Neural Light Field Estimation for Street Scenes with Differentiable Virtual Object Insertion

Zian Wang[1,2,3], Wenzheng Chen[1,2,3], David Acuna[1,2,3], Jan Kautz[1], and Sanja Fidler[1,2,3]

[1]NVIDIA [2]University of Toronto [3]Vector Institute
{zianw,wenzchen,dacunamarrer,jkautz,sfidler}@nvidia.com

Abstract. We consider the challenging problem of outdoor lighting estimation for the goal of photorealistic virtual object insertion into photographs. Existing works on outdoor lighting estimation typically simplify the scene lighting into an environment map which cannot capture the spatially-varying lighting effects in outdoor scenes. In this work, we propose a neural approach that estimates the 5D HDR light field from a single image, and a differentiable object insertion formulation that enables end-to-end training with image-based losses that encourage realism. Specifically, we design a hybrid lighting representation tailored to outdoor scenes, which contains an HDR sky dome that handles the extreme intensity of the sun, and a volumetric lighting representation that models the spatially-varying appearance of the surrounding scene. With the estimated lighting, our shadow-aware object insertion is fully differentiable, which enables adversarial training over the composited image to provide additional supervisory signal to the lighting prediction. We experimentally demonstrate that our hybrid lighting representation is more performant than existing outdoor lighting estimation methods. We further show the benefits of our AR object insertion in an autonomous driving application, where we obtain performance gains for a 3D object detector when trained on our augmented data.

Keywords: Lighting Estimation, Image Editing, Augmented Reality

1 Introduction

In this work, we address the task of outdoor lighting estimation from monocular imagery, specifically focusing on street scenes, as shown in Fig. 1. This is an important task, as it enables virtual object insertion that can cater to many downstream domains [8,28,18,35,10], such as virtually inserting newly planned buildings for architectural visualization, realistically rendering game characters into surroundings, or as a way to augment real datasets with objects that are otherwise hard to record in the real world, such as road debris and exotic animals, for the purpose of training more robust and performant computer vision models.

Lighting estimation for AR applications needs to account for complex 5D light transport [1], *i.e.* a function of spatial location and viewing direction. With

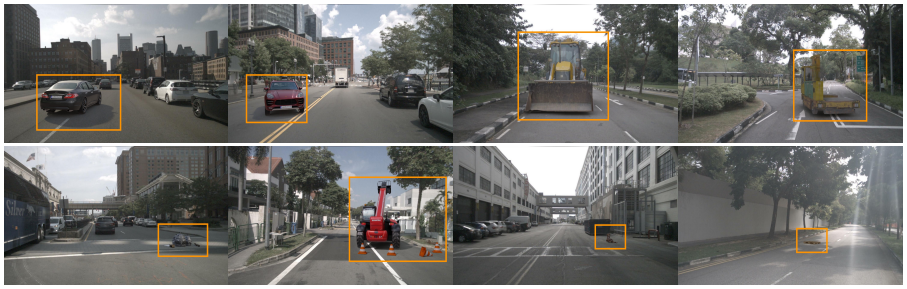

Fig. 1. We estimate lighting and perform virtual objects insertion in real street scenery. We insert cars and heavy vehicles (top), and composite rare but safety-critical scenarios with garbage, construction site, a dog, and debris (bottom). 3D assets provided courtesy of TurboSquid and their artists Hum3D, be fast, rabser, FirelightCGStudio, amaranthus and 3DTree_LLC.

usually a limited field-of-view observed from input, the task of estimating the light field is challenging and ill-posed. An additional challenge encountered for outdoor scenes, in contrast to indoor scenes, is the extreme high dynamic range (HDR) of the sun, which is critical to estimate correctly in order to render cast shadows. Existing literature on lighting estimation usually tackles a simplified problem setting. [37,34] focus on spatially-varying effects but do not handle HDR intensities. In contrast, methods that focus on HDR and predict parametric sky [17,39] or utilize learned sky models [16] typically ignore the spatially-varying effects and lack high-frequency details. These limitations not only result in inaccurate lighting estimation but also hamper virtual object insertion effects.

In this paper, we propose a unified approach that overcomes the previously mentioned limitations, estimating the HDR scene light field from a single image. Tailored to outdoor scenes, we estimate a hybrid lighting representation that comprises of two components: an HDR sky dome and a volumetric lighting representation for the surrounding scene. We employ a learned latent vector to represent the sky dome inspired by [16], which can be decoded into an HDR environment map that is designed to model the strong intensity of the sun. We adopt the volumetric spherical Gaussian representation [37] to represent the non-infinity surroundings such as road and buildings. The two components naturally combine with volume rendering and demonstrate superiority over prior works [16,37]. We further design a physics-based object insertion formulation that renders the inserted objects and their shadows cast on the scene. We utilize ray-tracing to capture the second-order lighting effects, which is fully differentiable with respect to the lighting parameters. We train our method with supervised and self-supervised losses, and show that adversarial training over the composited AR images provides complementary supervisory signal to improve lighting estimation.

Our method outperforms prior work in the tasks of lighting estimation and photorealistic object insertion, which we show through numerical results and a user study. We further showcase our virtual object insertion through the

application of 3D object detection in autonomous driving. Our approach, which can render synthetic 3D objects into real imagery in a realistic way, provides useful data augmentation that leads to notable performance gains over the vanilla dataset, and a naive insertion method that does not account for lighting.

2 Related Work

Lighting estimation aims to predict an HDR light field from image observations. Due to the ill-posed nature of the problem, prior works often tackle a simplified task and ignore spatially-varying effects, using lighting representations such as spherical lobes [3,24], light probes [21], sky parameters [16,17,39], and environment maps [12,31,38,32]. Recent works explored various representations for capturing spatially-varying lighting, including per-pixel spherical lobes [13,23,41], light source parameters [11], per-location environment map [33,43] and 3D volumetric lighting [34,37]. These works usually train with synthetic data due to the scarcity of available groundtruth HDR light field on real-world captures.

Outdoor scenes require special attention for the extreme High Dynamic Range (HDR) lighting intensity, *e.g.* the sun's intensity, which is several orders higher in magnitude than typical light sources found in indoor scenes. Prior works employed sky parameters [17,39], or learned sky models [16] with an encoder-decoder architecture for modeling HDR sky. However, these works typically only focus on modeling the sky, and ignore the high-frequency and spatially-varying effects, which are equally important to get right for AR applications. In this work, we propose a unified representation that can handle both HDR sky intensity as well as spatially varying effects in outdoor scenes to achieve better performance.

Self-supervised lighting estimation methods apply differentiable rendering to provide gradient for lighting estimation [6,7,40,22,27]. Differentiable rasterization-based renderers [6,40,7] are typically limited to images of single objects and ignores the spatially-varying effects. Physically-based rendering (PBR) methods [22,27] require intensive memory and running time, and are thus limited to optimization tasks. Note that existing differentiable renderers [6,7,27] typically do not provide direct functionality for object insertion, which is an image editing task. We propose a novel differentiable object insertion formulation, providing valuable supervision signal for lighting estimation.

Image manipulation. Related to ours is also work that aims to insert synthetic objects into images using alternative techniques, such as adversarial methods [25,20], or by perturbing real scenes using recent advances in neural rendering [28]. Alhaija *et al.* [2] assumes known lighting and propose to use AR as a data generation technique by inserting synthetic assets into real world scenes. Naive copy-paste object insertion has also been shown to boost downstream object recognition accuracy [35,10]. Neural Scene Graph [28] optimize neural implicit functions for each object in the scene. Despite realistic editing results, lighting effects are baked into the representation and thus swapping assets from one scene to another is not easily possible. GeoSim [8] reconstructs assets such as cars from real-world driving sequences, and inserts them into a given image

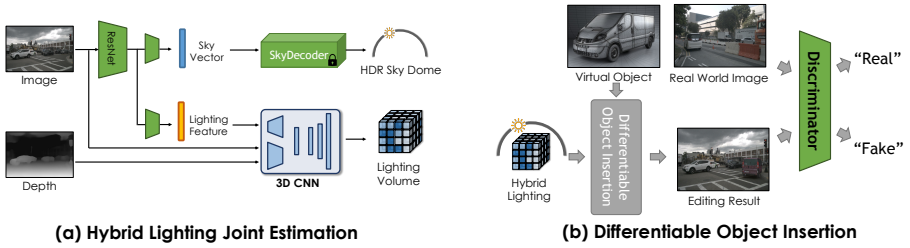

Fig. 2. Model overview. Our monocular lighting estimation model (a) predicts a hybrid lighting representation containing an HDR sky dome (top) representing sky and sun at infinity, and a lighting volume (bottom) representing the surrounding scene. The depth (a, left) for lighting volume prediction comes from off-the-shelf monocular depth estimator [14]. With the predicted lighting, our object insertion module (b) renders a 3D asset into a given image and is fully differentiable w.r.t. lighting parameters, thus enabling end-to-end training with adversarial objective for photorealism.

using a classical renderer followed by a shallow neural renderer that fixes errors such as unrealistic compositing effects. While achieving impressive results on car insertion, it remains difficult to apply on less frequent objects. In contrast, our method supports inserting 3D assets of various classes (Fig. 1).

3 Method

We aim to estimate scene lighting so as to render synthetic 3D assets into images realistically. In what follows, we introduce the hybrid representation, which comprises an HDR sky dome and a volumetric scene lighting (Sec. 3.1), the hybrid lighting prediction model (Sec. 3.2, Fig. 2a), the differentiable object insertion module that renders a virtual object into an image (Sec. 3.3, Fig. 2b), and the training schema (Sec. 3.4).

3.1 Hybrid Lighting Representation

Our goal is to model the 5D light field, which maps a spatial location $\mathbf{x} \in \mathbb{R}^3$ and light direction $\mathbf{l} \in \mathcal{S}^2$ into an HDR radiance value $\mathbf{r} \in \mathbb{R}^3_+$. In contrast to indoor, outdoor scenes require simultaneously modeling the extreme HDR sky as well as the surrounding environment. The peak magnitude of the former (sun) can be several orders higher than the latter. To address this, we propose to use a hybrid lighting representation that separately models the sky at infinity and the surrounding scene. This decomposition allows us to capture both, the extreme intensity of the sky while preserving the spatially-varying effects of the scene.

HDR sky representation. The sky dome typically contains a relatively simple structure, *i.e.* sun, sky, and possibly clouds, which affords a much lower dimensional representation than that of a typical environment map. Thus, instead of directly predicting a high resolution 2D environment map, we learn a feature space of the sky, and represent the sky dome with a sky feature vector $\mathbf{f} \in \mathbb{R}^d$,

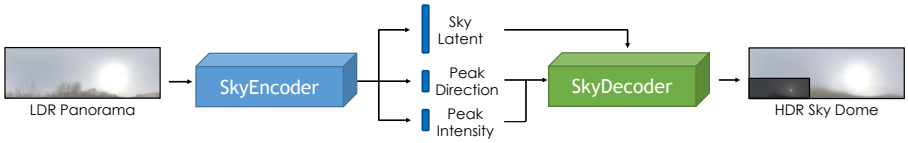

Fig. 3. Sky modeling network takes as input an LDR panorama and produces an HDR sky with an encoder-decoder structure, where the network also learns to compress sky information into the intermediate vector representation $\mathbf{f} \in \mathbb{R}^d$. The sky vector consists of explicit feature of peak intensity and direction, and a latent feature vector.

which can further be passed to a pretrained CNN decoder to decode into an HDR environment map, as shown in Fig. 2a (top). The sky feature space learning is described in Sec. 3.2, *sky modeling* part.

Spatially-varying surrounding scene representation. The outdoor scenes generally consist of complex geometric structures resulting in location-dependent lighting effects like shadows and reflections, which cannot be simply modeled as an environment map. To address this, we use a volumetric spherical Gaussian (VSG) [37] to represent the nearby surrounding scene. VSG is a 8-channel volumetric tensor $L_{\text{VSG}} \in \mathbb{R}^{8\times X\times Y\times Z}$, augmenting the RGB$\alpha$ volume with view-dependent spherical Gaussian lobes. Each voxel of VSG contains a spherical Gaussian lobe $G(\mathbf{l}) = \boldsymbol{c}e^{-(1-\mathbf{l}\cdot\boldsymbol{\mu})/\sigma^2}$, where $\mathbf{l}$ is viewing direction and $\boldsymbol{\xi} = \{\boldsymbol{c}, \boldsymbol{\mu}, \sigma\}$ are 7-dimensional parameters to model the exiting radiance of the corresponding 3D location. Each voxel also has an alpha channel $\alpha \in [0, 1]$ to represent occupancy. The 5D light field can be queried with volume rendering which we detail below.

Radiance query function. With our hybrid lighting representation utilizing both sky dome and volumetric lighting, the lighting intensity at any 3D point along any ray direction can be queried. To compute the radiance of a ray that starts inside the volume, we first shoot the ray through the lighting volume L_{VSG} and finally hit the HDR sky dome L_{env}. We adopt alpha compositing to combine the two lighting effects. Specifically, to compute the lighting intensity for the ray emitted from the location $\mathbf{x} \in \mathbb{R}^3$ in the direction $\mathbf{l} \in \mathcal{S}^2$, we select K equi-spaced locations along the ray and use nearest neighbor interpolation to get the voxel values $\{\alpha_k, \boldsymbol{\xi}_k\}_{k=1}^K$ from L_{VSG}. We then query the intensity of L_{env} in the direction $\mathbf{l}$, referred to as $L_{\text{env}}(\mathbf{l})$, via bilinear interpolation. The final HDR light intensity $L(\mathbf{x}, \mathbf{l}) \in \mathbb{R}^3_+$ can be computed using volume rendering:

$$L(\mathbf{x}, \mathbf{l}) = \Big(\sum_{k=1}^{K} \tau_{k-1}\alpha_k G(-\mathbf{l}; \boldsymbol{\xi}_k)\Big) + \tau_K L_{\text{env}}(\mathbf{l}) \tag{1}$$

where $\tau_k = \prod_{i=1}^{k}(1-\alpha_i)$ is the transmittance. This function will be used for rendering object insertion in Eq. 2 and shadows in Eq. 4.

3.2 Network Architecture

In this section, we introduce the network architecture for the pre-trained sky model and lighting prediction.

Sky modeling. To learn the sky feature space, we design a sky modeling network as shown in Fig. 3. Specifically, the encoder compresses the input LDR panorama into a feature vector, then the decoder decode it to the original HDR sky dome. Our sky feature vector $\mathbf{f}$ contains explicit HDR peak intensity $\mathbf{f}_{\text{intensity}} \in \mathbb{R}^3$, peak direction $\mathbf{f}_{\text{dir}} \in \mathbb{R}^3$, and a latent vector $\mathbf{f}_{\text{latent}} \in \mathbb{R}^{d-6}$ encoding the content of the sky dome. This design allows us to control the lighting of the sun by controlling $\mathbf{f}_{\text{dir}}$ and $\mathbf{f}_{\text{intensity}}$, which is convenient for applications that allow manual editing.

We pre-train the sky encoder-decoder network on a set of outdoor HDR panoramas and keep it freezed. Once trained, we integrate the fixed decoder into the sky prediction branch (Fig. 2a top). The pre-trained sky modeling network, which maps an LDR panorama into HDR, will also be used to generate HDR pseudo labels for supervision (Sec. 3.4) due to the lack of HDR data.

Hybrid lighting prediction. As shown in Fig. 2, we use a two-branch network to predict the sky dome and the lighting volume. We detail each branch below.

HDR sky prediction branch. Given an input image, the sky branch directly predicts the sky feature vector $\mathbf{f}$ from the ResNet [15] backbone. The pre-trained fixed sky decoder then maps $\mathbf{f}$ to an HDR sky environment map.

Lighting volume prediction branch. We adapt from a subnetwork of [37] for VSG prediction. Specifically, we first use an MLP [26] to map the lighting feature extracted by ResNet backbone into a feature volume, and then unproject the input image into a RGBα volume. We adopt a 3D UNet to fuse the two volumes and predict the VSG lighting representation. Since unprojection requires depth information, we adopt an off-the-shelf monocular depth estimator PackNet [14] to predict a dense depth map. Architecture details are included in the Appendix.

3.3 Differentiable Object Insertion

We now study the task of realistic object insertion, which is key in AR and our main final objective. Given an estimated lighting representation L, our goal is to composite a virtual object with known geometry $\mathcal{M}$ and material Θ into a real image I with known depth D and camera intrinsics. To achieve realistic lighting effects in this process, not only the inserted object should be influenced by the scene lighting, but it should also affect the scene to create cast shadows.

In our work, we aim to make this insertion module differentiable, including shadow rendering, such that it can afford gradient backpropagation from the losses defined on the composited image, back to our lighting parameters. Since groundtruth light field is not easily available for outdoor scenes, we argue that a plausible complementary supervision signal is to use the quality of object insertion as an objective. Thus, carefully designed adversarial training on the composite images can be a powerful approach to supervise lighting estimation. We also argue that even if groundtruth lighting information would be available, optimizing for the quality of object insertion end-to-end will likely lead to improved results.

Foreground object appearance rendering. We render the virtual object with our predicted hybrid lighting representation L using a physically-based renderer. We adopt the Disney BRDF [4,19,7] for enhanced realism.

Specifically, we first shoot rays from the camera origin to the scene, where we apply ray-mesh intersection detection for the rays and the inserted object $\mathcal{M}$. For each intersected ray, we create a G-buffer for the location of the intersection $\mathbf{x}$, the surface normal $\mathbf{n}$ and the material properties θ. We bounce multiple rays at $\mathbf{x}$ and render with Monte-Carlo numerical integration:

$$I_{\mathbf{x}} = \frac{1}{N}\sum_{k=1}^{N}\frac{f(\mathbf{l}_k, \mathbf{v}; \theta)L(\mathbf{x}, \mathbf{l}_k)(\mathbf{n}\cdot\mathbf{l}_k)^+}{p(\mathbf{l}_k)} \tag{2}$$

where $L(\cdot,\cdot)$ is the radiance query function as defined in Eq. 1, N and $\mathbf{l}_k$ is the number and direction of sampled lighting, $\mathbf{v}$ is the viewing direction of the camera ray, and f is the Disney BRDF.

Background shadow map rendering. The inserted object changes the light transport in the scene and affects the appearance of the background scene pixels, which typically causes shadows. We adopt ray tracing to generate faithful ratio shadow maps for the inserted object, inspired by classic ratio imaging techniques [29]. Specifically, for each scene pixel p, we compute its 3D location $\mathbf{x}$ from the depth map D. We first compute the lighting distribution before object insertion $\{L(\mathbf{x}_\mathrm{s}, \mathbf{l}_k)\}_{k=1}^{N_\mathrm{s}}$, where $\{\mathbf{l}_k\}_{k=1}^{N_\mathrm{s}}$ are uniformly selected light directions on the upper hemisphere. After object insertion, the rays may potentially get occluded by the inserted objects, resulting in a post-insertion lighting distribution $\{L^{'}(\mathbf{x}_\mathrm{s}, \mathbf{l}_k)\}_{k=1}^{N_\mathrm{s}}$. To compute $\{L^{'}(\mathbf{x}_\mathrm{s}, \mathbf{l}_k)\}_{k=1}^{N_\mathrm{s}}$, we perform ray-mesh query for all rays. If ray $(\mathbf{x}_\mathrm{s}, \mathbf{l}_k)$ is occluded by the inserted object, we set the lighting intensity value to an ambient value I_a which we empirically set to 0.1, while the lighting intensities of unoccluded rays remain the same as the original radiance. We then define the shadow effects as the ratio of the pixel intensity values before (I) and after ($I^{'}$) object insertion,

$$S_p = \frac{I_p^{'}}{I_p} = \frac{\sum_{k=1}^{N_\mathrm{s}} f_\mathrm{scene}(\mathbf{x}_\mathrm{s}, \mathbf{l}_k, \mathbf{v}; \theta)L^{'}(\mathbf{x}_\mathrm{s}, \mathbf{l}_k)(\mathbf{n}\cdot\mathbf{l}_k)^+}{\sum_{k=1}^{N_\mathrm{s}} f_\mathrm{scene}(\mathbf{x}_\mathrm{s}, \mathbf{l}_k, \mathbf{v}; \theta)L(\mathbf{x}_\mathrm{s}, \mathbf{l}_k)(\mathbf{n}\cdot\mathbf{l}_k)^+} \tag{3}$$

where the BRDF of the scene pixel f_scene and normal direction $\mathbf{n}$ are unknown. As our object insertion and shadows occur on flat surfaces in typical street scenes we consider, we simplify it by assuming the normal direction is pointing upward, and assume the scene surface is Lambertian with constant diffuse albedo $f_\mathrm{scene}(\mathbf{x}_\mathrm{s}, \mathbf{l}_k, \mathbf{v}) = f_\mathrm{d}$. As a result, we can move the BRDF term outside the sum in Eq. 3 and cancel it out to obtain a simpler term:

$$S_p = \frac{f_\mathrm{d}\sum_{k=1}^{N_\mathrm{s}} L^{'}(\mathbf{x}_\mathrm{s}, \mathbf{l}_k)(\mathbf{n}\cdot\mathbf{l}_k)^+}{f_\mathrm{d}\sum_{k=1}^{N_\mathrm{s}} L(\mathbf{x}_\mathrm{s}, \mathbf{l}_k)(\mathbf{n}\cdot\mathbf{l}_k)^+} = \frac{\sum_{k=1}^{N_\mathrm{s}} L^{'}(\mathbf{x}_\mathrm{s}, \mathbf{l}_k)(\mathbf{n}\cdot\mathbf{l}_k)^+}{\sum_{k=1}^{N_\mathrm{s}} L(\mathbf{x}_\mathrm{s}, \mathbf{l}_k)(\mathbf{n}\cdot\mathbf{l}_k)^+} \tag{4}$$

which can be computed with the estimated lighting L. Scene pixels after insertion can then be computed by multiplying the ratio shadow map $I^{'} = S \odot I$.

Gradient propagation. We design the forward rendering process to be differentiable for both foreground object and background shadows, which allows us to back propagate gradients from image pixels to the lighting parameters.

For each foreground pixel, the rendered appearance of the inserted object is differentiable via Eq. 2. Gradients from background pixels $I^{'}$ with respect to the lighting L, *i.e.* $\frac{\partial I^{'}}{\partial L}$, can be computed via $\frac{\partial I^{'}}{\partial L} = \frac{\partial S}{\partial L} I$, where the shadow ratio S in Eq. 4 is also differentiable wrt. lighting L. Intuitively, if we want the shadows around the object to be perceptually darker, this will encourage the occluded light directions to have stronger intensity.

3.4 Training

We first pre-train the sky modeling network on a collection of outdoor HDR panoramas, and then keep it fixed in the following training process for our hybrid lighting prediction. The supervision for our hybrid lighting joint estimation module comes from two parts: (1) the direction supervision that learns lighting information from the training data, and (2) the adversarial supervision that applies on the final editing results and optimizes for realism.

Sky Modeling Supervision. We train the sky modeling encoder-decoder network (Fig. 3) on a collection of outdoor HDR panoramas. For each HDR panorama I_{HDR}, we compute its ground-truth peak intensity $\mathbf{f}_{\text{intensity}}$ and direction $\mathbf{f}_{\text{dir}}$. We train the encoder-decoder with the LDR-HDR pair $(I_{\text{LDR}}, I_{\text{HDR}})$, where the input LDR panorama I_{LDR} is converted from the HDR panorama via gamma correction and intensity clipping. We supervise the network with a combination of three losses, including peak direction loss $\mathcal{L}_{\text{dir}}$ with L1 angular error, and peak intensity loss $\mathcal{L}_{\text{intensity}}$ and HDR reconstruction loss $\mathcal{L}_{\text{hdr}}$ using log-encoded L2 error defined as $\text{LogEncodedL2}(\hat{x}, x) = ||\log(1+\hat{x}) - \log(1+x)||_2^2$.

Supervision for Hybrid Lighting Prediction. To supervise lighting prediction, we use two complementary datasets: the self-driving dataset nuScenes [5], and the panoramic street view dataset HoliCity [42]. As both datasets are LDR, we predict HDR pseudo labels from the pre-trained modeling network (Fig. 3) by lifting HoliCity LDR panoramas into HDR. In what follows, we descibe the supervision for the sky prediction branch, the lighting volume prediction branch, and the loss signal to combine the two representation.

HDR sky branch losses. We train our sky branch on HoliCity [42], which contains LDR panoramas I_{pano} with sky masks M_{sky} and sun location $\mathbf{f}_{\text{dir}}$. To compute HDR pseudo labels, we feed the LDR panorama I_{pano} into the pre-trained sky network, and get the estimated sky peak intensity $\tilde{\mathbf{f}}_{\text{intensity}}$ and latent code $\tilde{\mathbf{f}}_{\text{latent}}$ as pseudo groundtruth to supervise our sky prediction branch. In a training step, we crop a perspective image I_{crop} from the panorama as input to our lighting estimation network, and predict the sky vector output $(\hat{\mathbf{f}}_{\text{intensity}}, \hat{\mathbf{f}}_{\text{dir}}, \hat{\mathbf{f}}_{\text{latent}})$ and the reconstructed HDR sky image $\hat{I}_{\text{pano}}$. We use a combination of the log-encoded L2 loss for peak intensity $(\hat{\mathbf{f}}_{\text{intensity}}, \tilde{\mathbf{f}}_{\text{intensity}})$, L1 loss for latent code $(\hat{\mathbf{f}}_{\text{latent}}, \tilde{\mathbf{f}}_{\text{latent}})$, L1 angular loss for peak direction $(\hat{\mathbf{f}}_{\text{dir}}, \mathbf{f}_{\text{dir}})$, and L1 reconstruction loss between $(\hat{I}_{\text{pano}} \odot M_{\text{sky}}, I_{\text{pano}} \odot M_{\text{sky}})$ within the LDR sky region indicated by M_{sky}.

Lighting volume branch loss. Recall that images in the datasets are inherently groundtruth of a subset of the light field captured by camera sensor rays, *e.g.* the videos captured by self-driving cars in nuScenes [5] and the panoramas in HoliCity [42]. Meanwhile, the predicted hybrid lighting representation supports radiance query along arbitrary rays as shown in Eq. 1. Thus, we can enforce the consistency of radiance between our predicted light field and captured image groundtruth, given known camera pose and intrinsics.

Specifically, we sample images from nuScenes, and crop perspective images from HoliCity panoramas as input images. We then predict the corresponding lighting volume together with the sky dome, query the radiance of the camera rays, and enforce consistency with ground truth captured images using L2 loss. Following [37], we also render the alpha channel into a depth map and enforce consistency with groundtruth depth.

Sky separation loss. Intuitively, the real world camera rays that directly reach the sky should also transmit through the lighting volume and hit the sky environment map. With the losses mentioned above, the model may still fall into the degenerate case where the lighting volume completely occludes the sky. To address this, we use the sky mask M_{sky} information to supervise the sky transmittance τ_K in Eq. 1 with binary cross entropy loss.

Training Lighting via Object Insertion. Our final goal is to realistically insert virtual objects into images. We formulate the object insertion process in an end-to-end fashion and use a discriminator to supervise the perceptual lighting effects on the image editing results.

Specifically, we collect a set of high quality 3D car models from Turbosquid[1]. Given an input image, we estimate scene lighting, randomly select a 3D asset, and insert it into the scene using our object insertion module to get $\hat{I}_{\text{edit}}$. We use the map information available in nuScenes to place the car on a driveable surface. We also perform collision and occlusion checking with the depth map to avoid unrealistic object insertion due to erroneous placement. As shown in Fig. 2b, we use a discriminator to judge the quality of $\hat{I}_{\text{edit}}$ compared to real cars, and employ adversarial supervision to optimize for realism of insertion: $\mathcal{L}_{\text{adv}} = -\mathcal{D}(\hat{I}_{\text{edit}})$. Intuitively, a discriminator could easily detect erroneous shadow direction and intensity, and error in specular highlights. Through the adversarial supervision, the estimated lighting is encouraged to produce object insertion results similar to real world image samples. We refer to further analysis in the Appendix.

4 Experiments

We extensively evaluate our method both qualitatively and quantitatively. We first provide experiment details (Sec. 4.1). We then compare lighting estimation, evaluate the quality of object insertion (Sec. 4.2) and perform ablation study (Sec. 4.3). Finally, we show that our AR data helps downstream self-driving perception tasks (Sec. 4.4).

[1] www.turbosquid.com

Method	Median angular error ↓
Hold-Geoffroy *et al.* [16]	24.88°
Wang *et al.* [37]	53.86°
Ours	**22.43°**
Ours (w/o sky modeling)	31.45°
Ours (w/o adv. supervision)	24.16°

Table 1. Quantitative results of peak direction on HoliCity [42]. We outperform past work, and each component (sky modeling, adversarial supervision) helps.

Method	PSNR ↑	si-PSNR ↑
Hold-Geoffroy *et al.* [16]	9.33	10.73
Hold-Geoffroy *et al.* [16]*	10.81	14.20
Wang *et al.* [37]	14.06	15.28
Ours (w/o adv. supervision)	14.23	15.31
Ours	**14.49**	**15.35**

Table 2. Quantitative results of LDR appearance on the nuScenes dataset [5]. * indicates constraining the evaluation on the upper hemisphere.

4.1 Experimental Details

Lighting estimation. Our lighting representation combines a sky feature vector and a VSG lighting volume. We set the dimension of the sky vector to be 64 and decode it to a 64x256 HDR sky dome. Different from [37], we tailor the size of VSG lighting volume to be 256x256x64 (xyz) to accommodate 300x300x80 (meters^3) outdoor scenes. As outdoor scenes are larger in scale while visible scene surfaces are relatively dense in close-to-camera regions, we employ log projection to map the volume representation to a 3D scene location. Inference time of the lighting estimation network is 180ms per image, clocked on a TITAN V GPU.
Object insertion. Our object insertion module relies on a differentiable rendering process of both foreground object and background shadows. During training, we sample 5000 rays for foreground objects. For background shadows, we render a 160x90 resolution shadow map and sample 450 rays per pixel to save memory and computation. After training, we do importance sampling for each pixel in foreground and can afford a high resolution shadow map for background to generate more realistic effects. During inference time, we also have the option to use commercial renderer such as Blender [9], which we detail in the Appendix.
Datasets. We collected 724 outdoor HDR panoramas from online HDRI databases to train the sky encoder-decoder. We train the full model with nuScenes [5] and HoliCity [42]. For nuScenes, we use the official split containing 700 scenes for training and 150 scenes for evaluation. For HoliCity dataset, we apply 90% v.s. 10% data split for training and evaluation.
Multi-view extension. While we focus on monocular estimation, our model is extendable to multi-view input. It can consume multi-view images to predict more accurate lighting, as shown in Fig. 6. For the HDR sky prediction branch, we apply max pooling for $\mathbf{f}_{\text{intensity}}$, $\mathbf{f}_{\text{latent}}$, and average pooling to $\mathbf{f}_{\text{dir}}$ after rotating $\mathbf{f}_{\text{dir}}$ in different views to the canonical view. As for the volumetric lighting, since it is defined in the "world" coordinate space, we unproject and fuse multi-view images into a common lighting volume representation, akin to [30].

4.2 Evaluation of Lighting Estimation

Baselines. We compare with current state-of-the-art lighting estimation methods [16,37]. Hold-Geoffroy *et al.* [16] estimates the HDR sky environment map from

Approach	% Ours (w/o adv. sup.) is preferred ↓
Hold-Geoffroy *et al.* [16]	68.1 ± 5.4 %
Wang *et al.* [37]	94.2 ± 2.0 %
Ours	$\mathbf{40.6 \pm 10.2}$ %

Table 3. Quantitative results of user study. Users compare baseline methods to an ablated version of our method (Ours w/o adv. supervision) in a pair-wise comparison. Each row reports the percentage of images that Ours w/o adv. supervision is preferred. Our method outperforms baselines, and adv. supervision improves performance.

Fig. 4. Qualitative comparison of lighting estimation. We insert a purely specular sphere into the image to visualize the lighting prediction, and display the environment maps on the bottom. Note the sun and environment map changes between locations.

a single image. Wang *et al.* [37] predicts Volumetric Spherical Gaussian. We re-train or finetune these methods on the same data sources we used for our method to ensure a fair comparison.

HDR evaluation of peak direction. We evaluate peak direction prediction on HoliCity dataset [42]. We report the median angular error between the predicted direction and GT in Tbl. 1. Wang *et al.* [37] predicts HDR component in a self-supervised manner and cannot learn strong peaks. We also outperform Hold-Geoffroy *et al.* [16], which separately predicts a sky dome and its azimuth.

LDR evaluation of novel view reconstruction. Recall that any lighting representation, such as environment map and our hybrid lighting, aims to represent the complete or a subset of the light field, which can be rendered into images by querying the radiance function with specified camera rays. Prior work [34,37] proposed to use novel view radiance reconstruction PSNR as quantitative evaluation of the quality of lighting estimation, which we report on nuScenes dataset [5].

As the nuScenes-captured images may have different exposure values, we report both PSNR and scale invariant PSNR (si-PSNR) in Tbl. 2. For the latter, we multiply the predicted novel view with a scaling factor that minimizes L2 error. Since Hold-Geoffroy *et al.* [16] only predicts lighting on the upper hemisphere, we

Fig. 5. Qualitative comparison of virtual object insertion. Our method produces realistic cast shadows and high-frequency "clear coat" effects.

Fig. 6. Qualitative results of spatially-varying shadow effects. Our method can handle the spatial changes of shadow intensity around shadow boundary, not possible previously. (Results take six surrounding perspective views as input.)

also constrain the evaluation on the upper hemisphere to make a fair comparison. Our method outperforms both baselines with a large margin, as [16] ignores spatially-varying effects and usually predicts a sky dome with little high-frequency details. Our method also outperforms Wang *et al.* [37] which cannot handle high HDR intensity of an outdoor scene.

Human study. To quantitatively evaluate the quality of object insertion, we perform a human study with Amazon Mechanical Turk, where we show two augmented images, randomly permuted, produced by our method and by the baseline. We then ask users to compare the realism of the inserted object, e.g. the cast shadows and the reflections, and select the more realistic image. For each comparison, we invite 15 users to judge 23 examples. We adopt majority vote for the preference of each example, and run three times to report mean and standard deviation in Tbl. 3. The ablated version of Ours (w/o adv. supervision) outperforms baselines, indicating the hybrid lighting representation improves upon prior works. Comparing Ours and Ours (w/o adv. supervision), the results indicate including adversarial supervision leads to more visually realistic editing.

Qualitative comparison. We first visualize the environment maps at different scene locations and insertion of a purely specular sphere in Fig. 4. Hold-Geoffroy *et al.* [16] only predicts one environment lighting and ignored spatially-varying effects. For inserted spheres around shadowed region, it still produces strong cast shadows. Also, the high-frequency details are not well preserved in the sky prediction. Wang *et al.* [37] can generate high-frequency details but fails to handle the extreme HDR intensity of the outdoor scene, and thus cannot generate

Fig. 7. Discriminator test-time optimization. Note the discriminator corrects the shadow direction (left), and removes erroneous specular highlight (right). The pixelated effect is due to low-resolution rendering during end-to-end training.

realistic cast shadows. Our method is the only one that handles extreme HDR, angular details and spatially-varying lighting. We show virtual object insertion results in Fig. 5. Our lighting prediction preserves high-frequency details with HDR intensity, producing realistic highlights and clear coating effects, while prior method cannot generate such effects.

Spatially-varying shadows. Benefiting from the accurate HDR sky and surrounding scene estimation, our method can produce spatially-varying shadow effects. As shown in Fig. 6, the car casts an intense shadow when outside the shadow of the building, while the shadow caused by the car is much weaker when the car is inside the shadow region. Especially, it also shows reasonably different shadow intensity around the shadow boundary. This challenging effect requires accurate prediction of direction, intensity and geometry of HDR lighting, and will not occur with an incapable lighting representation.

4.3 Ablation Study

We verify the effectiveness of the sky decoder module and the adversarial supervision. In Tbl. 1, compared to the ablated version that directly predict sky dome as an environment map (denoted as "Ours w/o sky modeling"), the full model reduces the error by around 50%, which demonstrates the sky modeling network is important for achieving accurate sun position prediction in outdoor scenes.

Adversarial supervision improves the performance on quantitative evaluation in Tbl. 1, 2, especially for peak direction estimation, which indicates that discriminating on the final image editing result is complementary to existing supervision and benefits lighting prediction. In the user study (Tbl. 3), adversarial supervision improves perceptual realism and receives a higher user preference. We also qualitatively visualize the behaviour of the discriminator in Fig. 7. To understand the "photorealism" implicitly perceived by the discriminator during the training process, we perform test-time optimization on the object insertion results to minimize the adversarial loss, and show the optimized results in Fig. 7. In the first example, the initial editing results fail to predict the correct sun location and produce wrong shadows. After test-time optimization, the shadow direction points to the bottom-left of the image and the shadow intensity also matches the visual cues from the rest of the scene. In the second example, the initial editing results contain an obviously erroneous highlight, and the discriminator detects the artifact and removes it. This agrees with the intuition that a neural

Method	mAP	car	bus	trailer	const.vehicle	bicycle
Real Data	0.190	0.356	0.124	0.011	0.016	0.116
+ Aug No Light.	0.201	0.363	0.163	0.029	<u>0.021</u>	0.120
+ Aug Light.	**0.211**	<u>0.369</u>	<u>0.182</u>	<u>0.036</u>	0.020	<u>0.146</u>

Table 4. Performance of a SOTA 3D object detector [36] on nuScenes benchmark. mAP represents the mean for the 10 object categories. We report individual categories that saw a significant boost (full table in the Appendix).

discriminator have the capacity to catch lighting effects such as cast shadows and incorrect specular highlights. Further details are included in the Appendix.

4.4 Downstream Perception Task

We investigate the benefits of our object insertion as data augmentation for a downstream 3D object detection task on nuScenes. The goal of this task is to place a 3D bounding box for 10 different object categories. We first train a state-of-the-art monocular 3D detector [36] on a 10% subset of real data from the nuScenes training set. This subset was chosen randomly across all scenes but in a way that the number of objects per category resembles the original nuScenes training set. We then augment the front-camera images of this subset with our method. Specifically, we collect a set of 3D models with categories of *car* and *construction vehicles*, and randomly insert one object per image. Our augmented dataset has approximately 15K augmented (new) images. We use the same training strategy and model hyperparameters as [36] but do not supervise attributes or velocity as these are not present for the augmented data. Quantitatively, in Tbl. 4 we can observe that the performance of the detector improves by 2% when comparing to real data. Moreover, we can also see that while naively adding objects leads to a 1% improvement, another 1% is a result of having better light estimation. Interestingly, we can also notice that the performance of the object detector also improves in different categories even though we do not directly augment those.

5 Discussion

In this paper, we proposed a hybrid representation of lighting and a novel differentiable object insertion module that allows end-to-end optimization of AR objectives. In a variety of comparisons, we demonstrate the effectiveness of our approach in lighting estimation. Furthermore, we showcase performance gains on a 3D object detection task when training the detector on our augmented dataset.

While our method presents an effective way of rendering 3D assets into images, some limitations remain for future work. Currently, the inserted virtual object pixels do not pass the same capturing process as the background scene, and the shadow rendering assumes Lambertian surface. Modeling of camera ISP, weather, and non-Lambertian scene materials can be interesting directions for future work.

References

1. Adelson, E.H., Bergen, J.R.: The plenoptic function and the elements of early vision. In: Computational Models of Visual Processing. pp. 3–20. MIT Press (1991) 1
2. Alhaija, H.A., Mustikovela, S.K., Mescheder, L., Geiger, A., Rother, C.: Augmented reality meets computer vision: Efficient data generation for urban driving scenes. International Journal of Computer Vision **126**(9), 961–972 (2018) 3
3. Boss, M., Jampani, V., Kim, K., Lensch, H., Kautz, J.: Two-shot spatially-varying brdf and shape estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3982–3991 (2020) 3
4. Burley, B., Studios, W.D.A.: Physically-based shading at disney. In: ACM SIGGRAPH. vol. 2012, pp. 1–7. vol. 2012 (2012) 6
5. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. arXiv preprint arXiv:1903.11027 (2019) 8, 9, 10, 11
6. Chen, W., Ling, H., Gao, J., Smith, E., Lehtinen, J., Jacobson, A., Fidler, S.: Learning to predict 3d objects with an interpolation-based differentiable renderer. In: NeurIPS (2019) 3
7. Chen, W., Litalien, J., Gao, J., Wang, Z., Tsang, C.F., Khalis, S., Litany, O., Fidler, S.: DIB-R++: Learning to predict lighting and material with a hybrid differentiable renderer. In: Advances in Neural Information Processing Systems (NeurIPS) (2021) 3, 6
8. Chen, Y., Rong, F., Duggal, S., Wang, S., Yan, X., Manivasagam, S., Xue, S., Yumer, E., Urtasun, R.: Geosim: Realistic video simulation via geometry-aware composition for self-driving. In: CVPR (2021) 1, 3
9. Community, B.O.: Blender - a 3D modelling and rendering package. Blender Foundation, Stichting Blender Foundation, Amsterdam (2018), http://www.blender.org 10
10. Dwibedi, D., Misra, I., Hebert, M.: Cut, paste and learn: Surprisingly easy synthesis for instance detection. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017) 1, 3
11. Gardner, M.A., Hold-Geoffroy, Y., Sunkavalli, K., Gagné, C., Lalonde, J.F.: Deep parametric indoor lighting estimation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 7175–7183 (2019) 3
12. Gardner, M.A., Sunkavalli, K., Yumer, E., Shen, X., Gambaretto, E., Gagné, C., Lalonde, J.F.: Learning to predict indoor illumination from a single image. arXiv preprint arXiv:1704.00090 (2017) 3
13. Garon, M., Sunkavalli, K., Hadap, S., Carr, N., Lalonde, J.F.: Fast spatially-varying indoor lighting estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6908–6917 (2019) 3
14. Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., Gaidon, A.: 3d packing for self-supervised monocular depth estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 4, 6
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015), http://arxiv.org/abs/1512.03385 6
16. Hold-Geoffroy, Y., Athawale, A., Lalonde, J.F.: Deep sky modeling for single image outdoor lighting estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6927–6935 (2019) 2, 3, 10, 11, 12
17. Hold-Geoffroy, Y., Sunkavalli, K., Hadap, S., Gambaretto, E., Lalonde, J.F.: Deep outdoor illumination estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7312–7321 (2017) 2, 3

18. Hong, S., Yan, X., Huang, T.E., Lee, H.: Learning hierarchical semantic image manipulation through structured representations. In: Advances in Neural Information Processing Systems. pp. 2713–2723 (2018) 1
19. Karis, B., Games, E.: Real shading in unreal engine 4. Proc. Physically Based Shading Theory Practice **4**(3) (2013) 6
20. Kim, S.W., Philion, J., Torralba, A., Fidler, S.: DriveGAN: Towards a Controllable High-Quality Neural Simulation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 3
21. LeGendre, C., Ma, W.C., Fyffe, G., Flynn, J., Charbonnel, L., Busch, J., Debevec, P.: Deeplight: Learning illumination for unconstrained mobile mixed reality. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5918–5928 (2019) 3
22. Li, T.M., Aittala, M., Durand, F., Lehtinen, J.: Differentiable monte carlo ray tracing through edge sampling. ACM Trans. Graph. (Proc. SIGGRAPH Asia) **37**(6), 222:1–222:11 (2018) 3
23. Li, Z., Shafiei, M., Ramamoorthi, R., Sunkavalli, K., Chandraker, M.: Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2475–2484 (2020) 3
24. Li, Z., Xu, Z., Ramamoorthi, R., Sunkavalli, K., Chandraker, M.: Learning to reconstruct shape and spatially-varying reflectance from a single image. ACM Transactions on Graphics (TOG) **37**(6), 1–11 (2018) 3
25. Ling, H., Acuna, D., Kreis, K., Kim, S.W., Fidler, S.: Variational amodal object completion. Advances in Neural Information Processing Systems (2020) 3
26. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4460–4470 (2019) 6
27. Nimier-David, M., Vicini, D., Zeltner, T., Jakob, W.: Mitsuba 2: A retargetable forward and inverse renderer. Transactions on Graphics (Proceedings of SIGGRAPH Asia) **38**(6) (Dec 2019). https://doi.org/10.1145/3355089.3356498 3
28. Ost, J., Mannan, F., Thuerey, N., Knodt, J., Heide, F.: Neural scene graphs for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2856–2865 (June 2021) 1, 3
29. Peers, P., Tamura, N., Matusik, W., Debevec, P.: Post-production facial performance relighting using reflectance transfer. ACM Transactions on Graphics (TOG) **26**(3), 52–es (2007) 7
30. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. arXiv preprint arXiv:2008.05711 (2020) 10
31. Sengupta, S., Gu, J., Kim, K., Liu, G., Jacobs, D.W., Kautz, J.: Neural inverse rendering of an indoor scene from a single image. In: International Conference on Computer Vision (ICCV) (2019) 3
32. Somanath, G., Kurz, D.: Hdr environment map estimation for real-time augmented reality. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021) 3
33. Song, S., Funkhouser, T.: Neural illumination: Lighting prediction for indoor environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6918–6926 (2019) 3
34. Srinivasan, P.P., Mildenhall, B., Tancik, M., Barron, J.T., Tucker, R., Snavely, N.: Lighthouse: Predicting lighting volumes for spatially-coherent illumination. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8080–8089 (2020) 2, 3, 11

35. Su, H., Qi, C.R., Li, Y., Guibas, L.J.: Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In: The IEEE International Conference on Computer Vision (ICCV) (December 2015) 1, 3
36. Wang, T., Zhu, X., Pang, J., Lin, D.: Fcos3d: Fully convolutional one-stage monocular 3d object detection. arXiv preprint arXiv:2104.10956 (2021) 14
37. Wang, Z., Philion, J., Fidler, S., Kautz, J.: Learning indoor inverse rendering with 3d spatially-varying lighting. In: Proceedings of International Conference on Computer Vision (ICCV) (2021) 2, 3, 5, 6, 9, 10, 11, 12
38. Wei, X., Chen, G., Dong, Y., Lin, S., Tong, X.: Object-based illumination estimation with rendering-aware neural networks. arXiv preprint arXiv:2008.02514 (2020) 3
39. Zhang, J., Sunkavalli, K., Hold-Geoffroy, Y., Hadap, S., Eisenman, J., Lalonde, J.F.: All-weather deep outdoor lighting estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10158–10166 (2019) 2, 3
40. Zhang, Y., Chen, W., Ling, H., Gao, J., Zhang, Y., Torralba, A., Fidler, S.: Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. In: International Conference on Learning Representations (2021) 3
41. Zhao, Y., Guo, T.: Pointar: Efficient lighting estimation for mobile augmented reality. arXiv preprint arXiv:2004.00006 (2020) 3
42. Zhou, Y., Huang, J., Dai, X., Luo, L., Chen, Z., Ma, Y.: HoliCity: A city-scale data platform for learning holistic 3D structures (2020), arXiv:2008.03286 [cs.CV] 8, 9, 10, 11
43. Zhu, Y., Zhang, Y., Li, S., Shi, B.: Spatially-varying outdoor lighting estimation from intrinsics. In: CVPR (2021) 3