Learning-based Inverse Rendering of Complex Indoor Scenes with Differentiable Monte Carlo Raytracing

Jingsen Zhu
zhujingsen@zju.edu.cn
State Key Lab of CAD&CG, Zhejiang University
China

Fujun Luan
fuan@adobe.com
Adobe Research
USA

Yuchi Huo
huoyuchi.sc@gmail.com
State Key Lab of CAD&CG, Zhejiang University
China

Zihao Lin
zhaolin@zju.edu.cn
State Key Lab of CAD&CG, Zhejiang University
China

Zhihua Zhong
zhongzhihua@zju.edu.cn
State Key Lab of CAD&CG, Zhejiang University
China

Dianbing Xi
dh.xi@zju.edu.cn
State Key Lab of CAD&CG, Zhejiang University
China

Jiaxiang Zheng
xuanfeng@qunhemail.com
KooLab, Manycore
China

Rui Tang
ati@qunhemail.com
KooLab, Manycore
China

Hujun Bao
bao@cad.zju.edu.cn
State Key Lab of CAD&CG, Zhejiang University
China

Rui Wang
rwang@cad.zju.edu.cn
State Key Lab of CAD&CG, Zhejiang University
China

Holistic inverse rendering

Multiple complex object insertion

Material editing

Figure 1: We present a learning-based approach for inverse rendering of complex indoor scenes with differentiable Monte Carlo raytracing. Our method takes a single indoor scene RGB image as input and automatically infers its underlying surface reflectance (represented by microfacet GGX), geometry, and spatially-varying illumination (first column). Consequently, this enables us to perform photorealistic editing of the scene, such as inserting multiple complex virtual objects (second column, note that the inserted models are highly glossy) and editing surface materials faithfully with global illumination (last two columns, note that the wall is modified to a mirror that correctly presents specular reflections of the kitchen, and the glossy cooktop is modified to Lambertian appearance).

ABSTRACT

Indoor scenes typically exhibit complex, spatially-varying appearance from global illumination, making inverse rendering a challenging ill-posed problem. This work presents an end-to-end, learning-based inverse rendering framework incorporating differentiable Monte Carlo raytracing with importance sampling. The framework takes a single image as input to jointly recover the underlying geometry, spatially-varying lighting, and photorealistic materials. Specifically, we introduce a physically-based differentiable rendering layer with screen-space ray tracing, resulting in more realistic specular reflections that match the input photo. In addition, we create a large-scale, photorealistic indoor scene dataset with significantly richer details like complex furniture and dedicated decorations. Furthermore, we design a novel out-of-view lighting network with uncertainty-aware refinement leveraging hypernetwork-based neural radiance fields to predict lighting outside the view of the input photo. Through extensive evaluations on common benchmark datasets, we demonstrate superior inverse rendering quality of our method compared to state-of-the-art baselines, enabling various applications such as complex object insertion and material
editing with high fidelity. Code and data will be made available at https://jingsenzhu.github.io/invrend.

**CCS CONCEPTS**
- Computing methodologies → Rendering, Ray tracing.

**KEYWORDS**
ray tracing, lighting estimation, inverse rendering

**ACM Reference Format:**
Jingsen Zhu, Fujun Luan, Yuchi Huo, Zihao Lin, Zhiluua Zhong, Dianbing Xi, Jiaxiang Zheng, Rui Tang, Hujuan Bao, and Rui Wang. 2022. Learning-based Inverse Rendering of Complex Indoor Scenes with Differentiable Monte Carlo Ray tracing. In *SIGGRAPH Asia 2022 Conference Papers (SA ’22 Conference Papers)*, December 6–9, 2022, Daegu, Republic of Korea. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3550469.3555407

1 INTRODUCTION

Inverse rendering of complex indoor scenes has been a long-standing challenge in computer graphics and vision. Given a single real-world image, global illumination effects such as shadows, specular highlights, and glossy interreflections are baked into the observed pixel values, imposing a particularly difficult task of simultaneously recovering the underlying scene geometry, spatially-varying surface reflectance, and arbitrary unknown illumination. Traditional optimization-based approaches rely on dedicatedly designed regularization and hand-crafted priors to tackle this problem. Unfortunately, such methods often fail in real-world scenarios due to overly simplified assumptions, leading to noticeable artifacts in both the decomposition and re-rendered results.

On the other hand, recent advances in inverse rendering [Li et al. 2020; Srinivasan et al. 2020; Wang et al. 2021b] leveraging deep learning methods have demonstrated impressive results on such scene inference tasks, where the underlying physical priors are supposed to be learnt automatically through an offline, supervised training process, typically on a large-scale, synthetic or labeled real training dataset of complex indoor scenes. Note that, it is extremely difficult, if not impossible, to generate ground truth labels of spatially-varying illumination and materials of an arbitrary real-world scene, and hence one crucial keypoint to the success of such methods is the fidelity and photorealism of the synthetic training data. Another essential factor strongly influencing the inference accuracy is the network structure design. Intuitively speaking, since inverse rendering is inverting the physical light transport, a physically-based differentiable rendering layer regularizing the parameter space can act as a meaningful prior, improving the robustness and generalization capability of the neural network regarding material and lighting decomposition, and thus also the geometry estimation in return. Consequently, the performance of these learning-based inverse rendering methods heavily depends on: 1) the quality of the training datasets, and 2) the design of the neural network architecture.

To address the aforementioned challenges, we propose a novel Monte Carlo differentiable rendering layer with importance sampling to faithfully simulate the physical light transport of an indoor scene. Experiments show that this is especially helpful in restoring the specular reflections of a given scene, and our method produces much more realistic re-rendered results comparing previous baselines. Unlike previous work that directly uses the local feature at a ray-surface intersection point, our approach importance samples the local incident radiance field of it via screen space ray tracing (SSRT) and uncertainty-aware, hypernetwork-based out-of-view lighting estimation. To facilitate training, we introduce a large-scale (~4000) complex indoor scene dataset, *INTERIORVERSE*. As far as we know, our dataset contains the highest quality with rich details compared to existing indoor scene datasets (e.g., OpenRooms [Li et al. 2021] or SUNCG [Song et al. 2017]), including complex furniture and dedicated decorations procedurally designed by professional digital artists, rendered with physically-based GGX model [Walter et al. 2007] using a modern GPU-based path tracing engine.

Concretely, our contributions include:

- A learning-based monocular inverse rendering framework of complex indoor scenes that recovers albedo, surface normal, depth, metallic, and roughness from a single indoor scene image.
- A novel Monte Carlo differentiable rendering layer with importance sampling, which correctly estimates the local incident radiance field using screen space ray tracing.
- An uncertainty-aware out-of-view light network leveraging hypernetwork-based neural radiance fields for robust out-of-view lighting estimation.
- A high-quality, large-scale complex indoor scene dataset, *INTERIORVERSE*, that contains rich details with high fidelity.

2 RELATED WORK

**Inverse Rendering of Indoor Scenes.** Inverse rendering attempts to reconstruct geometry and spatially-varying material and lighting information from monocular (which is our case) or multiple RGB images. Most previous methods only recognize one or part of the above attributes. Geometry reconstructions, including depth estimation and surface normal reconstruction, has been widely studied [Eigen and Fergus 2015; Liu et al. 2019]. Most material reconstruction methods are only able to either estimate diffuse albedo [Barron and Malik 2013; Karsch et al. 2014; Li and Snavely 2018] or classify material categories [Bell et al. 2015]. For lighting estimation, recent deep learning methods have made progress in estimating global [Gardner et al. 2019, 2017] and even spatially-varying [Garon et al. 2019; Li et al. 2020; Song and Funkhouser 2019] lighting conditions. Recent works attempt to predict multiple intrinsics jointly by a holistic inverse rendering framework. Li et al. [2020] proposed a method to reconstruct disentangled geometry, spatially-varying reflectance and lighting from a single RGB indoor scene image.

**Lighting Estimation and Relighting.** Light estimation is one of the sub-tasks of inverse rendering. Most previous works ignore spatially-varying effects and predict a single environment map for the whole scene [Gardner et al. 2017; Munkberg et al. 2022; Sengupta et al. 2019]. Indoor scenes suffer from spatial variations, thus recent work explores spatially-varying lighting estimation for indoor scenes. The representation of spatially-varying illumination includes environment maps, per-pixel spherical lobes [Li et al. 2020] (spherical Harmonics/Gaussians), or 3D voxel grids [Wang et al. 2021b]. Relighting is also a widely-studied relevant task. Griffiths et al. [2022] leverages screen-space method to detect occlusion and
Neural Scene Representations. Neural representations are a rapidly growing area of research. Recent advances include voxels [Sun et al. 2021; Yu et al. 2021a], hashgrids [Müller et al. 2022], point clouds [Aliev et al. 2020], and neural implicit functions [Mildenhall et al. 2020; Wang et al. 2021a; Variv et al. 2021, 2020]. Neural radiance fields (NeRFs) [Mildenhall et al. 2020] represents scenes as neural implicit functions, encoding a scene as a continuous volumetric radiance field of color and density. With volume rendering, a NeRF can synthesize novel view images with promising results. Our proposed method uses a NeRF as the representation of the out-of-view area of the scene (Sec. 4.3).

Differentiable Rendering. A number of recent inverse rendering works utilize differentiable rendering to recover complex light transport effects. Some recent works have proposed general-purpose physically-based differentiable renderers [Li et al. 2018a; Nimier-David et al. 2019]. Zhang et al. [2020] and Zeltner et al. [2021] discussed a rigorous theory of differentiable light transport and Monte-Carlo combinations. These physically-based methods achieve high-quality global illumination effects at the cost of substantial performance overhead. Some differentiable rendering techniques are customized for specific purpose such as texture [Nimier-David et al. 2021], split-sum lighting and mesh extraction [Munkberg et al. 2022]. Our method designs a Monte-Carlo based in-network differentiable rendering layer to recover the appearance of indoor scenes (Sec. 4.4).

Indoor Scene Datasets. Supervised learning requires a large database of indoor scene images and their corresponding ground truth geometry, material, and lighting for network training. Datasets include 3D shape models [Chang et al. 2015], real-world scans [Chang et al. 2017; Dai et al. 2017], and scene datasets [Li et al. 2018b, 2021; Savva et al. 2017; Song et al. 2017], which can be classified as either real or synthetic data. Real datasets provide real-world images and geometry, while synthetic datasets provide arbitrary scene annotations for inverse rendering, some of which, such as materials and illumination, are difficult to acquire from real world. To the best of our knowledge, InteriorNet [Li et al. 2018b] and OpenRooms [Li et al. 2021] are so far the highest-quality public indoor datasets with spatially-varying photorealistic material and illumination annotations. Unfortunately, InteriorNet provides only LDR results, while OpenRooms provides only lighting information on the scene surface (instead of at any 3D location), and lacks the complexity of material and furniture variations. We present a new indoor scene HDR dataset to tackle their shortcomings.

3 INTERIORVERSE: A LARGE-SCALE, PHOTOREALISTIC INDOOR SCENE DATASET

A high-quality dataset is crucial for learning-based inverse rendering. It’s extremely difficult to acquire spatially-varying material and lighting ground truth in real world complex indoor scenes. Therefore, we render a synthetic dataset to supervise training. The SUNCG dataset [Song et al. 2017] is a manually-created large-scale dataset for indoor scenes, but they use non-physical Phong BRDF and render with OpenGL, which severely limits its photorealism. PBRS [Zhang et al. 2017] and CG-PBR [Sengupta et al. 2019] datasets are rendered with physically-based renderers, but both still use Phong BRDF and do not provide spatially-varying lighting ground truth for an arbitrary 3D location. InteriorNet [Li et al. 2018b] is a large-scale photorealistic indoor scene dataset providing multiple camera views and panoramas, but the images they provide are LDR, limiting the dynamic range of illumination. OpenRooms [Li et al. 2021] is by far the only HDR dataset with spatially-varying lighting rendered using physically-based microfacet BRDF. However, as shown in Fig. 2, it presents overly simplified furniture models and layouts, insufficient material and lighting variations, leading to less faithful appearance comparing to real world data consequently.

In this work, we create a new high-quality indoor scene dataset called INTERIORVERSE, which has the following advantages in data quality over existing alternatives: (1) the scene layouts of our dataset have richer details, including complex furniture and decorations. (2) Our dataset is rendered with GGX BRDF model [Walter et al. 2007], which has stronger material modeling capability than any BRDF models that existing indoor scene datasets use. (3) Our dataset provides not only pixel-wise surface environment maps, but also contains environment maps located anywhere in the 3D scene space. Fig. 2 compares some example scenes in our dataset and OpenRooms, showing our dataset’s higher scene quality.

4 NETWORK DESIGN

Our inverse rendering framework takes a single image of indoor scene as input and jointly predicts the spatially-varying material, geometry and lighting of the scene, and can further re-render the appearance of the input image. Fig. 3 overviews the architecture of our framework, which consists of three parts: material-geometry network (§4.1), lighting network (§4.2 and §4.3), and a differentiable Monte-Carlo rendering layer (§4.4). The material-geometry network is an end-to-end convolutional network which directly predict the reconstruction results. The lighting network LightNet is comprised of three sub-parts: A Resnet34 encoder to produce local feature map from the input image (like pix2NeRF [Yu et al. 2021b]), a...
Figure 3: Overview of the pipeline. On the left, we show the workflow throughout our inverse rendering framework: (i) The spatially-varying material and geometry maps are predicted by MGNet (ii) According to the predicted material, geometry and view direction associated with each pixel point p, a BRDF importance sampling is performed to generate per-pixel incident directions d_i (iii) We use screen-space ray tracing to trace the source point s of the query light. The corresponding local feature vector is extracted from feature map F via projection of point s. (iv) The feature is passed to LightNet along with light direction and auxiliary G-Buffer information to predict the incident radiance ˜L_i (v) Monte-Carlo integration (Eq. 7) is used to calculate the final re-rendered result. On the right, we show our out-of-view light estimation. We use a hypernetwork to predict the weights of the NeRF MLP and volume render the background lighting.

screen-space ray tracer to trace the source of the light, and a final MLP decoder to predict the lighting radiance result. The rendering layer takes G-Buffers and lighting as input, and uses Monte Carlo ray tracing to reproduce realistic rendering results.

4.1 Material and Geometry Network
The input to our material and geometry prediction network MGNet is a single high dynamic range image, which can be directly obtained from our synthetic dataset. For real-world photos, we preprocess them with an inverse gamma correction. We use a single DenseNet121 [Huang et al. 2017] encoder to extract deep features of the material and shape parameters of the scene with different depth, as well as four separate decoders to obtain the final predicted albedo (A), material (M), normal (N), and depth (D), where M consists of two parts: roughness R and metallic M_e. While decoding neural features of different depths, upsampling and skip links are used to preserve multi-level details. Please refer to supplementary material for the detailed architecture of MGNet.

4.2 Lighting Network
We now describe our approach to predict any incident light intensity L_i(p, d) at point p with direction d from a single image. We fix our coordinate system as the view space of the input image and specify position p and light direction d in this coordinate system.

Given an input image I of a scene, we first extract a feature map F = E(I), where E is an encoder with ResNet34 [He et al. 2016] architecture. For any location x in the scene, we can retrieve the corresponding image feature by projecting x onto the image coordinates π(x) using camera intrinsics and extract the local feature vector F[π(x)]. Instead of directly using the local feature at incident point p, we trace the ray from p with direction −d to point s in the scene, which can be treated as a virtual point light of L_i(p, d). We extract the local feature vector F[π(s)]. The local feature is then passed into the final MLP decoder f, along with view direction d and some local G-Buffers (diffuse albedo K_d, specular albedo K_s, normal N and roughness R) at π(s) G[π(s)], as

\[ s = \text{trace}(p, -d), \]
\[ L_i(p, d) = f(γ(d), F[π(s)], G[π(s)]), \]

where γ(·) is positional encoding function which is common used in NeRF [Mildenhall et al. 2020] to capture the high-frequency details in the data. The trace operation is implemented by screen space ray tracing (SSRT). We show our pipeline schematically in Fig. 3.

Our screen space ray tracer works on the depth map of the scene. It takes depth map D, starting point p, and the tracing direction d as inputs. The screen space ray tracer performs ray marching through pixels from the start point. At each step, the current depth of the ray is updated and compared with the surface depth of the pixel. If the ray depth is larger, it indicates that the ray has passed through the pixel surface, i.e., an intersection has occurred. Otherwise, it continues ray marching to an adjacent pixel until hitting the edge of the image.

4.3 Uncertainty-Aware Out-of-View Lighting Network
A limitation of screen space ray tracing is that the traced ray does not necessarily intersect within the field of view of the image. Therefore, an additional network (named “out-of-view lighting network”) is designed to handle lights from the out-of-view area of the scene. The design of our out-of-view lighting network is inspired by Neural Radiance Fields (NeRF) [Mildenhall et al. 2020], which uses an MLP to represent the scene and uses volume rendering to predict the radiance of a ray. In the original version, the weights of the MLP are trained scene-specifically. Instead, we leverage hypernetwork [Ha et al. 2016] to reconstruct out-of-view lighting by predicting the scene-specific weights of the NeRF MLP, and then query the radiance by the same volume rendering and alpha compositing techniques.
The out-of-view lighting network architecture is shown in Fig. 3 (right-hand side). Given the input image I, we first extract a global feature $F_g = G(I)$, where $G$ is an encoder with ResNet34 architecture (separate from the encoder in Section 4.2). Then, $F_g$ is taken by hypernetwork $H$ and the MLP’s weights $\Phi$ are returned. To query an incident light intensity $L_i(p, d)$, we sample $N$ 3D points $\{x_i = p - td\}_{i=1}^N$ on ray $\langle p, -d \rangle$. With positional encoding $y$ and NeRF MLP $f$, density $\sigma$ and RGB color $c$ are returned. The complete process can be formulated as:

$$
\Phi = H(G(I)),
$$

$$
\{\sigma_i, c_i = f(y(x_i); \Phi)\}_{i=1}^N.
$$

Then light intensity $L$ can be composited by

$$
\hat{L} = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) c_i, \text{ where } T_i = \exp \left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right),
$$

where $\delta_i = t_{i+1} - t_i$ is the distance between adjacent samples. Due to performance consideration, our NeRF MLP is a small-scale network and does not take ray direction $d$ as the MLP input like the original paper [Mildenhall et al. 2020] does.

The out-of-view lighting network is capable of predicting lighting anywhere in the scene. We now describe how we use it to refine the light predictions within field-of-view. Screen space ray tracing has a limitation that it may report some false positive of intersections. For real intersections, the difference between surface depth and ray depth is small, while the depth difference will increase when the intersection is a false positive. We model it as the “uncertainty” of SSRT, which is activated by hyperbolic tangent function: $\hat{u} = \tanh(10 \Delta d)$ where $\Delta d \in [0, \infty)$ is the depth difference. The refined light prediction is then formulated as

$$
\hat{L}_{\text{refined}} = (1 - u) \times \hat{L}_{\text{SSRT}} + u \times \hat{L}_{\text{out-of-view}},
$$

where $\hat{L}_{\text{SSRT}}$ is the light prediction by our SSRT-based lighting network and $\hat{L}_{\text{out-of-view}}$ is the light prediction by our out-of-view lighting network. When uncertainty value $u$ is large, screen space ray tracing becomes untrusted and the final prediction is dominated by out-of-view lighting prediction. We ablate between using only out-of-view network predictions and using full model predictions combined with Eq. 6 in our supplementary material.

### 4.4 Rendering Layer

Unlike Li et al. [2020] which discretizes the incident hemisphere to approximate the integration, we leverage differentiable Monte Carlo raytracing to produce photorealistic re-rendering results. Given sample count $N$, we use BRDF importance sampling to sample $N$ ray directions $\{d_i\} = \{\phi_i, \theta_i\}$ according to view direction, surface normal and material parameters (roughness and metallic) at pixel point $p$. We then perform screen-space raytracing according to $d_i$ to trace the source point and predict the radiance of the corresponding direction $L_i$ from LightNet. The rendering layer computes the unbiased re-rendered image by

$$
\hat{I} = \frac{1}{N} \sum_{i=1}^N f_{\hat{L}}(v, d_i; \hat{A}, \hat{N}, \hat{R}, \hat{M}_i) L_i \cos \theta_i,
$$

where $f_{\hat{L}}(v, d_i; \hat{A}, \hat{N}, \hat{R}, \hat{M}_i)$ is the BRDF evaluation value and $p(v, d_i; \hat{A}, \hat{N}, \hat{R}, \hat{M}_i)$ is the probability distribution function (PDF) value of BRDF importance sampling, and $v$ is the view direction. Our importance sampling rendering layer can produce much more realistic re-rendered images compared to [Li et al. 2020], especially in specular reflections and highlights. As shown in Fig. 4, our rendering layer is capable of recovering specular reflections on the glossy floor, while the rendering layer used by [Li et al. 2020] produces significant artifacts. The artifacts of [Li et al. 2020]’s discretization rendering algorithm are caused by the deterministic discrete direction sampling at each pixel, which is likely to miss important directions in the specular BRDF term. The missing of important reflection directions results in interleaved patterns in the re-rendered result. In contrast, our importance sampling strategy can faithfully recover high-frequency reflections on glossy surfaces.

### 5 Training

We train our network models with the supervision of ground truth $\{I, A, N, D, R, M, L\}$ from our synthetic INTERIORVERSE dataset, where $A, N, D, R, M, L$ denote albedo, normal, depth, roughness, and metallic, respectively, and $L$ denotes spatially-varying lighting ground truth.

For geometry and material reconstruction, we use direct supervision to calculate the error between ground truth and network prediction. For lighting estimation, inspired by prior work [Li et al. 2020; Wang et al. 2021b], to encourage photorealistic scene appearance reconstruction, we additionally use a differentiable in-network rendering layer to re-render the image according to the predicted material, geometry, and lighting, and try to recover the original input image through an image loss. Note that, unlike prior work, our render layer incorporates physically-based Monte Carlo sampling via screen space ray tracing, which explicitly regularizes the physical parameter space with GGX importance sampling. As we will demonstrate later, this makes our method significantly more robust to handle specular reflections in the interior scene.

#### 5.1 Material-Geometry Network

We train MGNet with the weighted combination of material losses (albedo loss $L_{\text{albedo}}$, roughness-metallic loss $L_{\text{material}}$) and geometry losses (normal loss $L_{\text{normal}}$ and depth loss $L_{\text{depth}}$):

$$
L_{\text{MGNet}} = \lambda_{\text{a}} L_{\text{albedo}} + \lambda_{\text{n}} L_{\text{normal}} + \lambda_{\text{m}} L_{\text{material}} + \lambda_d L_{\text{depth}}.
$$
We add perceptual loss [Johnson et al. 2016] in the albedo, normal and material term, which helps to recognize the semantic boundaries in the image. The detailed definitions of separate losses and weights are presented in the supplemental material.

### 5.2 Lighting Network

We train LightNet with the weighted combination of direct light supervision loss $L_{\text{light}}$ and re-rendering loss $L_{\text{re-render}}$:

$$L_{\text{LightNet}} = L_{\text{light}} + \lambda_r L_{\text{re-render}} \tag{9}$$

where $L_{\text{light}}$ is the HDR supervision loss function proposed by [Mildenhall et al. 2021], while $L_{\text{re-render}}$ is an $L_2$ loss between the re-rendered image and the original image. Please refer to supplemental material for the detailed definition of $L_{\text{light}}$ and $L_{\text{re-render}}$.

We find that re-rendering loss can significantly improve the lighting prediction, especially on specular surfaces. This benefit comes from enforcing the network to learn correct pixel brightness in $\hat{I}$, thus producing accurate lighting supervision in the scene and preventing blurry or spot artifacts in the re-rendered image. Ablation studies on the usage of re-rendering loss are presented in the supplemental material.

### 5.3 Training Scheme

We use a progressive training scheme to train our model in the order of data dependencies between different components of our framework. We first train material-geometry module to ensure correct predictions of albedo, normal, roughness, metallic, and depth. This is because our lighting network depends on these properties (e.g., SSRs depend on depth, and MLP decoder depends on G-Buffers). Then we train lighting module with re-rendering loss.

### 6 EXPERIMENTS

#### 6.1 Experiment Settings

**Training data.** We train our network on our new photorealistic indoor scene dataset, introduced in Sec. 3. When evaluating on real world data, we also fine-tune our model on IIW dataset [Bell et al. 2014] for albedo and NYUv2 [Silberman et al. 2012] for depth and normal. Please refer to our supplementary material for more details on training and evaluation data.

**Baselines.** We compare our method with Li et al. [2020], which is the state-of-the-art holistic inverse rendering framework for indoor scenes. To ensure a fair comparison, we fine-tune [Li et al. 2020] on our new dataset, which significantly improves its performance (Fig. 5). For lighting prediction, we compare with [Li et al. 2020] as well as another state-of-the-art lighting estimation method Lighthouse [Srinivasan et al. 2020], which requires a stereo image pair as input instead of a single image.

#### 6.2 Evaluation of Material and Geometry

We evaluate material (albedo, roughness, and metallic) and geometry (normal and depth) prediction on InteriorVerse synthetic indoor dataset, as well as real-world dataset (NYUv2 dataset [Silberman et al. 2012] for geometry and IIW dataset [Bell et al. 2014] for albedo).

#### 6.3 Evaluation of Lighting

**Evaluation on virtual object insertion.** We evaluate our lighting estimation method on a crucial augmented reality application: virtual object insertion. With the help of screen space ray tracing and
the Monte Carlo rendering layer, we can achieve promising results in specular reflection effects. Fig. 6 shows results of our method compared to baselines, consisting of both synthetic data and real world images. In order to emphasize the ability to recover high-frequency lighting details, the materials of the inserted objects are highly specular. For synthetic data, we insert complex objects and ground truths are provided. Li et al. [2020]’s lighting estimation is 2D spatially-varying, which cannot handle 3D points far from 2D surfaces. Moreover, their Spherical Gaussian lighting representation is incapable of capturing high-frequency angular details. Therefore, the appearance of inserted highly specular objects does not contain sharp reflections. In contrast, our method produces photorealistic shading and specular highlights on the inserted object. For real world data, we choose to insert highly specular spheres. The reflection on the sphere is supposed to be consistent with the surrounding environments. Li et al. [2020] also fails in this task, due to its low-frequency lighting predictions, while our method manages to faithfully recover angular details of the surrounding environment on the inserted sphere.

We also compare our method with another state-of-the-art lighting estimation method Lighthouse [Srinivasan et al. 2020], which requires a stereo pair of images as input. To show our method’s cross-domain ability, we evaluate on Lighthouse’s test set from InteriorNet [Li et al. 2018b] without fine-tuning our network. As shown in Figure 7, our method outperforms Lighthouse, even with a lower number of input images and a potential domain gap. We can observe that Lighthouse’s lighting prediction has significantly less variation in lighting intensity. This may be because Lighthouse is trained from LDR panoramas, and cannot handle HDR lighting.

We also explore more applications of our lighting estimation method, including re-rendering and scene material edit. Please refer to our supplementary material for these additional results.

7 CONCLUSION AND LIMITATIONS

We present a learning-based method for inverse rendering of complex indoor scenes. Our approach handles spatially-varying illumination and faithfully recovers specular reflections thanks to the differentiable Monte Carlo rendering layer, enabling photorealistic editing such as complex object insertion and material change. Lastly, we introduce a large-scale indoor dataset, InteriorVerse, which contains much richer details than existing alternatives.

There are some limitations of our method. Our out-of-view lighting network is not capable of predicting high-frequency details due to its limited network capacity. Monte Carlo sampling would also lead to noisy re-render results, and raising the required sample budget can be computationally expensive. Further, emission of light sources is not supported currently, which we leave as future work.

ACKNOWLEDGMENTS

This work was supported in part by Key R&D Program of Zhejiang Province (No. 2022C01025), NSFC (No. 61872319), the Fundamental Research Funds for the Central Universities, Zhejiang Lab (121005-PJ2101), and Information Technology Center and State Key Lab of CAD&CG, Zhejiang University.
