometry over the object surface, particularly for materials with very small roughness. A tiny fluctuation on the surface would significantly alter the specular highlight intensities of the related pixels. Learning without taking the local geometry into account leads to an over-smoothed pixel intensity prediction as the ground truth HDR values tend to naively average with each other with the highlight on or off (Fig. 4.12(a)). We found that by accurately predicting the normals over the object surface can significantly enhance the accuracy of the highlight and reflection prediction. Motivated by [131], where a separately estimated surface normal is used for computing the reflection direction encoding for avoiding linking with the highly noisy density gradient, we propose to learn an additional normal prediction that is solely for the reflection purpose.

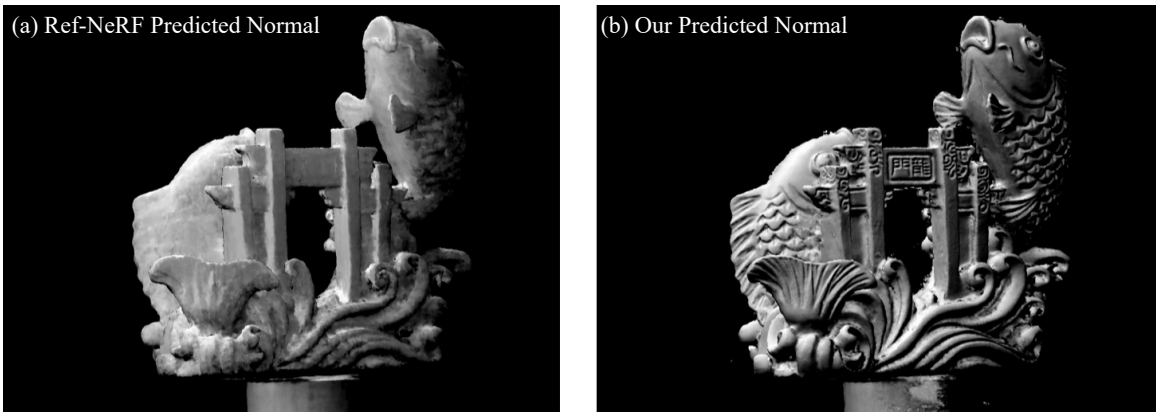


Figure 4.14: Surface normal comparison between RefNeRF predicted normals (left) and the proposed detailed normals for the rendering of micro geometry (right). The gray-scale denotes the dot product between the normal and $\mathbf{h}(\mathbf{r})$. These detailed normals are tied to the RefNeRF normals to quickly find a good starting point—yet they benefit greatly from the high frequency model information and are notably more detailed.

We found that the availability of the abundant relighting data showcasing the appearance of the scene under different lighting condition presents a great opportunity to acquire significantly more accurate surface reconstruction compared to the NeRF-based approach [131], as the changing highlight cues presented under the varying lighting conditions provide strong indications of the accurate normal that otherwise is harder to capture in a fixed lighting environment. Our readily available point-light data that provides the best separation of effects between lighting sources serve the best use here. More precisely, given the set of captured images under the point light training data that could precisely separate the reflection effects from only a single light source, we seek to optimize the separately predicted normal, denoted as $\hat{\mathbf{n}}''(\mathbf{r})$, on top of the two existing normals in the Ref-NeRF framework [131], namely the "density gradient normal" ($\hat{\mathbf{n}}(\mathbf{r})$) and the "Ref-NeRF predicted normal" ($\hat{\mathbf{n}}'(\mathbf{r})$). We name this new normal $\hat{\mathbf{n}}''(\mathbf{r})$ as "predicted normal for reflection purpose only". The normal $\hat{\mathbf{n}}''$ is associated with the ray $\mathbf{r}$ and is obtained via volume rendering accumulation of the normal predictions along $\mathbf{r}$. As we found, under the Ref-NeRF density modeling framework, the density is highly concentrated near the surface [131], and the obtained normal $\hat{\mathbf{n}}''(\mathbf{r})$ is close enough for representing the predicted normal on the surface. We use a simple Blinn-Phong shading model to predict the specular highlight via

$$\hat{L}_{\text{high-freq}}(\mathbf{r}) = \hat{A}(\mathbf{r})\ \text{Dot}(\hat{\mathbf{n}}''(\mathbf{r}), \mathbf{h}(\mathbf{r}))^{\hat{\alpha}(\mathbf{r})}, \tag{4.6}$$

where both $\hat{A}(\mathbf{r})$ and $\hat{\alpha}(\mathbf{r})$ are predicted per-point and accumulated along the ray $\mathbf{r}$ similar to $\hat{\mathbf{n}}''(\mathbf{r})$. $\mathbf{h}(\mathbf{r})$ is the half direction between the lighting direction and the viewing direction as used in the Blinn-Phong model. To train this new normal $\hat{\mathbf{n}}''(\mathbf{r})$, we impose three loss terms to acquire the high quality prediction. First, we compute

the final prediction $\hat{L}_{\mathbf{r}}$ in the same way as in Eq. 4.5, and then impose the pixel loss

$$Loss_{\text{pixel}}(\mathbf{r}) = \|\hat{L}(\mathbf{r}) - L(\mathbf{r})\|_2^2 \tag{4.7}$$

Second, by collecting a small set of pixels that demonstrate extremely high radiance intensities, we directly impose the dot-product loss on these rays:

$$Loss_{\text{dot-product}} = \text{ReLU}(\|I(\mathbf{r})\|_1 - I_0) \cdot (1 - \text{Dot}(\hat{\mathbf{n}}''(\mathbf{r}), \mathbf{h}(\mathbf{r}))), \tag{4.8}$$

where $\text{ReLU}(\|I(\mathbf{r})\|_1 - I_0)$ is a weighting factor and I_0 is a constant threshold for clipping out non-highlight pixels. Lastly, we regularize our predicted normal $\hat{\mathbf{n}}''(\mathbf{r})$ by let it tie to the Ref-NeRF predicted normal $\hat{\mathbf{n}}'(\mathbf{r})$,

$$Loss_{\text{tie-ref}} = \|\hat{\mathbf{n}}''(\mathbf{r}) - \hat{\mathbf{n}}'(\mathbf{r})\|_2^2, \tag{4.9}$$

to ensure $\hat{\mathbf{n}}''(\mathbf{r})$ is not deviating too far.

Our predicted normals for reflection purpose $\hat{\mathbf{n}}''(\mathbf{r})$ demonstrated superior quality compared to other candidates (e.g. $\hat{\mathbf{n}}(\mathbf{r})$ and $\hat{\mathbf{n}}(\mathbf{r})$). It is free from heavy constrain with the density as in $\hat{\mathbf{n}}(\mathbf{r})$ or overly smooth prediction as in $\hat{\mathbf{n}}'(\mathbf{r})$ [131] (Fig. 4.14). With this level of fidelity regarding geometry and normal prediction, our model is enabled to predict specular reflections over glossy surface with higher quality.

4.6.5 Network Optimization

Our learning process undergoes two stages. In the first stage, we primarily aims to learn the geometry as well as the normal $\hat{\mathbf{n}}''(\mathbf{r})$. We train this stage on the point-light data, with the loss function in Eq. 4.7, 4.8, 4.9. Note in this stage, for low-frequency branch prediction, we stick to the point-light modeling in Sec. 4.6.2, and utilize the high-frequency prediction introduced in Sec. 4.6.4 instead of that in Sec. 4.6.3. On the second stage, we fixed the geometry as well as the predicted normals learned in the previous stage, devise the hyper-net (Sec. 4.6.2) as well as the reflection hints prediction (Sec. 4.6.3), and train our model with just the pixel loss (Eq. 4.7) over the image-based lighting training data. Please refer to our supplementary materials for further details.

Transferring from Point-Light to Envmap. Our model demonstrated the degree of flexibility for transferring the learning of the point light data (e.g. from a light stage capture) to the image-based lighting scenario, if the envmap relighting data is not readily available. The key is the compatibility of our proposed reflection hints (Sec. 4.6.3) for both the point-light and the environment map prediction mode. More precisely, when learning the second stage, where we only have the point-light training data, we treat the point-light lighting source as a one-hot high-resolution environment map for learning the high-frequency branch. We fix all the geometry and normal prediction in this stage. Note that now we keep the low-frequency model to be in the

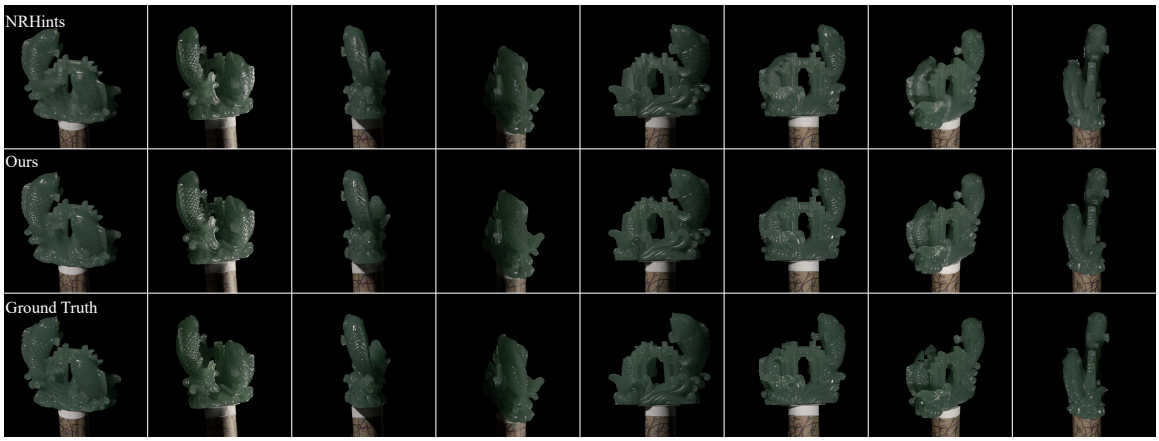


Figure 4.15: We compare our results to the latest state-of-the-art approach on neural relighting [10]. Our results demonstrate clear advantages regarding the handling of the specular highlights. Our approach does not particularly handle hard shadow as in [10] but can easily incorporate their hint machenism into our framework.

point-light mode (Sec. 4.6.2), while we let the high-frequency branch be in the "reflection hint" mode (Sec. 4.6.3). After the optimization of the second stage is finished, we additionally learn a third stage by replacing the point-light-based low-frequency branch with the hyper-net (Sec. 4.6.2) via model distillation. More precisely, we use the learned point-light-based low-frequency branch to predict and aggregate the prediction result for each environment map. We thus fix all the geometry, normal as well as high-frequency prediction and just tune the hyper-net for low-frequency prediction in this stage.

4.6.6 Qualitative Results

We provide qualitative evaluation of our approach both on the point light setting and the environment map setting. For point light please refer to Fig. 4.15-4.16. For environment map please refer to Fig. 4.17-4.18. Our results show clear evidence that our model for hybrid modeling of both the translucency and glossy reflection could achieve promising results for specular highlights, while maintaining good recovering of the other lighting effects, such as subsurface scattering.

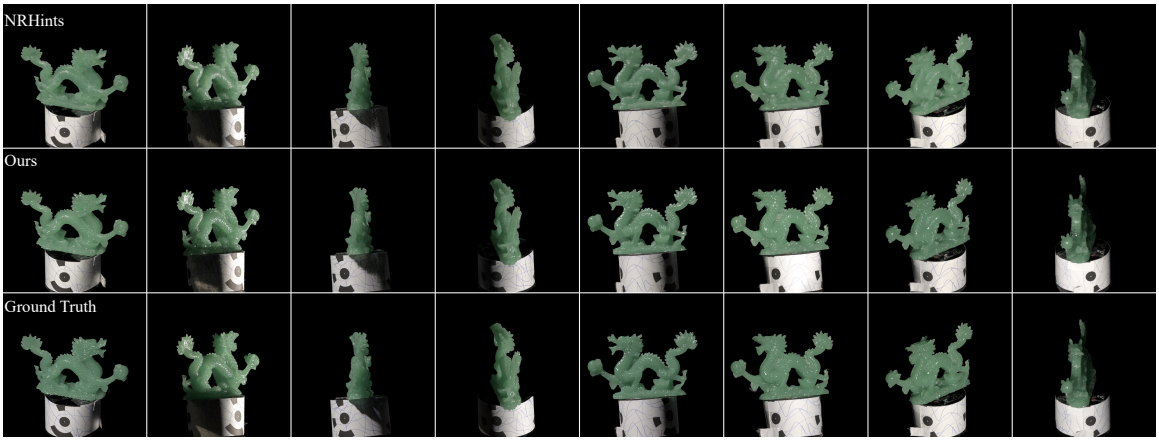


Figure 4.16: We compare our results to the latest state-of-the-art approach on neural relighting [10]. Our results demonstrate clear advantages regarding the handling of the specular highlights. Our approach does not particularly handle hard shadow as in [10] but can easily incorporate their hint machenism into our framework.

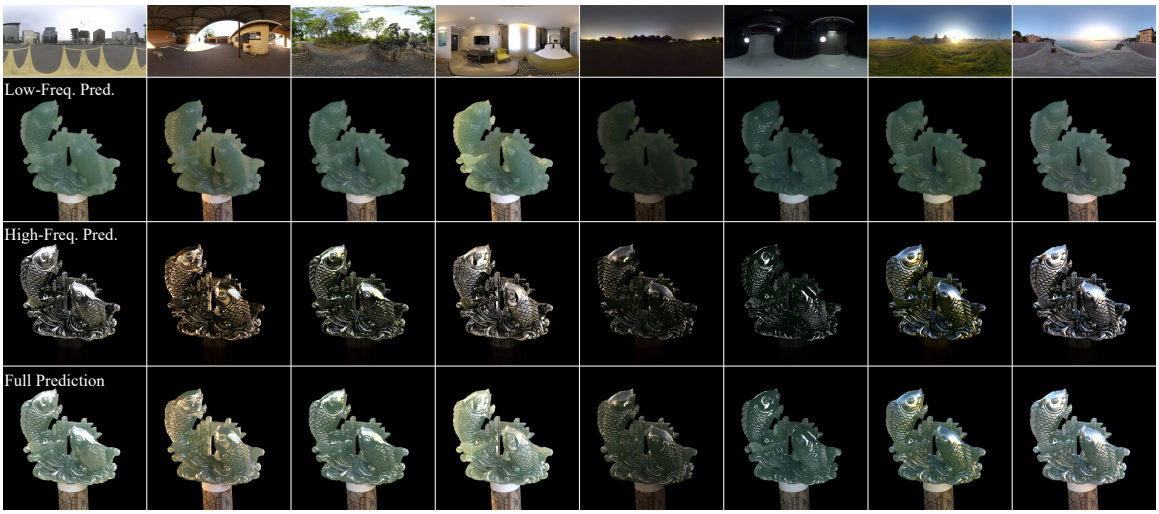


Figure 4.17: Our results as well as the predicted low-frequency and high-frequency results under the image-based lighting (environment map) setting. It is worth pointing out that our training data for these real captured scenes only contain point-light based image.

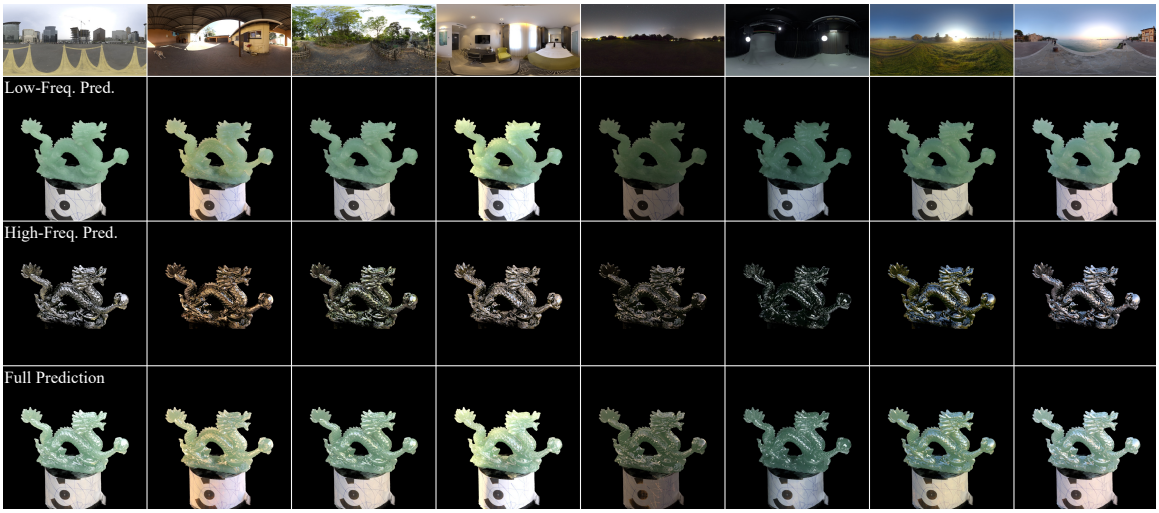


Figure 4.18: Our results as well as the predicted low-frequency and high-frequency results under the image-based lighting (environment map) setting. It is worth pointing out that our training data for these real captured scenes only contain point-light based image.

Chapter 5

Conclusion

5.1 Discussions

Discussions on Differentiable Gradient Sampling (Chapter 2) - While a closed-form solution for the forward and backward propagation for the grid sampling operation would facilitate our training with the losses we proposed in Chapter 2, we further found that a quick double backward implementation could facilitate the training more directly. The double backward process could be summarized below. Suppose the forward pass of a function is denoted as $f(x)$, and we use f to denote the obtained results from the forward pass $f(x)$. The backward pass of f, as defined as g, is denoted as

$$g(x, \frac{\partial L_f}{\partial x}) = \frac{\partial L_f}{\partial f}\frac{\partial f}{\partial x}, \tag{5.1}$$

where L_f denotes the loss signal that is used to compute the gradient g. In our training pipeline, the obtained results g would be a new independent forwarding variable, and also receive its loss L_g. We thus need to implement the backward pass for g, which is referred to as double backward. We could get the backward gradient for the two input arguments of g as

$$\begin{aligned} \frac{\partial L_g}{\partial x} &= \frac{\partial L_g}{\partial g}\frac{\partial g}{\partial x} = \frac{\partial L_g}{\partial g}\frac{\partial \frac{\partial f}{\partial x}}{\partial x} \\ \frac{\partial L_g}{\partial \frac{\partial L_f}{\partial x}} &= \frac{\partial L_g}{\partial g}\frac{\partial g}{\partial \frac{\partial L_f}{\partial x}} = \frac{\partial L_g}{\partial g}\frac{\partial f}{\partial x} \end{aligned} \tag{5.2}$$

By implementing the standard forward (f) and backward (Eq. 5.1) as well as the double backward (Eq. 5.2) pass, it could achieve the same goal as in our closed-form solution while consumes less GPU memory footprint. On the other hand, our proposed training supervision signal (Fig. 2.3, Eq. 2.1) could still further benefit from the double backward implementations and play the critical role for getting the reconstruction result shown in Chapter 2.

Discussions on our neural relighting approach - While our extension (Sec. 4.6) improves our modeling of the glossy reflection on top of the low-frequency subsurface scattering representations, our representation of the newly predicted normal (for reflection purposes) is predicted in a deterministic way, in contrast to the modeling of distribution as in [131]. This leads to two major limitations. One one hand, our representation could not rectify the reflection normal prediction under the image-based lighting condition (suppose we have that type of training data, e.g. synthetic data). Our current training of the reflection normal is all under the point-light setting. On the other hand, while our reflection normal could capture local details of the geometry, it fluctuates among adjacent rays, and the texture reflected in the mirror based on our predicted normal is generally incorrectly twisted.

5.2 Future Directions

Learning generalizable surface reconstructions. With the emergence of Neural Radiance Field [15], more research attention have been paid to approaches for optimizing the geometry with the RGB learning labels. Given that capturing highly accurate 3D surface data in real scenes is extremely difficult, it is indeed of high research interest to explore multi-view data for scene reconstructions. One of the promising data sources that does not require significant manual efforts would be captured video sequences with calibrated camera poses. For instance, the RealEstate-10K dataset [132] collected video sequences from YouTube all capturing static indoor scenes that provided a great chance for learning the generalizable surface reconstruction. Incorporation of the depth data [133] or synthetic scenes could further boost the reconstruction accuracy and generalizability. It is then up to the neural architecture capacities as well as the abudentness of the data to push the reconstruction accuracy up. Another interesting direction would be online acquiring the observations from an embodied robot agent for gradually adapting to the right geometric reconstruction, which is beyond the scope of this thesis.

Neural relighting. The incorporation of the neural components (e.g. NeRF [15]) into the solving of the relighting task has brought promising performance. The transfer vector modeling for relighting further alleviates assumptions of the algorithm concerning global illumination, and subsurface scattering and is significantly more data-driven than other constrained material representations, such as BRDF or BSSRDF. One existing issue concerning the neural field with transfer vector modeling is that neither the point light nor the hyper-net can well model high-frequency details. Ad-hoc solutions such as modeling hard shadows [10] or reflection normal optimizations (Chapter 4) can only address the particular subset of all the high-frequency signals, with translucent shadowing or light refraction highlights are left unrepresented - if not considering the reflection highlights are not yet solved satisfyingly. To our knowledge, finding a unified and principled representation for handling high-frequency signals in

the neural transfer vector modeling framework without introducing all the ad-hoc hints is still an open problem.

Simplifying the lighting data-capturing process and even making it generalizable to unseen objects would also be of great research value and interest. The current data-capturing process is typically coupled with a high-cost light stage, followed by labour intense capturing process and camera pose calibration stage. It is important to seek a solution where only a single lighting condition is given for the multi-view image captures, typically known as the material decomposition problem, where major ambiguity is exhibited as the captured pixel values typically baked all the material properties and the lighting information into a single RGB vector. Taking subsurface scattering effects into account would add to the ambiguities that a translucent object can be precisely fitted by a BRDF-based material model under only one lighting condition. It is important to add proper regularizer, or learn proper priors within the variational learning framework for a high-quality modeling of this challenging relighting problem.

Bibliography

[1] 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*, pages 303–312, 1996.

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

[3] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[4] Farid Yagubbayli, Alessio Tonioni, and Federico Tombari. Legoformer: Transformers for block-by-block multi-view 3d reconstruction. *arXiv preprint arXiv:2106.12102*, 2021.

[5] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, light & material decomposition from images using monte carlo rendering and denoising. *arXiv preprint arXiv:2206.03380*, 2022.

[6] Linjie Lyu, Ayush Tewari, Thomas Leimkühler, Marc Habermann, and Christian Theobalt. Neural radiance transfer fields for relightable novel-view synthesis with global illumination. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pages 153–169. Springer, 2022.

[7] 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.

[8] Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. Iron: Inverse rendering by optimizing neural sdfs and materials from photometric images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5565–5574, 2022.

[9] Xi Deng, Fujun Luan, Bruce Walter, Kavita Bala, and Steve Marschner. Reconstructing translucent objects using differentiable rendering. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022.

[10] Chong Zeng, Guojun Chen, Yue Dong, Pieter Peers, Hongzhi Wu, and Xin Tong. Relighting neural radiance fields with shadow and highlight hints. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023.

[11] Jesse Thomason, Mohit Shridhar, Yonatan Bisk, Chris Paxton, and Luke Zettlemoyer. Language grounding with 3d objects. In *5th Annual Conference on Robot Learning*, 2021.

[12] Frederik Warburg, Ethan Weber, Matthew Tancik, Aleksander Holynski, and Angjoo Kanazawa. Nerfbusters: Removing ghostly artifacts from casually captured nerfs. *arXiv preprint arXiv:2304.10532*, 2023.

[13] 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*, pages 165–174, 2019.

[14] 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*, pages 4460–4470, 2019.

[15] 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.

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

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

[18] 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*, pages 2304–2314, 2019.

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

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

[21] 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*, pages 605–613, 2017.

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

[23] 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*, pages 490–500, 2019.

[24] 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*, pages 10246–10255, 2021.

[25] 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.

[26] 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*, pages 2565–2574, 2020.

[27] 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.

[28] 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.

[29] 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*, pages 5828–5839, 2017.

[30] 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*, pages 1746–1754, 2017.

[31] 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.

[32] 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.

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

[34] 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*, pages 8026–8037, 2019.

[35] 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)*, pages 265–283, 2016.

[36] 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.

[37] 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.

[38] 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*, pages 5939–5948, 2019.

[39] 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*, pages 3504–3515, 2020.

[40] 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.

[41] 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.

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

[43] 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*, pages 414–431. Springer, 2020.

[44] 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*, pages 15598–15607, 2021.

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

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

[47] 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*, pages 7467–7477, 2020.

[48] 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.

[49] 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*, pages 2626–2634, 2017.

[50] 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.

[51] 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*, pages 1251–1261, 2020.

[52] 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.

[53] 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*, pages 628–644. Springer, 2016.

[54] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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

[56] 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.

[57] 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*, pages 204–213, 2021.

[58] 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.

[59] 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*, pages 2974–2983, 2018.

[60] 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, pages 1492–1500, 2017.

[61] Joelle Pineau, Michael Montemerlo, Martha Pollack, Nicholas Roy, and Sebastian Thrun. Towards robotic assistants in nursing homes: Challenges and results. *Robotics and autonomous systems*, 42(3-4):271–281, 2003.

[62] Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, et al. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571, 2020.

[63] Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, et al. Experience grounds language. *arXiv preprint arXiv:2004.10151*, 2020.

[64] Emily M Bender and Alexander Koller. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, 2020.

[65] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.

[66] Ruocheng Wang, Jiayuan Mao, Samuel J Gershman, and Jiajun Wu. Language-mediated, object-centric representation learning. *arXiv preprint arXiv:2012.15814*, 2020.

[67] Drew A Hudson and Christopher D Manning. Learning by abstraction: The neural state machine. *arXiv preprint arXiv:1907.03950*, 2019.

[68] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2891–2903, 2013.

[69] Angel Chang, Manolis Savva, and Christopher D Manning. Learning spatial knowledge for text to 3d scene generation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2028–2038, 2014.

[70] Chaitanya Ahuja and Louis-Philippe Morency. Language2pose: Natural language grounded pose forecasting. In *2019 International Conference on 3D Vision (3DV)*, pages 719–728. IEEE, 2019.

[71] Angela S Lin, Lemeng Wu, Rodolfo Corona, Kevin Tai, Qixing Huang, and Raymond J Mooney. Generating animated videos of human activities from natural language descriptions. *Learning*, 2018:1, 2018.

[72] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. In *Asian Conference on Computer Vision*, pages 100–116. Springer, 2018.

[73] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas J. Guibas. ReferIt3D: Neural listeners for fine-grained 3d object identification in real-world scenes. In *16th European Conference on Computer Vision (ECCV)*, 2020.

[74] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 202–221. Springer, 2020.

[75] Lichen Zhao, Daigang Cai, Lu Sheng, and Dong Xu. 3dvg-transformer: Relation modeling for visual grounding on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2928–2937, October 2021.

[76] Junha Roh, Karthik Desingh, Ali Farhadi, and Dieter Fox. Languagerefer: Spatial-language model for 3d visual grounding. In *Conference on Robot Learning*, pages 1046–1056. PMLR, 2022.

[77] Juil Koo, Ian Huang, Panos Achlioptas, Leonidas Guibas, and Minhyuk Sung. Partglot: Learning shape part segmentation from language reference games. *arXiv preprint arXiv:2112.06390*, 2021.

[78] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33, 2020.

[79] Valts Blukis, Chris Paxton, Dieter Fox, Animesh Garg, and Yoav Artzi. A persistent spatial semantic representation for high-level natural language instruction execution. *arXiv preprint arXiv:2107.05612*, 2021.

[80] Mihir Prabhudesai, Hsiao-Yu Fish Tung, Syed Ashar Javed, Maximilian Sieb, Adam W Harley, and Katerina Fragkiadaki. Embodied language grounding with 3d visual feature representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2220–2229, 2020.

[81] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017.

[82] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[83] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[84] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[85] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. Acronym: A large-scale grasp dataset based on simulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6222–6227. IEEE, 2021.

[86] Manolis Savva, Angel X. Chang, and Pat Hanrahan. Semantically-Enriched 3D Models for Common-sense Knowledge. *CVPR 2015 Workshop on Functionality, Physics, Intentionality and Causality*, 2015.

[87] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10437–10446, 2020.

[88] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019.

[89] Panos Achlioptas, Judy Fan, Robert Hawkins, Noah Goodman, and Leonidas J Guibas. Shapeglot: Learning language for shape differentiation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8938–8947, 2019.

[90] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[91] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[92] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5453–5462, 2021.

[93] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8280–8290, 2022.

[94] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics (TOG)*, 40(6):1–18, 2021.

[95] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan Barron, and Hendrik Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. *Advances in Neural Information Processing Systems*, 34:10691–10704, 2021.

[96] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12684–12694, 2021.

[97] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7495–7504, 2021.

[98] Zhang Chen, Anpei Chen, Guli Zhang, Chengyuan Wang, Yu Ji, Kiriakos N Kutulakos, and Jingyi Yu. A neural rendering framework for free-viewpoint relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5599–5610, 2020.

[99] Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs. In

Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pages 215–224, 1999.

[100] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 145–156, 2000.

[101] Graham Fyffe. Cosine lobe based relighting from gradient illumination photographs. In *SIGGRAPH'09: Posters*, pages 1–1. 2009.

[102] Kaiwen Guo, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, et al. The relightables: Volumetric performance capture of humans with realistic relighting. *ACM Transactions on Graphics (ToG)*, 38(6):1–19, 2019.

[103] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, volume 41, pages 641–676. Wiley Online Library, 2022.

[104] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics (TOG)*, 38(6):1–17, 2019.

[105] Fujun Luan, Shuang Zhao, Kavita Bala, and Zhao Dong. Unified shape and svbrdf recovery using differentiable monte carlo rendering. In *Computer Graphics Forum*, volume 40, pages 101–113. Wiley Online Library, 2021.

[106] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Deep reflectance volumes: Relightable reconstructions from multi-view photometric images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 294–311. Springer, 2020.

[107] Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Neural reflectance fields for appearance acquisition. *arXiv preprint arXiv:2008.03824*, 2020.

[108] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 195–206, 2007.

[109] Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. In *Acm Siggraph*, volume 2012, pages 1–7. vol. 2012, 2012.

[110] Jan Novák, Iliyan Georgiev, Johannes Hanika, Jaroslav Krivánek, and Wojciech Jarosz. Monte carlo methods for physically based volume rendering. In *SIGGRAPH Courses*, pages 14–1, 2018.

[111] Delio Vicini, Vladlen Koltun, and Wenzel Jakob. A learned shape-adaptive subsurface scattering model. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019.

[112] Chengqian Che, Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. Towards learning-based inverse subsurface scattering. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2020.

[113] Ioannis Gkioulekas, Shuang Zhao, Kavita Bala, Todd Zickler, and Anat Levin. Inverse volume rendering with material dictionaries. *ACM Transactions on Graphics (TOG)*, 32(6):1–13, 2013.

[114] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[115] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022.

[116] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 527–536, 2002.

[117] Quan Zheng, Gurprit Singh, and Hans-Peter Seidel. Neural relightable participating media rendering. *Advances in Neural Information Processing Systems*, 34:15203–15215, 2021.

[118] Hong-Xing Yu, Michelle Guo, Alireza Fathi, Yen-Yu Chang, Eric Ryan Chan, Ruohan Gao, Thomas Funkhouser, and Jiajun Wu. Learning object-centric neural scattering functions for free-viewpoint relighting and scene composition. *arXiv preprint arXiv:2303.06138*, 2023.

[119] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16190–16199, 2022.

[120] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022.

[121] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022.

[122] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 333–350. Springer, 2022.

[123] Paul Debevec. A median cut algorithm for light probe sampling. In *ACM SIGGRAPH 2008 classes*, pages 1–3. 2008.

[124] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[125] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[126] Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, and Delio Vicini. Dr. jit: a just-in-time compiler for differentiable rendering. *ACM Transactions on Graphics (TOG)*, 41(4):1–19, 2022.

[127] Ha David, Andrew M Mai, and Quoc V Le. Hypernetworks. In *ICLR*, 2017.

[128] Sai Bi, Stephen Lombardi, Shunsuke Saito, Tomas Simon, Shih-En Wei, Kevyn Mcphail, Ravi Ramamoorthi, Yaser Sheikh, and Jason Saragih. Deep relightable appearance models for animatable faces. *ACM Transactions on Graphics (TOG)*, 40(4):1–15, 2021.

[129] Shun Iwase, Shunsuke Saito, Tomas Simon, Stephen Lombardi, Timur Bagautdinov, Rohan Joshi, Fabian Prada, Takaaki Shiratori, Yaser Sheikh, and Jason Saragih. Relightablehands: Efficient neural relighting of articulated hand models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16663–16673, 2023.

[130] Hendrik Baatz, Jonathan Granskog, Marios Papas, Fabrice Rousselle, and Jan Novák. Nerf-tex: Neural reflectance field textures. In *Computer Graphics Forum*, volume 41, pages 287–301. Wiley Online Library, 2022.

[131] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022.

[132] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

[133] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10786–10796, 2021.