Photorealistic Object Insertion with Diffusion-Guided Inverse Rendering

Ruofan Liang[1,2,3], Zan Gojcic[1], Merlin Nimier-David[1], David Acuna[1],
Nandita Vijaykumar[2,3], Sanja Fidler[1,2,3], and Zian Wang[1,2,3]

[1]NVIDIA   [2]University of Toronto   [3]Vector Institute

Abstract. The correct insertion of virtual objects in images of real-world scenes requires a deep understanding of the scene's lighting, geometry and materials, as well as the image formation process. While recent large-scale diffusion models have shown strong generative and inpainting capabilities, we find that current models do not sufficiently "understand" the scene shown in a single picture to generate consistent lighting effects (shadows, bright reflections, etc.) while preserving the identity and details of the composited object. We propose using a personalized large diffusion model as *guidance* to a physically based inverse rendering process. Our method recovers scene lighting and tone-mapping parameters, allowing the photorealistic composition of arbitrary virtual objects in single frames or videos of indoor or outdoor scenes. Our physically based pipeline further enables automatic materials and tone-mapping refinement.

Keywords: Inverse rendering · Diffusion models · Personalization · Virtual object insertion · Physically based rendering

1 Introduction

Virtual object insertion enables a range of applications from virtual production, to interactive gaming and synthetic data generation. To produce photorealistic insertions, the interactions between the virtual objects and the environment need to be modeled faithfully, such as accurate specular highlights and shadows.

A standard virtual object insertion pipeline typically includes three key steps: i) lighting estimation from the input image, ii) 3D proxy geometry creation, and iii) composited image rendering in a rendering engine. However, the first and arguably most important step is still an open research question. Lighting estimation is particularly challenging when dealing with limited inputs such as a *single* image, captured using a consumer device with a low dynamic range. Indeed, inverse rendering is a fundamentally ill-posed problem.

To constrain its solution space, prior works either aimed to define hand-crafted priors [9,18,31,75] or to learn them from the data [15–17,22,23,32,36,56,62,63,77]. However, the former often fall short when applied to real-world scenes, while the latter suffer from scarcity of the ground truth data. As a result, these algorithms are often heavily tailored to a specific domain, e.g. indoor [15–17,50,63] or outdoor scenes [22,23,56,62,77].

Virtual Object Insertion   Applications

A vase with dried flowers   *A black SUV car*   Insert multiple objects

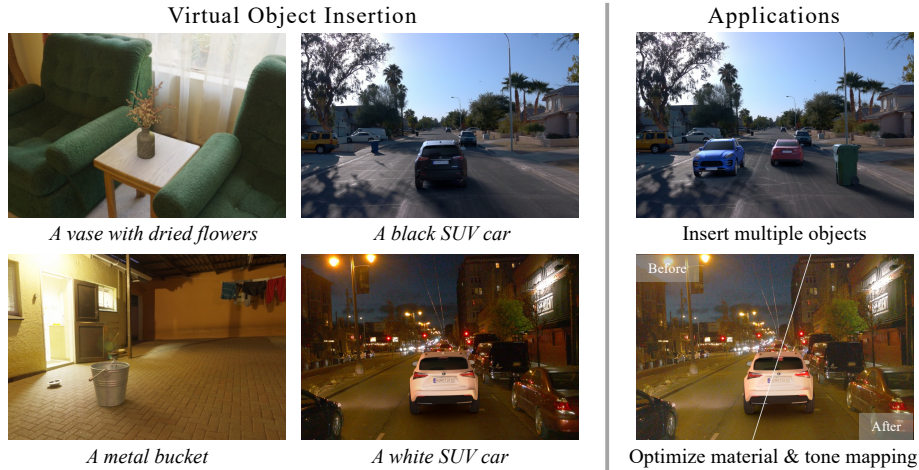*A metal bucket*   *A white SUV car*   Optimize material & tone mapping

Fig. 1: We propose DiPIR, a physically based method to recover lighting from a single image, enabling arbitrary virtual object compositing into indoor and outdoor scenes, as well as material and tone-mapping optimization. Project page: `https://research.nvidia.com/labs/toronto-ai/DiPIR/`

To address these challenges, we propose to reuse the strong image generation priors learned by large diffusion models (DMs) [46] as a guidance for inverse rendering. Unlike hand-crafted or supervised data-driven priors that are often specific to a domain, DMs are trained on massive datasets and show a remarkable "understanding" of the world and the underlying physical concepts. While DMs still often fail to produce accurate lighting effects such as shadows and reflections [49] in their generations, we observe that they can provide valuable guidance when combined with a physically-based renderer and are adapted to the scene.

Specifically, we present *Diffusion Prior for Inverse Rendering* (DiPIR), building on three main contributions. **First**, we use a physically based renderer to accurately simulate the interaction between the light and the 3D asset to generate the final composited image. We also account for the unknown tone-mapping curve to mimic the camera sensor response. **Second**, we propose a lightweight personalization scheme of the pre-trained DM, based on the input image and the type of inserted asset. **Third**, we design a variant of the SDS loss [45] which makes use of this personalization and improves training stability.

In DiPIR, the DM acts similarly to a human evaluator. It takes the edited image as input and propagates the feedback signal to physically-based scene attributes via differentiable rendering, thus enabling end-to-end optimization. We experimentally show that DiPIR outperforms existing state-of-the-art lighting estimation methods for object insertion across indoor and outdoor datasets.

2   Related Work

Inverse rendering   is the task of recovering intrinsic properties of a scene, including materials, shape, and lighting, from a single or multiple images [5].

The main challenge of inverse rendering lies in the ill-posed nature of the task. Prior knowledge about materials and lighting effects is crucial to solving underconstrained inverse rendering problems, but how to best define and incorporate such priors remains an open research question.

Early methods formulated inverse rendering as an optimization problem and used hand-crafted regularization terms such as local smoothness and sparsity of materials [4, 5, 9, 18, 31, 75] to constrain the solution space. However, real-world lighting effects are often complex and difficult to describe using hand-crafted priors. The advent of deep learning has facilitated the learning of data-driven priors from ground truth supervision [6,8,30,34,35,38,50,62–64,69]. Yet, acquiring real-world data with accurately annotated intrinsic decomposition necessitates specialized capture devices [18] or expert annotators [6,30]. Collecting real-world datasets at a large scale is therefore extremely challenging. As a result, existing methods often resort to synthetic datasets [17, 35, 50, 56, 63, 77], or are carefully designed to be training data-efficient [22, 23, 62]. The resulting algorithms are therefore often tailored to specific domains, such as indoor [15–17, 50, 63] or day-time outdoor scenes [22, 23, 56, 62, 77].

Lighting estimation is a subtask of inverse rendering that specifically focuses on inferring lighting and is a core component of photorealistic virtual object insertion pipelines. Existing lighting estimation methods are often designed as feed-forward neural networks that take a single image as input and directly regress lighting information in form of spherical lobes [17, 35, 70, 76], environment maps [16, 50, 53, 56, 77], parametric light sources [15], or volumetric lighting [37, 54, 62, 63]. The environment lighting can also be recovered through the proxy geometry shown in the single image [32, 67]. Recently, lighting estimation was also formulated as a generative task. StyleLight [61] is based on a dual-branch StyleGAN model, which first performs GAN inversion through the LDR branch and then generates the panorama with the HDR branch. EverLight [12] instead regresses an initial HDR environment map from the input image and later refines it with a GAN model [27]. Concurrent to our work, DiffusionLight [44] proposed to inpaint a chrome ball in the center of the given image using an inpainting diffusion model and unwarp it to an environment map. To obtain the HDR environment map, DiffusionLight trains a LoRA [24] for exposure bracketing and merges multiple generated LDR chrome balls with different exposures.

Note that lighting is fundamentally a High Dynamic Range (HDR) quantity, and reproducing this full range still remains challenging. HDR is crucial for basic effects such as sharp shadows, particularly in outdoor day-time settings where the sun is several orders of magnitudes brighter than the rest of the environment.

Physically-based differentiable rendering. Rendering algorithms such as path tracing [26] aim to produce realistic images of virtual scenes by accurately simulating the physical processes involved in light transport. Physically-based rendering (PBR) typically uses a geometric representation of the scene, scattering functions modeling the behavior of the scene's surfaces, as well as camera and lighting models. Recent years have seen the development of many *differentiable* rendering methods and frameworks [25, 33, 42, 43, 60, 71, 72].

4 R. Liang et al.

While particular attention has been devoted to the difficult problem of computing derivatives *at discontinuities* [3, 33, 39, 65, 74], such as due to visibility changes, we focus here on the comparatively simpler lighting, material and tone-mapping derivatives, which generally do not introduce discontinuities.

Diffusion model priors and personalization. Priors learned by the text-to-image DMs [2, 11, 46, 48], on large-scale datasets can be adapted to various applications such as monocular depth estimation [28], authentic image completion [57], and image restoration [10]. Among various adaptation techniques, DreamBooth [47] proposes to finetune the DM on subject images, while Textual Inversion [14] optimizes new word embeddings for target generation. LoRA [24,51] simplifies adaptation by freezing the pretrained model weights and injecting trainable rank decomposition matrices, thus greatly reducing the number of trainable parameters. Most similar to ours are concurrent works on DM adaptation for lighting estimation [44] and intrinsic image decomposition [29]. The former adapts the DM to exposure bracketing for LDR to HDR lifting, while the latter fine-tunes a pretrained SD model to directly output the albedo and BRDF properties of an image.

3 Preliminaries

Diffusion Models. Diffusion models are a family of generative models built around two key processes. A forward process, which gradually adds noise to data samples $\mathbf{x} \sim p(\mathbf{x})$ removing their structure over time t. This is achieved using a noise schedule determined by α_t and σ_t as $\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. In contrast, the reverse process gradually removes this noise, restoring the structure. The reverse process is parameterized by a conditional neural network, $\boldsymbol{\epsilon_\theta}$, trained to predict the noise $\boldsymbol{\epsilon}$ at a given timestep t according to the following simplified objective [20]:

$$\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}),t\sim T} \left[w(t)||\boldsymbol{\epsilon_\theta}(\mathbf{x}_t,t,\boldsymbol{c}) - \boldsymbol{\epsilon}||_2^2 \right], \tag{1}$$

where $\boldsymbol{c}$ represents a condition (*e.g.* text, image, etc.) that allows controlling the generation process, $w(t)$ represents a time-conditional weighting, and T is a set containing a selection of timesteps.

In this work, we use a Latent Diffusion Model (LDM) [46] in which the diffusion process is conducted in a lower-dimensional latent space. Specifically, the encoder $\mathcal{E}$ maps samples from the data distribution $\mathbf{x} \sim p(\mathbf{x})$ into a latent space $\mathbb{Z}$. The decoder $\mathcal{D}$ performs the inverse operation, such that $\mathcal{D}\left(\mathcal{E}(\mathbf{x})\right) \approx \mathbf{x}$. As follows, for LDMs $\mathbf{x}$ in Eq. 1 is replaced by its latent $\mathbf{z} = \mathcal{E}(\mathbf{x})$.

Personalization and Fine-tuning. Fine-tuning all parameters of a pretrained DM requires significant computational resources and time. To alleviate this, LoRA [24] injects trainable low-rank decomposition matrices and aims to learn only the variations from the pretrained weights. Specifically, consider a linear layer represented as $\boldsymbol{h} = \boldsymbol{W}_0\boldsymbol{x}$. Here $\boldsymbol{W}_0 \in \mathbb{R}^{n\times n}$ and $\boldsymbol{x} \in \mathbb{R}^n$ are the pretrained
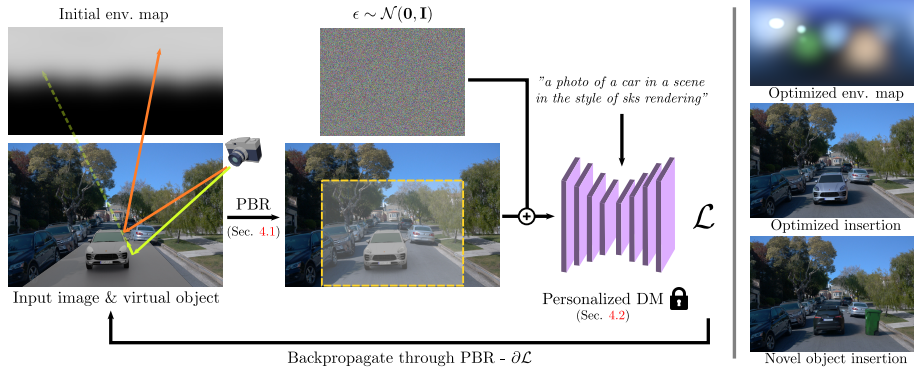
Fig. 2: Method overview. Given an input image, we first construct a virtual 3D scene with a virtual object and proxy plane. Our physically-based renderer then differentiably simulates the interactions of the optimizable environment map with the inserted virtual object and its effect on the background scene (shadowing) (*left*). At each iteration, the rendered image is diffused and passed through a personalized diffusion model (*middle*). The gradient of the adapted Score Distillation formulation is propagated back to the environment map and the tone-mapping curve through the differentiable renderer. Upon convergence, we recover lighting and tone-mapping parameters, which allow photorealistic compositing of virtual objects from a single image (*right*).

weights and input, respectively. Applying LoRA, this layer is modified to $\boldsymbol{h} = (\boldsymbol{W}_0 + \boldsymbol{\Delta W})\boldsymbol{x} = \boldsymbol{W}_0\boldsymbol{x} + \boldsymbol{A}\boldsymbol{B}\boldsymbol{x}$. Notably, $\boldsymbol{\Delta W} = \boldsymbol{A}\boldsymbol{B}$ is a combination of low-rank matrices $\boldsymbol{A} \in \mathbb{R}^{n \times r}$ and $\boldsymbol{B} \in \mathbb{R}^{r \times n}$ with $r \ll n$. During LoRA fine-tuning, only the added term $\boldsymbol{\Delta W}$ is updated while the parameters $\boldsymbol{W}_0$ remain unchanged. The rank r is a hyperparameter that trades efficiency with model capacity.

4 Method

Given a single image as input, DiPIR recovers scene lighting and tone-mapping parameters, with the goal of photorealistic insertion of virtual objects. An overview of our method is shown in Fig. 2. Sec. 4.1 describes our representation and the differentiable rendering process, while Sec. 4.2 and Sec. 4.3 provide details on diffusion model guidance and optimization formulation, respectively.

4.1 Physically-based Virtual Object Insertion

Virtual scene.   Inserting a virtual object $\mathcal{X}$ into an image $\mathbf{I}_{bg} \in \mathbb{R}^{h \times w \times 3}$ requires creating a 3D proxy virtual scene, viewed from the correct camera pose. Here, we assume that the user provides a specific placement (pose) for $\mathcal{X}$, but in some cases, an appropriate pose can also be determined automatically, e.g. by detecting the floor plane and scene scale.

To model the effects of the inserted object on the original image, such as shadows cast by $\mathcal{X}$, we also assume a known proxy geometry $\mathcal{P}$. We found a simple ground plane acting as a *shadow catcher* underneath the virtual object to be sufficient in all of our experiments. This proxy plane can be easily placed manually or automatically generated based on e.g. depth or LiDAR data.

Light representation.   We represent the scene's lighting $\mathbf{L}$ with a set of N optimizable Spherical Gaussian (SG) parameters $\{\mathbf{c}_k, \boldsymbol{\mu}_k, \sigma_k\}_{k=1}^{N} \in \mathbb{R}^{N \times 7}$, where the radiance for one SG lobe at the direction $\boldsymbol{v} \in \mathbb{R}^3$ is defined as

$$\mathbf{G}_k(\boldsymbol{v}; \mathbf{c}_k, \boldsymbol{\mu}_k, \sigma_k) = \mathbf{c}e^{-(1-\boldsymbol{v}\cdot\boldsymbol{\mu})/\sigma^2}, \quad \text{where } \mathbf{c}_k \in \mathbb{R}^3, \boldsymbol{\mu}_k \in \mathbb{R}^3, \sigma_k \in \mathbb{R}_+. \quad (2)$$

The overall environment map $\mathbf{L} \in \mathbb{R}^{H \times W \times 3}$ is computed as:

$$\mathbf{L}_{i,j} = \sum_{k=1}^{N} \mathbf{G}_k\left(\boldsymbol{v}_{i,j}; \mathbf{c}_k, \boldsymbol{\mu}_k, \sigma_k\right), \quad (3)$$

where $\boldsymbol{v}_{i,j}$ is the direction corresponding to the pixel (i, j) using the standard spherical environment parameterization. Note that we chose a SG-based lighting representation for its good convergence properties and simplicity, but many alternatives exist and might be combined with our method.

Differentiable rendering.   Inserting a virtual object into a scene involves simulating the interactions of the optimizable environment map with the inserted virtual object (foreground), and the inserted object's effect on the background scene (shadow). We describe below the details respectively.

Foreground image. Given the optimized light representation $\mathbf{L}$ from above and the virtual object $\mathcal{X}$ (including its geometry and materials), a foreground image $\mathbf{I}_{\text{fg}}$ of the inserted object can be rendered directly using standard path tracing:

$$\mathbf{I}_{\text{fg}} = \text{PathTrace}(\mathcal{X}, \mathbf{L}, D), \quad (4)$$

D is the maximum number of interactions along the light path. Since we do not assume that the provided proxy geometry is accurate nor that it has materials, we omit the effect of light reflected by the object into the scene, for example, due to a highly specular inserted object.

Shadow ratio. As the background image $\mathbf{I}_{\text{bg}}$ already faithfully represents the scene, our task is limited to simulating the appearance of the inserted object and its effect on the nearby parts of the scene, *i.e.* the shadows cast by the object onto the proxy geometry.

The shadow ratio $\boldsymbol{\beta}_{\text{shadow}} \in \mathbb{R}^{h \times w \times 3}$, which accounts for the effect of the object on the surrounding scene, is computed as the ratio between the radiance received by the proxy geometry with and without the inserted object present. A low value of $\boldsymbol{\beta}_{\text{shadow}}$ indicates a strongly shadowed region:

$$\boldsymbol{\beta}_{\text{shadow}} = \frac{\text{PathTrace}(\mathcal{X} \cup \mathcal{P}, \mathbf{L}, 1)}{\text{PathTrace}(\mathcal{P}, \mathbf{L}, 1)}, \quad (5)$$

This is similar to the method of Wang *et al.* [62], however we use Multiple Importance Sampling (MIS) [59] between lighting and BSDF for better sampling efficiency. The maximum path length has been limited to 1 interaction in order to reduce memory and computational cost. Unless specific material information is provided with the proxy geometry, we use a Lambertian BSDF, in which case the albedo cancels out in the ratio. In practice, the computation of the shadow ratio and foreground image are combined in order to reuse shared terms.

Tone-mapping.   To compensate for the unknown tone-mapping of the input image, we introduces an optimizable tone correction function $f(\cdot)$ that is applied on the inserted object $\hat{\mathbf{I}}_{\mathrm{fg}} = f(\mathbf{I}_{\mathrm{fg}}; \boldsymbol{\theta}_{\mathrm{fg}})$ and shadows $\hat{\boldsymbol{\beta}}_{\mathrm{shadow}} = f(\boldsymbol{\beta}_{\mathrm{shadow}}; \boldsymbol{\theta}_{\mathrm{shadow}})$. Our tone curve parameterization follows from [13] and employs a monotonic rational-quadratic spline as its basic building block. This spline is composed of K_{s} bins, each defined by the quotient of two quadratic polynomials. Notably, these functions are differentiable, and allow direct parameterization of the derivatives and heights at each knot. We apply a single spline for the foreground image, and different correction splines for each RGB channel for the shadow ratio to provide the flexibility to adjust the color of the shadow. We set $K_{\mathrm{s}} = 5$ for these curves.

Differentiability.   The final output image is an alpha-composite of the foreground object, shadows, and background image:

$$\mathbf{I}_{\mathrm{comp}} = (1 - \mathbf{V}(\mathcal{X})) \cdot \hat{\boldsymbol{\beta}}_{\mathrm{shadow}} \cdot \mathbf{I}_{\mathrm{bg}} + \mathbf{V}(\mathcal{X}) \cdot \hat{\mathbf{I}}_{\mathrm{fg}}, \tag{6}$$

where $\mathbf{V}(\mathcal{X})$ equals 1 when $\mathcal{X}$ is directly visible from the camera, and 0 otherwise.

As neither foreground rendering, shadow rendering, or the compositing operation require derivatives w.r.t. the object placement or other discontinuous quantities, we can rely on automatic differentiation to obtain gradients of any pixelwise loss w.r.t. the lighting or material properties. To this end, we use the Path Replay Backpropagation [60] integrator of Mitsuba 3 [25]. The optimizable parameters are the Spherical Gaussian coefficients used in the lighting representation, and the tone curves' parameters $\boldsymbol{\theta}_{\mathrm{fg}}, \boldsymbol{\theta}_{\mathrm{shadow}}$.

While each of the operations above include simplifications (e.g. we do not account for secondary lighting from the proxy geometry), we have found them to be sufficient in practice: the remaining imperfections in the simulation can be sufficiently offset by the lighting optimization.

4.2   Diffusion Guidance

The composited image produced by our differentiable rendering pipeline serves as the input to a DM that is used to compute a guidance signal, employing an optimization objective similar to Score Distillation Sampling (SDS) [45]. However, while DMs inherently have robust priors for lighting, we found that they do not provide out-of-the-box the necessary guidance for our specific needs. Consequently, we propose an adaptive score distillation loss specifically designed for object insertion tasks that exploits a personalization strategy which we detail in the following section.

Personalization with concept preservation.   Off-the-shelf DMs often do not provide robust guidance for virtual object insertion, especially in out-of-distribution scenes such as outdoor driving environments. A potential solution, inspired by [57], is to adapt the DM using an image from the target scene. In our experience, however, this approach often resulted in too much overfitting to the target scene's content, reducing the model's ability to adapt to the scene with a newly inserted object. This led to artifacts and an unstable optimization process.

To mitigate this issue, we propose fine-tuning the DM with a focus on preserving the identity of the objects to be inserted (Fig. 3). We specifically achieved this by generating additional synthetic images for the insertable class concept (e.g. car). We sample those images from the off-the-self DM, starting with a base prompt such as "a photo of a car" and appending attributes such as color, background, lighting, and size to ensure diversity in the generated data.

"a scene in the style of sks rendering"

Input image

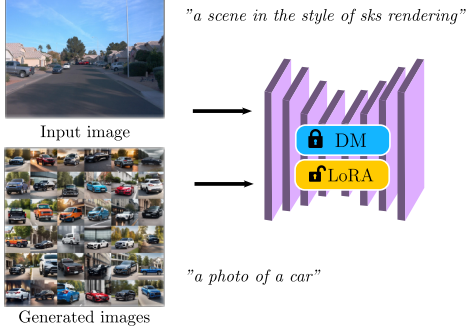*"a photo of a car"*

Generated images

Fig. 3: Personalization with concept preservation.

We employ LoRA with rank 4 to fine-tune the DM, combining the in-domain target example with this supplemental data. Our training follows the objective described in Eq. 1, where c corresponds to two predefined prompts: *"a scene in the style of sks rendering"* for the target image, and *"a photo of a {concept class}"*. We sample approximately 30-40 supplementary images for indoor scenes and 200 for outdoor scenes. The total time spent for fine-tuning is typically less than 15 minutes on one high-end GPU with mixed precision training.

Score Distillation with adapted guidance. Score Distillation Sampling [45] leverages a pretrained DM to guide the optimization of a differentiable, parametric image-rendering function $g_{\phi} := x$. Specifically, the parameters ϕ in our scenario corresponds to parameters of the Spherical Gaussian lighting and tone-mapping curves, are updated using the gradient:

$$\nabla_{\phi}\mathcal{L}_{\text{SDS}}(\phi,\theta) := \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t\sim T}\left[w(t)\, (\hat{\epsilon}_{\theta}(z_t,t,c) - \epsilon)\, \frac{\partial z_t}{\partial\phi} \right], \qquad (7)$$

where $z = \mathcal{E}(g_{\phi})$ and $\hat{\epsilon}_{\theta}(z_t,t,c) := (1+s)\,\epsilon_{\theta}(z_t,t,c) - s\,\epsilon_{\theta}(z_t,t,\varnothing)$. $\hat{\epsilon}$ denotes the classifier free guidance (CFG) version [21] of ϵ_{θ} used in text-conditioned DMs to enable higher quality generation via a guidance scale parameter s.

Unfortunately, we encountered training instabilities when applying the original formulation of the SDS loss to our problem. Instead, we adopt the following alternative that integrates the LoRA personalization, and we call it LDS loss:

$$\nabla_{\phi}\mathcal{L}_{\text{LDS}}(\phi,\theta) := \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t\sim T}\left[w(t)\, \left(\epsilon_{(\theta+\Delta W)}(z_t,t,c) - \epsilon_{\theta}(z_t,t,\varnothing)\right)\, \frac{\partial z_t}{\partial\phi} \right],$$
$$(8)$$

where $\epsilon_{(\theta+\Delta W)}$ represents the predicted noise of the LoRA personalized model described in the previous section. Notably, this loss function bears resemblance to the Classifier Score Distillation (CSD) [68]. However, it's important to note that in our case, the delta is calculated between the adapted and non-adapted versions of the diffusion model. Intuitively, this delta guides the optimization process in a direction determined by the personalized model which preserves the concept of the inserted object while also capturing the appearance and semantics of the specific scene.

4.3   Optimization Formulation

After completing the diffusion model's personalization, we optimize the lighting and tone-mapping parameters using the following loss function:

$$\mathcal{L} = \mathcal{L}_{\mathrm{LDS}} + \lambda_{\mathrm{consistency}}\mathcal{L}_{\mathrm{consistency}} + \lambda_{\mathrm{reg}}\mathcal{L}_{\mathrm{reg}}. \tag{9}$$

The loss' main component is the diffusion guidance $\mathcal{L}_{\mathrm{LDS}}$, which provides a perceptual realism objective on the edited result with object insertion. We use the text prompt *"a photo of a {concept class} in a scene in the style of sks rendering"*, which is a natural combination of the two text prompts used in personalization (Fig. 3). Here *{concept class}*, *e.g. car*, provides the context of the object to insert, while the personalized token *sks* provides the lighting style of the input image. The diffusion guidance $\mathcal{L}_{\mathrm{LDS}}$ is applied on the edited image $\mathbf{I}_{\mathrm{comp}}$ (Eq. 6), and backpropagated to the optimizable parameters through the differentiable rendering process (Eq. 8).

The two regularizers $\mathcal{L}_{\mathrm{consistency}}$ and $\mathcal{L}_{\mathrm{reg}}$ are related to environment map fusion, which we detail below.

Environment map initialization and fusion. The personalized DM provides guidance in two ways: (i) encouraging lighting consistency of the foreground object and the scene, such as the reflections and scale of the inserted object; and (ii) encouraging accurate shadows cast by the inserted object onto the background scene, such as the scale, direction, and color of the shadow. However, we empirically observe that these two signals can conflict in the early phase of the optimization.

To address this, we initialize two separate environment maps, $\mathbf{L}^{\mathrm{fg}}, \mathbf{L}^{\mathrm{shadow}} \in \mathbb{R}^{H \times W \times 3}$, to light the foreground inserted object (Eq. 4) and cast shadows (Eq. 5) respectively. As the optimization progresses, the two environment maps are progressively fused into a single environment map $\mathbf{L}^{\mathrm{fused}}$ through scaling the the $\mathbf{L}^{\mathrm{fg}}$ by the relative luminance of between $\mathbf{L}^{\mathrm{fg}}$ and $\mathbf{L}^{\mathrm{shadow}}$. We also use the following two regularization terms for this fusion process. First, we encourage the consistency between the normalized luminance $\tilde{\mathbf{L}}^{\mathrm{fg}}, \tilde{\mathbf{L}}^{\mathrm{shadow}} \in \mathbb{R}^{H \times W}$ of the environment maps by minimizing:

$$\mathcal{L}_{\mathrm{consistency}} = -\sum_{i,j} \tilde{\mathbf{L}}_{i,j}^{\mathrm{shadow}} \log\left(\tilde{\mathbf{L}}_{i,j}^{\mathrm{fg}}\right) \Delta\Omega_{i,j} \tag{10}$$

where $\Delta\Omega_{i,j}$ is the corresponding solid angle at pixel (i,j) and the gradient for $\tilde{\mathbf{L}}^{\mathrm{shadow}}$ is detached. Second, as the shadow environment map $\mathbf{L}^{\mathrm{shadow}}$ is supervised mainly via the shadow ratio $\boldsymbol{\beta}_{\mathrm{shadow}} \in \mathbb{R}^{H \times W \times 3}$ of Eq. 5, to encourage concentrated high peaks for the sharp shadows and suppress the ambient light in $\mathbf{L}^{\mathrm{shadow}}$, we also add a L2 regularizer in log-space with the Cauchy loss [7]:

$$\mathcal{L}_{\mathrm{reg}} = \sum_{i,j,c} \log\left(1 + 2\left(\mathbf{L}_{i,j,c}^{\mathrm{shadow}}\right)^2\right) \Delta\Omega_{i,j}. \tag{11}$$

Please refer to the Supplement for more details.

Table 1: Quantitative user study on outdoor street scenes. For each scene, users are shown two results—one produced by our method, and another produced by one of the baselines—and select which is more realistic. We report the results averaged across 3 user studies with 9 users each. Our method outperforms all baselines ($> 50\%$) and is preferred in almost all illumination conditions.

	Daytime		Twilight	Night	All scenes
	Sunny	Cloudy			
Hold-Geoffroy *et al*. [22]	60.8%	66.7%	74.1%	85.7%	68.8%
NLFE [62]	80.4%	73.3%	44.4%	52.4%	67.4%
StyleLight [61]	76.5%	91.1%	66.7%	66.7%	77.8%
DiffusionLight [44]	80.4%	68.9%	55.6%	71.4%	70.8%

Table 2: Quantitative evaluation on PolyHaven scenes. We report user study preference scores similar to Table 1. Metrics are computed w.r.t. to a "reference" image where the virtual object is lit by the ground-truth environment map.

Methods	Ours preferred ↓	RMSE ↓	SSIM ↑	LPIPS [73] ↓	si-RMSE [18] ↓
Wang *et al*. [63]	84.8%	0.063	0.985	0.0175	0.037
StyleLight [61]	75.8%	0.056	0.986	0.0202	0.039
DiffusionLight [44]	66.7%	0.062	0.985	0.0162	0.034
Ours	/	**0.048**	**0.989**	**0.0147**	**0.027**

Training details. The personalized DM receives a crop of $\mathbf{I}_{\mathrm{comp}}$ (Eq. 6) with added noise. Since the inserted object likely does not cover the full background image, we use the visibility mask of the virtual object $V(\mathcal{X})$ to locate the 2D bounding box of the inserted object, and randomly crop around the inserted object. The size of the cropped image is also randomized, with smaller crops covering local details of the inserted object and larger crops providing more visual cues of the lighting effects within the scene. Each scene is optimized for 600 iterations, and the maximum strength of the diffusion guidance is linearly decreased from 0.6 to 0.3. We use $\lambda_{\mathrm{consistency}} = 0.03$ and $\lambda_{\mathrm{reg}} = 0.01$.

5   Experiments

In this section, we evaluate DiPIR on a collection of outdoor and indoor scenes, covering diverse lighting conditions and different application domains. We first outline the experiment settings in Sec. 5.1 and then provide evaluation results on benchmark datasets in Sec. 5.2. Finally, we conduct an ablation study in Sec. 5.3.

5.1   Experiment Settings

Waymo dataset. Following the data split from [66], we use 48 scenes with diverse illumination conditions from the Waymo Open Dataset [55] including 32 daytime (17 sunny and 15 cloudy), 9 twilight, and 7 night scenes. Each scene consists of an input image, a 3D car asset randomly selected from an asset bank, the ground plane, and a location to insert the car. Inspired by applications in synthetic data generation, this process is fully automatic: we use the LiDAR point cloud and semantic segmentation [58] to fit a ground plane and detect the empty space to insert a car.

PolyHaven dataset. We use 11 HDR environment maps from PolyHaven [1] and manually place a known ground plane and virtual object in each scene. Each

Reference	Ours	DiffusionLight [44]	StyleLight [61]	Wang *et al.* [63]

Fig. 4: Comparisons on inserting objects into cropped HDRIs from PolyHaven.

Ours	DiffusionLight [44]	StyleLight [61]	NLFE [62]	H-G *et al.* [22]

Fig. 5: Inserting car assets into Waymo driving scenes. Note the direction and sharpness of shadows, as well as overall brightness, color and specular highlights on inserted cars.

scenario therefore consists of an LDR background image $\mathbf{I}_{bg}$ rendered directly from the environment map, the posed virtual object, and the proxy plane. A pseudo-ground-truth rendering is created by lighting the inserted object with the environment map itself.

User study. To evaluate the perceptual realism of the virtual object insertion, we conducted a user study where participants received a pair of two object insertion results in a random order – one generated using our proposed method, the other using a baseline approach. The participants were then instructed to compare the differences between the two results, inspect the lighting effects of the inserted object, and choose the image they perceived as *more realistic*. We invited 9 users to perform a binary selection for each image pair and used majority voting to determine the preferred image for each comparison. In the following, we report the percentage of times that our method was selected over the baseline. This process was repeated three times and a preferred percentage $> 50\%$ indicates

Ours outperforming baselines. We include more details about the user study and statistical evaluation of its results in the Supplement.

Baselines.   On the Waymo dataset of urban street scenes, we compare our method to the representative outdoor lighting estimation methods of Hold-Geoffroy *et al.* [22] and NLFE [62]; as well as the generative lighting estimation methods StyleLight [61] and DiffusionLight [44]. On the PolyHaven dataset, we compare with Wang *et al.* [63], StyleLight [61], DiffusionLight [44], and pseudo-ground-truth rendering.

5.2   Evaluation on Benchmark Datasets

Urban street scenes.   In Table 1, we provide quantitative user study results on the Waymo dataset [55], separated into 4 subsets based on illumination conditions, as well as on all scenes. Compared to all baselines, our method outperforms prior state-of-the-art methods (chosen $> 50\%$), and is preferred in almost every subset.

StyleLight [61] is primarily trained on indoor HDR panoramas and suffers from domain gap to outdoor scenes. Similarly, the sky model predicted by Hold-Geoffroy *et al.* [22] is tailored to outdoor daytime scenes and suffers severely from out-of-domain twilight and night scenes. NLFE [62] unprojects the surrounding scene geometry and predicts volumetric lighting, which makes it generalize better to night scenes, but it can fail to estimate an accurate scale and color of the peak for daytime scenes. DiffusionLight [44], a concurrent work, shows impressive results by inpainting a chrome sphere and predicting environment maps with realistic high-frequency details. However, it often struggles to predict a high-intensity peak in sunny daytime scenes or a proper scale for night scenes.

A qualitative comparison is shown in Fig. 5. Our method achieves high-quality object insertion, making it a promising approach for applications such as simulating synthetic data, augmented reality navigation and urban planning.

PolyHaven dataset.   We show the quantitative comparison in Table 2 and qualitative results in Fig. 4. Our method is preferred over all baseline methods in the user study, and outperforms baselines on quantitative metrics. Our method produces virtual object insertion that naturally and consistently blend in the scenes, making it promising for virtual production tasks.

5.3   Ablation Study

We extensively ablate our proposed design choices on diffusion guidance and scene representations, and provide the quantitative ablation in Table 3 with user study and qualitative results in Fig. 6. For the diffusion guidance, we compare with simplified or alternative versions: (i) Ours (SDS) removes the adaptive guidance (Eq. 8) and uses the original SDS [45] formulation (Eq. 7); (ii) Ours (SDS w/o LoRA) uses the original SDS loss with non-adapted DM; (iii) Ours (w/o concept preservation) uses our LDS loss with adapted DM but not using any class images for concept preservation. We additionally compare with a baseline of "dataset update" which applies a photometric L1 loss on DM repeatedly edited results

Methods	Ours preferred ↓
Ours (dataset update)	85.2%
Ours (SDS [45])	74.1%
Ours (SDS [45] w/o LoRA)	90.7%
Ours (w/o concept preservation)	64.8%
Ours (w/o tone curve)	68.5%
Ours (w/o env. map fusion)	66.7%

Table 3: Ablation study on outdoor driving scenes [55]. We report the percentage of images that users preferred DiPIR compared to its ablated versions. Our full pipeline produces results that are preferred more often over its ablated versions.
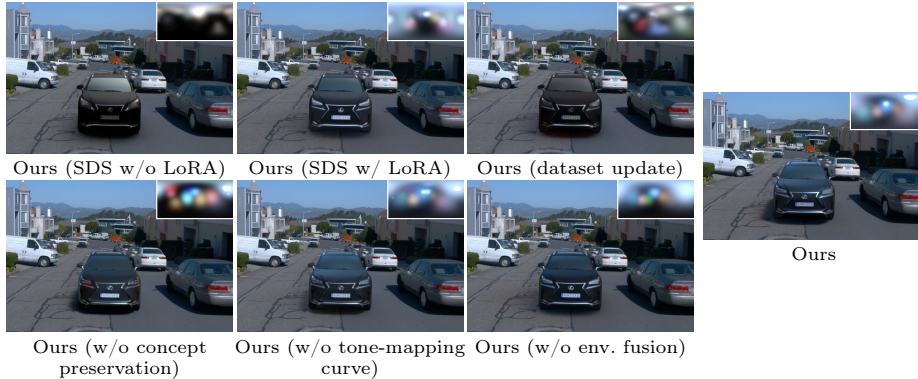

Ours (SDS w/o LoRA)   Ours (SDS w/ LoRA)   Ours (dataset update)

Ours

Ours (w/o concept preservation)   Ours (w/o tone-mapping curve)   Ours (w/o env. fusion)

Fig. 6: Qualitative ablation study of our design choices.

via SDEdit [41], denoted as Ours (dataset update). Although this approach is adopted in text-guided scene editing [19], we found it challenging to produce stable gradients from such a "discrete" guidance.

When designing the light and tone representations, we observe that the trainable tone-mapping curve provides additional flexibility to compensate for unknown tone-mapping in the input image, and better match the color and scale of the shadows. The two environment map fusion scheme (Sec. 4.3) allows to robustly recover the high intensity peaks in the early phase of training, and converge to better quality. Their respective effect is ablated as "Ours (w/o tone curve)" and "Ours (w/o env. map fusion)".

6   Applications

Since our method recovers physically based lighting information, arbitrary new virtual objects can be inserted after the optimization, as shown in Fig. 2. DiPIR can also optimize other scene attributes such as materials and local lighting. We perform preliminary experiments in this direction.

Material optimization.   When combined with differentiable rendering, DMs can provide a guidance signal for material attributes, as shown in Fig. 7. Given a purely diffuse car and enabling Metallic and Roughness properties as optimizable parameters, the diffusion guidance can optimize and make the car look more shiny. By additionally changing the text prompt to "a carmine red car" and making the base color of the car an optimizable parameter, we show that the DM can propagate the text-condition to the PBR attribute and change the color of the car to red. When enabling local emission as an optimizable parameter, the diffusion model can also turn on the headlights of cars in night scenes.
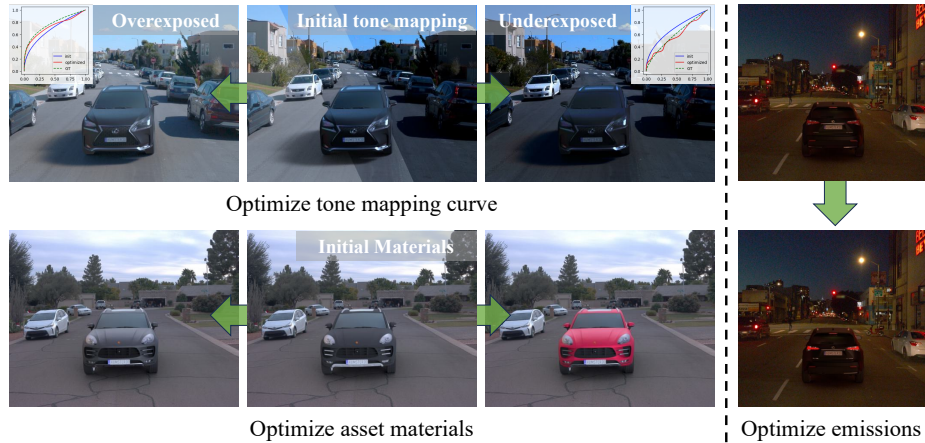
Fig. 7: Our physically based inverse rendering pipeline unlocks further applications such as material, local emission and tone-mapping refinement.

Tone-mapping adjustment.   We use a controlled experiment to further evaluate how well DMs understand tone-mapping. In Fig. 7, we freeze the estimated environment map and apply a manual tone adjustment on the background image. The diffusion guidance is used to optimize the tone curve such that the inserted object matches the surrounding background in the final composited result.

7   Discussion

Our method leverages large diffusion models' inherent scene understanding capabilities as guidance to a physically based inverse rendering pipeline. We design a diffusion guidance signal with scene-specific personalization and a differentiable inverse rendering pipeline to recover lighting and tone-mapping parameters. Our method enables inserting virtual objects into scenes, but also optimizing other scene parameters such as the materials of the inserted object or account for tone-mapping mismatches between cameras. We believe that this combination of the differentiable rendering process and data-driven priors can be used successfully in many other content creation applications such as relighting and animation.

Limitations and future work.   Our Spherical Gaussians-based lighting representation is adequate for general objects [35], but might not behave realistically for highly specular materials. For more complex lighting representations adding generative priors on the environment map [40] is a direction worth exploring. The rendering formulation could be extended to account for effects such as reflections from the scene itself onto the inserted object (e.g. color bleeding), but that might introduce more ambiguities and require knowing the materials of the proxy geometry (refer to Supplement C.4 for failure case examples). Finally, while DM personalization significantly improves the quality of the results, it adds overhead and complexity to the pipeline. Recent personalization methods that do not require test-time finetuning [52] could be used to mitigate this overhead.

Acknowledgements

References

1. Poly Haven - The Public 3D Asset Library, `https://polyhaven.com`
2. Balaji, Y., Nah, S., Huang, X., Vahdat, A., Song, J., Zhang, Q., Kreis, K., Aittala, M., Aila, T., Laine, S., Catanzaro, B., Karras, T., Liu, M.Y.: ediff-i: Text-to-image diffusion models with ensemble of expert denoisers. arXiv preprint arXiv:2211.01324 (2022)
3. Bangaru, S.P., Li, T.M., Durand, F.: Unbiased warped-area sampling for differentiable rendering. ACM Trans. Graph. **39**(6) (nov 2020). `https://doi.org/10.1145/3414685.3417833`, `https://doi.org/10.1145/3414685.3417833`
4. Barron, J.T., Malik, J.: Shape, illumination, and reflectance from shading. IEEE transactions on pattern analysis and machine intelligence **37**(8), 1670–1687 (2014)
5. Barrow, H., Tenenbaum, J., Hanson, A., Riseman, E.: Recovering intrinsic scene characteristics. Comput. Vis. Syst **2**, 3–26 (1978)
6. Bell, S., Bala, K., Snavely, N.: Intrinsic images in the wild. ACM Transactions on Graphics (TOG) **33**(4), 159 (2014)
7. Black, M.J., Anandan, P.: The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. Computer vision and image understanding **63**(1), 75–104 (1996)
8. Boss, M., Jampani, V., Kim, K., Lensch, H.P., Kautz, J.: Two-shot spatially-varying brdf and shape estimation. In: CVPR (2020)
9. Bousseau, A., Paris, S., Durand, F.: User-assisted intrinsic images. In: ACM Transactions on Graphics (TOG). vol. 28, p. 130. ACM (2009)
10. Chari, P., Ma, S., Ostashev, D., Kadambi, A., Krishnan, G., Wang, J., Aberman, K.: Personalized restoration via dual-pivot tuning. arXiv preprint arXiv:2312.17234 (2023)
11. Dai, X., Hou, J., Ma, C.Y., Tsai, S., Wang, J., Wang, R., Zhang, P., Vandenhende, S., Wang, X., Dubey, A., et al.: Emu: Enhancing image generation models using photogenic needles in a haystack. arXiv preprint arXiv:2309.15807 (2023)
12. Dastjerdi, M.R.K., Eisenmann, J., Hold-Geoffroy, Y., Lalonde, J.F.: Everlight: Indoor-outdoor editable hdr lighting estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7420–7429 (October 2023)
13. Durkan, C., Bekasov, A., Murray, I., Papamakarios, G.: Neural spline flows. Advances in neural information processing systems **32** (2019)
14. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A.H., Chechik, G., Cohen-Or, D.: An image is worth one word: Personalizing text-to-image generation using textual inversion (2022). `https://doi.org/10.48550/ARXIV.2208.01618`, `https://arxiv.org/abs/2208.01618`
15. Gardner, M.A., Hold-Geoffroy, Y., Sunkavalli, K., Gagné, C., Lalonde, J.F.: Deep parametric indoor lighting estimation. In: ICCV. pp. 7175–7183 (2019)

16. 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)
17. Garon, M., Sunkavalli, K., Hadap, S., Carr, N., Lalonde, J.F.: Fast spatially-varying indoor lighting estimation. In: CVPR. pp. 6908–6917 (2019)
18. Grosse, R., Johnson, M.K., Adelson, E.H., Freeman, W.T.: Ground truth dataset and baseline evaluations for intrinsic image algorithms. In: ICCV. pp. 2335–2342. IEEE (2009)
19. Haque, A., Tancik, M., Efros, A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2023)
20. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. arXiv preprint arxiv:2006.11239 (2020)
21. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
22. Hold-Geoffroy, Y., Athawale, A., Lalonde, J.F.: Deep sky modeling for single image outdoor lighting estimation. In: CVPR. pp. 6927–6935 (2019)
23. Hold-Geoffroy, Y., Sunkavalli, K., Hadap, S., Gambaretto, E., Lalonde, J.F.: Deep outdoor illumination estimation. In: CVPR. pp. 7312–7321 (2017)
24. Hu, E.J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=nZeVKeeFYf9
25. Jakob, W., Speierer, S., Roussel, N., Nimier-David, M., Vicini, D., Zeltner, T., Nicolet, B., Crespo, M., Leroy, V., Zhang, Z.: Mitsuba 3 renderer (2022), https://mitsuba-renderer.org
26. Kajiya, J.T.: The rendering equation. In: Proceedings of the 13th annual conference on Computer graphics and interactive techniques. pp. 143–150 (1986)
27. Karimi Dastjerdi, M.R., Hold-Geoffroy, Y., Eisenmann, J., Khodadadeh, S., Lalonde, J.F.: Guided co-modulated gan for 360 field of view extrapolation. International Conference on 3D Vision (3DV) (2022)
28. Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R.C., Schindler, K.: Repurposing diffusion-based image generators for monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
29. Kocsis, P., Sitzmann, V., Nießner, M.: Intrinsic image diffusion for single-view material estimation. In: arxiv (2023)
30. Kovacs, B., Bell, S., Snavely, N., Bala, K.: Shading annotations in the wild. In: CVPR. pp. 6998–7007 (2017)
31. Land, E.H., McCann, J.J.: Lightness and retinex theory. Josa **61**(1), 1–11 (1971)
32. LeGendre, C., Ma, W.C., Fyffe, G., Flynn, J., Charbonnel, L., Busch, J., Debevec, P.: Deeplight: Learning illumination for unconstrained mobile mixed reality. In: CVPR. pp. 5918–5928 (2019)
33. Li, T.M., Aittala, M., Durand, F., Lehtinen, J.: Differentiable monte carlo ray tracing through edge sampling. ACM Trans. Graph. **37**(6) (dec 2018). https://doi.org/10.1145/3272127.3275109, https://doi.org/10.1145/3272127.3275109
34. Li, Z., Snavely, N.: Cgintrinsics: Better intrinsic image decomposition through physically-based rendering. In: ECCV. pp. 371–387 (2018)
35. 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: CVPR. pp. 2475–2484 (2020)

36. 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)

37. Li, Z., Yu, L., Okunev, M., Chandraker, M., Dong, Z.: Spatiotemporally consistent hdr indoor lighting estimation. ACM Trans. Graph. **42**(3) (jun 2023). `https://doi.org/10.1145/3595921`, `https://doi.org/10.1145/3595921`

38. Li, Z., Yu, T.W., Sang, S., Wang, S., Bi, S., Xu, Z., Yu, H.X., Sunkavalli, K., Hašan, M., Ramamoorthi, R., et al.: Openrooms: An end-to-end open framework for photorealistic indoor scene datasets. arXiv preprint arXiv:2007.12868 (2020)

39. Loubet, G., Holzschuch, N., Jakob, W.: Reparameterizing discontinuous integrands for differentiable rendering. ACM Trans. Graph. **38**(6) (nov 2019). `https://doi.org/10.1145/3355089.3356510`, `https://doi.org/10.1145/3355089.3356510`

40. Lyu, L., Tewari, A., Habermann, M., Saito, S., Zollhöfer, M., Leimküehler, T., Theobalt, C.: Diffusion posterior illumination for ambiguity-aware inverse rendering. ACM Transactions on Graphics **42**(6) (2023)

41. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: SDEdit: Guided image synthesis and editing with stochastic differential equations. In: International Conference on Learning Representations (2022)

42. Nimier-David, M., Speierer, S., Ruiz, B., Jakob, W.: Radiative backpropagation: an adjoint method for lightning-fast differentiable rendering. ACM Trans. Graph. **39**(4) (aug 2020). `https://doi.org/10.1145/3386569.3392406`, `https://doi.org/10.1145/3386569.3392406`

43. Nimier-David, M., Vicini, D., Zeltner, T., Jakob, W.: Mitsuba 2: a retargetable forward and inverse renderer. ACM Trans. Graph. **38**(6) (nov 2019). `https://doi.org/10.1145/3355089.3356498`, `https://doi.org/10.1145/3355089.3356498`

44. Phongthawee, P., Chinchuthakun, W., Sinsunthithet, N., Raj, A., Jampani, V., Khungurn, P., Suwajanakorn, S.: Diffusionlight: Light probes for free by painting a chrome ball. In: ArXiv (2023)

45. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv (2022)

46. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models (2021)

47. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation (2022)

48. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in Neural Information Processing Systems **35**, 36479–36494 (2022)

49. Sarkar, A., Mai, H., Mahapatra, A., Lazebnik, S., Forsyth, D., Bhattad, A.: Shadows don't lie and lines can't bend! generative models don't know projective geometry...for now (2023)

50. 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: ICCV (2019)

51. Shah, V., Ruiz, N., Cole, F., Lu, E., Lazebnik, S., Li, Y., Jampani, V.: Ziplora: Any subject in any style by effectively merging loras (2023)

52. Shi, J., Xiong, W., Lin, Z., Jung, H.J.: Instantbooth: Personalized text-to-image generation without test-time finetuning. arXiv preprint arXiv:2304.03411 (2023)

53. Song, S., Funkhouser, T.: Neural illumination: Lighting prediction for indoor environments. In: CVPR. pp. 6918–6926 (2019)

54. Srinivasan, P.P., Mildenhall, B., Tancik, M., Barron, J.T., Tucker, R., Snavely, N.: Lighthouse: Predicting lighting volumes for spatially-coherent illumination. In: CVPR. pp. 8080–8089 (2020)
55. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR (2020)
56. Tang, J., Zhu, Y., Wang, H., Chan, J.H., Li, S., Shi, B.: Estimating spatially-varying lighting in urban scenes with disentangled representation. In: ECCV (2022)
57. Tang, L., Ruiz, N., Qinghao, C., Li, Y., Holynski, A., Jacobs, D.E., Hariharan, B., Pritch, Y., Wadhwa, N., Aberman, K., Rubinstein, M.: Realfill: Reference-driven generation for authentic image completion. arXiv preprint arXiv:2309.16668 (2023)
58. Tao, A., Sapra, K., Catanzaro, B.: Hierarchical multi-scale attention for semantic segmentation. arXiv preprint arXiv:2005.10821 (2020)
59. Veach, E., Guibas, L.J.: Optimally combining sampling techniques for monte carlo rendering. In: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques. pp. 419–428 (1995)
60. Vicini, D., Speierer, S., Jakob, W.: Path replay backpropagation: differentiating light paths using constant memory and linear time. ACM Trans. Graph. **40**(4) (jul 2021). `https://doi.org/10.1145/3450626.3459804`, `https://doi.org/10.1145/3450626.3459804`
61. Wang, G., Yang, Y., Loy, C.C., Liu, Z.: Stylelight: Hdr panorama generation for lighting estimation and editing. In: European Conference on Computer Vision (ECCV) (2022)
62. Wang, Z., Chen, W., Acuna, D., Kautz, J., Fidler, S.: Neural light field estimation for street scenes with differentiable virtual object insertion. In: ECCV (2022)
63. Wang, Z., Philion, J., Fidler, S., Kautz, J.: Learning indoor inverse rendering with 3d spatially-varying lighting. In: ICCV (2021)
64. Wimbauer, F., Wu, S., Rupprecht, C.: De-rendering 3d objects in the wild. In: CVPR (2022)
65. Yan, K., Lassner, C., Budge, B., Dong, Z., Zhao, S.: Efficient estimation of boundary integrals for path-space differentiable rendering. ACM Trans. Graph. **41**(4) (jul 2022). `https://doi.org/10.1145/3528223.3530080`, `https://doi.org/10.1145/3528223.3530080`
66. Yang, J., Ivanovic, B., Litany, O., Weng, X., Kim, S.W., Li, B., Che, T., Xu, D., Fidler, S., Pavone, M., Wang, Y.: Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. arXiv preprint arXiv:2311.02077 (2023)
67. Yu, H.X., Agarwala, S., Herrmann, C., Szeliski, R., Snavely, N., Wu, J., Sun, D.: Accidental light probes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12521–12530 (2023)
68. Yu, X., Guo, Y.C., Li, Y., Liang, D., Zhang, S.H., Qi, X.: Text-to-3d with classifier score distillation (2023)
69. Yu, Y., Smith, W.A.: Inverserendernet: Learning single image inverse rendering. In: CVPR (2019)
70. Zhan, F., Zhang, C., Yu, Y., Chang, Y., Lu, S., Ma, F., Xie, X.: Emlight: Lighting estimation via spherical distribution approximation. In: Proceedings of the AAAI Conference on Artificial Intelligence (2021)
71. Zhang, C., Miller, B., Yan, K., Gkioulekas, I., Zhao, S.: Path-space differentiable rendering. ACM Trans. Graph. **39**(4) (aug 2020). `https://doi.org/10.1145/3386569.3392383`, `https://doi.org/10.1145/3386569.3392383`

72. Zhang, C., Yu, Z., Zhao, S.: Path-space differentiable rendering of participating media. ACM Trans. Graph. **40**(4) (jul 2021). `https://doi.org/10.1145/3450626.3459782`, `https://doi.org/10.1145/3450626.3459782`
73. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
74. Zhang, Z., Roussel, N., Jakob, W.: Projective sampling for differentiable rendering of geometry. ACM Trans. Graph. **42**(6) (dec 2023). `https://doi.org/10.1145/3618385`, `https://doi.org/10.1145/3618385`
75. Zhao, Q., Tan, P., Dai, Q., Shen, L., Wu, E., Lin, S.: A closed-form solution to retinex with nonlocal texture constraints **34**(7), 1437–1444 (2012)
76. Zhao, Y., Guo, T.: Pointar: Efficient lighting estimation for mobile augmented reality. arXiv preprint arXiv:2004.00006 (2020)
77. Zhu, Y., Zhang, Y., Li, S., Shi, B.: Spatially-varying outdoor lighting estimation from intrinsics. In: CVPR (2021)