Inverse Global Illumination using a Neural Radiometric Prior
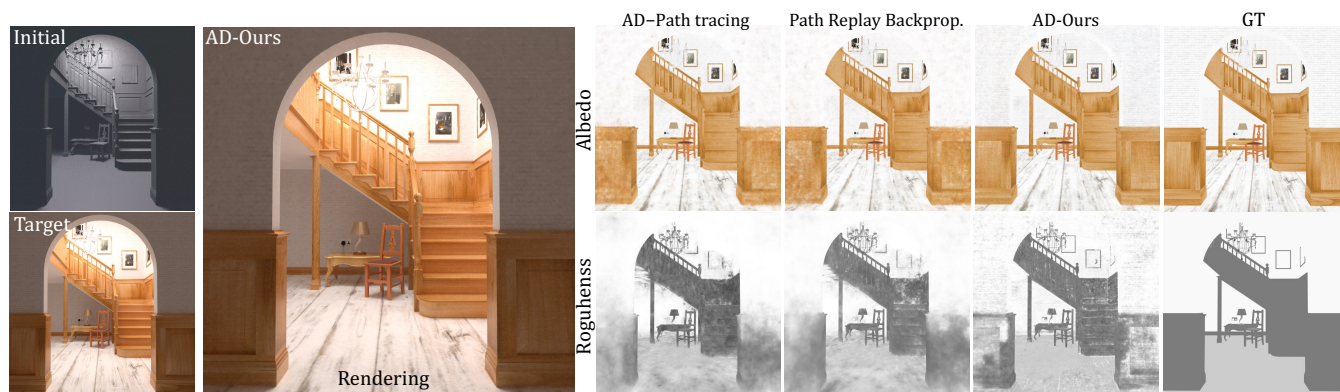
Saeed Hadadan
University of Maryland, College Park
NVIDIA
MD, USA
saeedhd@umd.edu

Geng Lin
University of Maryland, College Park
MD, USA
geng@umd.edu

Jan Novák
NVIDIA
Czech Republic
jnovak@nvidia.com

Fabrice Rousselle
NVIDIA
Switzerland
frousselle@nvidia.com

Matthias Zwicker
University of Maryland, College Park
MD, USA
zwicker@cs.umd.edu

Figure 1: We propose an inverse rendering method that uses a radiometric prior to account for global illumination as opposed to building and differentiating path integrals. Our method uses standard automatic differentiation (AD) to compute gradients with respect to the scene parameters, while satisfying the rendering equation using our radiometric prior, which is represented by a neural network. Here we compare a traditional auto-differentiable path tracer (AD-PT), an advanced technique (Path Replay Backpropagation, or PRB), and our method (AD-Ours) for recovering non-diffuse spatially varying BRDF properties (also represented as neural networks) under known illumination and geometry from 26 views of the STAIRCASE scene. Despite its simplicity, our approach takes into account global illumination, and recovers albedo and roughness with similar accuracy as differentiable path tracing and PRB. Each method used a total of $16384 \times 16 \times 18000$ (batch size $\times$ spp $\times$ steps) = 4.7B samples, i.e., 690 training samples per pixel (26 views $\times$ 512 $\times$ 512 pixels). We conducted all experiments with a single RTX3090 GPU, and the total runtimes for AD-PT, PRB, and our method were 760, 970, and 260 minutes, respectively.

ABSTRACT

Inverse rendering methods that account for global illumination are becoming more popular, but current methods require evaluating and automatically differentiating millions of path integrals by tracing multiple light bounces, which remains expensive and prone to noise. Instead, this paper proposes a radiometric prior as a simple alternative to building complete path integrals in a traditional differentiable path tracer, while still correctly accounting for global illumination. Inspired by the Neural Radiosity technique, we use a neural network as a radiance function, and we introduce a prior consisting of the norm of the residual of the rendering equation in the inverse rendering loss. We train our radiance network and optimize scene parameters simultaneously using a loss consisting of both a photometric term between renderings and the multi-view input images, and our radiometric prior (the residual term). This residual term enforces a physical constraint on the optimization that ensures that the radiance field accounts for global illumination. We compare our method to a vanilla differentiable path tracer, and more advanced techniques such as Path Replay Backpropagation. Despite the simplicity of our approach, we can recover scene parameters with comparable and in some cases better quality, at considerably lower computation times.

CCS CONCEPTS

• **Computing methodologies** → **Ray tracing**; *Neural networks*.

KEYWORDS

Photo-realistic rendering, Ray Tracing, Global Illumination, Differentiable Rendering, Neural Rendering, Neural Radiance Fields

1 INTRODUCTION

Inverse rendering—the task of recovering the parameters of a 3D scene from 2D images—has seen rapid progress and adoption in recent years. Two key components of an inverse rendering approach are a *differentiable renderer* and an algorithm for propagating gradients of the objective function to scene parameters. Differentiable rendering is a challenging problem, especially if global illumination effects are to be taken into account, since this process typically involves Monte Carlo path tracing and calculating derivatives of millions of path integrals with respect to scene parameters.

In principle, path integrals can be differentiated using standard *automatic differentiation* (AD), a universal approach that constructs a graph of operations, which is then traversed in a second pass to compute gradients of the objective function with respect to scene parameters. However, AD suffers from high memory requirements when differentiating multi-bounce path integrals in complex scenes, which limits its use to simple light transport models, such as direct illumination or one-bounce indirect illumination. Yet such simplified light transport models cannot explain appearance changes caused by multiple interreflections, and as a result, the ignored interreflections will be "baked" into the recovered scene parameters.

To reduce the memory requirements of AD, a group of algorithms called *radiative back-propagation* (RB) develop a problem-specific automatic differentiation approach that removes the need to store the enormous graph of operations in standard AD. Thus RB is more efficient in its use of memory, but relies on sophisticated light transport algorithms that require familiarity with transport theory and have not yet been fully commoditized. The main benefit of RB, thanks to its light transport foundation, is its ability to accurately reconstruct the scene under a global illumination rendering model.

Instead, our main contribution is a simple approach for differentiable rendering under a global illumination model that does not require computing multi-bounce light path integrals. We leverage a neural network both to represent the radiance function, and to account for global illumination without building path integrals. We achieve this via a radiometric prior, inspired by Neural Radiosity [Hadadan et al. 2021], that represents the norm of the residual of the rendering equation [Kajiya 1986]. To recover scene parameters, we simultaneously optimize our radiance network and the scene parameters, using a loss that takes into account both the radiometric prior and the difference between the multi-view ground truth images and rendered images. Because this process does not require computing multi-bounce light paths, we can efficiently differentiate it using standard automatic differentiation to compute gradients.

The radiometric prior achieves the following benefits: (1) it ensures that the radiance field represented by our radiance network satisfies the rendering equation and accurately accounts for global illumination, and the optimization of scene parameters is guided towards a physically valid solution; (2) it enables automatic differentiation to yield accurate results without having to simulate many bounces of light using path integrals, which is key to curbing the memory requirements of AD and also results in significant performance gains; (3) it allows us to achieve accurate reconstruction quality similar as radiative back-propagation, but with the simple and straightforward approach of standard automatic differentiation. Please note we will use the terms radiometric prior, radiometric term, and residual term interchangeably throughout the paper.

Scope. Our work presents a contribution that lifts some of the constraints of inverse rendering under a global illumination model with standard automatic differentiation. We analyze the method on synthetic scenes, reconstructing either the material parameters, or lighting; geometry reconstruction and joint optimization are outside of our scope. We use a neural radiance field instead of Monte Carlo estimators of (differential) radiance. Our gradients are thus biased, yet the optimization reaches comparable reconstruction accuracy, often faster than prior works. Explicitly handling visibility-related discontinuities are outside of our scope, but we believe that the smoothing property of our radiance network can reduce the negative effects of discontinuities in the integrands of the rendering equation.

2 RELATED WORK

For brevity, this section focuses on differentiable rendering techniques, radiance field representations, and neural inverse rendering techniques. We refer to existing resources for an overview of classical [Pharr et al. 2016] and neural [Tewari et al. 2020, 2022] rendering.

Differentiable Rendering. Many techniques in visual computing rely on differentiable rendering to extract scene properties from images. Techniques building on differentiable rasterizers [Kato et al. 2018; Laine et al. 2020; Liu et al. 2019; Loper and Black 2014; Petersen et al. 2019; Rhodin et al. 2016] are fast, but do not produce accurate results as effects due to indirect illumination tend to be incorrectly attributed to scene parameters. Naively differentiating advanced rendering algorithms, such as path tracing, via automatic differentiation is severely limited by the memory requirements of AD. *Radiative Backpropagation* (RB) [Nimier-David et al. 2020] sidesteps this issue via an adjoint approach that propagates gradients of the objective function to scene parameters using Monte Carlo path sampling. The quadratic cost of the algorithm has been addressed by Vicini et al. [2021] (PRB), who propose to "replay" paths to save cost. Both of these methods yield accurate scene reconstructions, but their implementation is intricate and sampling strategies for building paths optimally are not yet fully developed.

Conversely, our method accounts for global illumination by enforcing a radiometric prior instead of building and differentiating complete, multi-bounce path integrals, and hence we can use standard AD to compute gradients. Automatic differentiation and PRB

are among the baselines we compare our method to and there is more discussion about it throughout this paper.

Recently, a neural-network based representation of the *differential radiance* was proposed by Hadadan et al. [2022], which is also inspired by Neural Radiosity. However, the two methods are very different. The prior work requires an adjoint estimator and caches *differential* radiance. As a result, it needs one network output per scene parameter and scales poorly to practical scenarios. In contrast, our method caches radiance, and can optimize large parameter sets as we only require AD.

Radiance field representations. The computation of radiometric quantities is a significant burden in rendering, which also applies to the computation of our radiometric prior. Various caching methods have been proposed to tackle this problem, starting with the seminal work of Ward et al. [1988] on diffuse interreflection, later extended to volumes [Greger et al. 1998] and glossy surfaces [Krivanek et al. 2005]. A common challenge for these techniques is the design of cache data structures, as these must accommodate well to any scene geometry and support view-dependent queries. We also utilize a caching mechanism, but we sidestep the need for a sophisticated data structure by representing the radiance cache with a neural network as proposed in previous works [Mildenhall et al. 2020; Müller et al. 2020, 2021; Ren et al. 2013]. Specifically, our radiometric prior builds upon the Neural Radiosity model introduced by Hadadan et al. [2021].

Neural inverse rendering. NeRF, introduced by Mildenhall et al. [2020], proposes to represent a scene as a neural radiance and density field encoded with an MLP. While this approach achieves state-of-the-art accuracy in novel view synthesis of complex scenes, it effectively bakes the scene parameters, such as reflectance and lighting, into the neural representation. The NeRV method of Srinivasan et al. [2020] proposes to separately recover the reflectance of the scene elements, while accounting for indirect illumination, to enable relighting applications. Their technique, however, explicitly differentiates the path integral and is therefore limited in practice to one-bounce indirect illumination to fit a reasonable memory and computational budget. In contrast, our radiometric prior captures the steady-state of light transport, thereby modelling the full scope of indirect illumination at a fixed memory and computational overhead. Another work, NerFactor [Zhang et al. 2021] extracts a geometric representation from the NeRF density field and proceeds to recover the albedo, visibility and surface normals in the scene, albeit under a simplified direct illumination model. More recently, Zhang et al. [2022] proposed to estimate the global illumination from a radiance field obtained with an existing off-the-shelf technique. The accuracy of their approach, however, is inherently bounded by the accuracy of the underlying neural radiance field, whereas our approach explicitly applies physically-based constraints to the optimization of scene parameters.

Visibility-related Discontinuities. Differentiating the rendering equation without taking into account discontinuities may result in incorrect gradients, and special treatment is necessary for discontinuities caused by occlusions [Bangaru et al. 2021; Li et al. 2018; Loubet et al. 2019; Zhang et al. 2020]. Li et al [2018] have addressed this issue by separating the rendering equation into continuous and discontinuous parts, and proposing a silhouette edge sampling approach. Loubet et al [2019] have suggested a reparametrization of the rendering integrals so that the positions of discontinuities are independent of the scene parameters. Although this paper does not explicitly address discontinuities, and therefore does not study the issue, the authors suggest that the smoothing property of a radiance neural network may help mitigate the problems arising from not handling discontinuities in the pixel and light integrals.

3 BACKGROUND

For differentiable rendering under a global illumination model, we build on the rendering equation [Kajiya 1986], which defines the radiometric equilibrium between the outgoing, emitted, and incident radiance that is scattered at each point in the scene,

$$L(x, \omega_o) = E(x, \omega_o) + \int_{\mathcal{H}^2} f(x, \omega_i, \omega_o) L(r(x, \omega_i), -\omega_i) d\omega_i^\perp, \quad (1)$$

where x is a surface point, ω_i and ω_o are directions of incidence and exitance at x, $L(x, \omega_o)$ is the outgoing radiance as a function over surface locations and outgoing directions, $E(x, \omega_o)$ is emitted radiance, $f(x, \omega_i, \omega_o)$ is the bidirectional reflectance distribution function (BRDF), $r(x, \omega_i)$ is the ray tracing operator returning the closest surface intersection of ray (x, ω_i), $\mathcal{H}^2$ is the upper hemisphere, and $d\omega_i^\perp$ is the differential projected solid-angle measure.

The rendering equation is written more concisely in operator notation as $L(x, \omega_o) = E(x, \omega_o) + \mathcal{T}(L)(x, \omega_o)$, where the transport operator $\mathcal{T}$ represents the hemispherical integral in Equation (1). Its solution is given by the Liouville–Neumann series $L(x, \omega_o) = \sum_{i=0}^{\infty} \mathcal{T}^i(E)(x, \omega_o)$, where each term $\mathcal{T}^i(E)(x, \omega_o)$ is a $2i$-dimensional integral over multi-bounce light paths with i segments.

Denoting an image consisting of a set of pixels as $I = \{I_k\}$, each pixel I_k is given by the so-called measurement integral,

$$I_k = \int_{\mathcal{A}} \int_{\mathcal{H}^2} W_k(x, \omega) L(r(x, \omega), -\omega) dx d\omega^\perp, \quad (2)$$

where $W_k(x, \omega)$ models the response of a sensor pixel to incident radiance over its area $\mathcal{A}$ and the hemisphere of directions $\mathcal{H}^2$. In practice, the integrals in Equation (2) are most commonly estimated using Monte Carlo sampling and path tracing.

3.1 Neural Radiosity

Instead of building on the series expansion and path integrals, Neural Radiosity [Hadadan et al. 2021] finds a solution of the rendering equation using a neural network, without requiring multi-bounce light path integrals. The radiance function $L(x, \omega_o)$ in Equation (1) is represented by a neural network with a set of parameters θ, denoted as $L_\theta(x, \omega_o)$, that takes location x and direction ω_o and returns the outgoing radiance. Neural Radiosity then determines the network parameters θ by minimizing the norm of the *residual* of the rendering equation, which is

$$r_\theta(x, \omega_o) = L_\theta(x, \omega_o) - E(x, \omega_o)$$
$$- \int_{\mathcal{H}^2} f(x, \omega_i, \omega_o) L_\theta(x'(x, \omega_i), -\omega_i) d\omega_i^\perp, \quad (3)$$

that is, the difference between the left and right-hand sides of Equation (1) when the radiance function L is substituted by L_θ.

Monte Carlo sampling is used to estimate the norm of the residual as in

$$\mathcal{L}(\theta) \approx \frac{1}{N} \sum_{j=1}^{N} \frac{r_\theta(x_j, \omega_{o,j})^2}{p(x_j, \omega_{o,j})}, \quad (4)$$

where N is the number of samples, and samples of surface locations x_j and outgoing directions $\omega_{o,j}$ are distributed according to probability density $p(x, \omega)$.

Neural Radiosity is a self-training approach, that is, there is no use of ground truth data to supervise the training. The resulting radiance network accounts for global illumination effects without integrating over the path space. Our method builds on Neural Radiosity as we will elaborate in Section 4.

3.2 Inverse Rendering under Global Illumination

Inverse rendering is the problem of finding unknown scene parameters ϕ in 3D such that, under a given rendering model, they match with a set of given 2D images. We denote an image rendered using certain scene parameters ϕ as $I(\phi)$, where ϕ may represent surface geometry, appearance, and lighting.

The objective of inverse rendering is to recover optimal parameters ϕ^* by minimizing a photometric loss function $\mathcal{L}(I(\phi))$,

$$\phi^* = \arg\min_\phi \mathcal{L}(I(\phi)), \quad (5)$$

where $\mathcal{L}(I(\phi))$ quantifies the similarity of a rendered image $I(\phi)$ and a given input image I^{GT}. For simplicity our notation assumes a single input image, and the photometric loss can be defined for example as the L_2 distance $\mathcal{L}(I(\phi)) = \|I(\phi) - I^{\text{GT}}\|$, but in practice images from multiple viewpoints are typically involved.

In a practical approach, we minimize the loss $\mathcal{L}(I(\phi))$ using a gradient-based optimization algorithm, and in principle, the gradient with respect to scene parameters $\partial \mathcal{L}(I(\phi))/\partial \phi$ can be computed using standard automatic differentiation. Because rendering $I(\phi)$ using Equation (2) requires computing millions of pixels and path integrals, however, standard AD creates large computation graphs that easily exceed available memory.

Adjoint Methods. To reduce memory requirements of standard AD, a group of algorithms called *radiative backpropagation* [Nimier-David et al. 2020] use the chain rule to separate the computation of an adjoint image, consisting of the derivatives of the loss with respect to image pixels, and the differentiation of image pixels with respect to scene parameters,

$$\frac{\partial \mathcal{L}}{\partial \phi} = \frac{\partial \mathcal{L}}{\partial I} \cdot \frac{\partial I}{\partial \phi}. \quad (6)$$

These methods require differentiated versions of Equations (1) and (2) with respect to the scene parameters ϕ,

$$\partial_\phi I_k = \int_{\mathcal{A}} \int_{\mathcal{H}^2} W_k(x, \omega) \partial_\phi L(x, \omega) dx d\omega^\perp \quad (7)$$

and

$$\partial_\phi L(x, \omega_o) = \partial_\phi E(x, \omega_o) + \int_{\mathcal{H}^2} f(x, \omega_i, \omega_o) \partial_\phi L(x'(x, \omega_i), -\omega_i) d\omega_i^\perp$$
$$+ \int_{\mathcal{H}^2} \partial_\phi f(x, \omega_i, \omega_o) L(x'(x, \omega_i), -\omega_i) d\omega_i^\perp, \quad (8)$$

where ∂_ϕ is a short form for ∂/∂_ϕ. Nimier David [2020] call the latter the *differential rendering equation*.

The differential rendering equation can be solved using a series expansion similar as the original rendering equation, leading to an expression for $\partial_\phi L$ given by the path integrals

$$\partial_\phi L = \sum_{i=0}^{\infty} \mathcal{T}^i(\partial_\phi E) + \sum_{i=0}^{\infty} \mathcal{T}^i(\mathcal{T}'(\sum_{k=0}^{\infty} \mathcal{T}^k(E))), \quad (9)$$

where $\mathcal{T}'$ is the operator notation for the second integral in Equation (8). Path replay backpropagation (PRB) [Vicini et al. 2021] samples and evaluates these integrals efficiently by reusing computations in a clever way. In particular, one first evaluates a so-called primal path to compute samples of the term $\sum_k \mathcal{T}^k(E)$ (one sample for each path length k, paths are terminated using Russian roulette as usual), which is the same as the series expansion of the usual rendering equation. Then, one reuses this same path together with various terms evaluated and stored along the path, to evaluate samples of the other integrals in Equation (9).

Even with advanced techniques such as PRB, the main issue remains to be the need to build long path integrals to correctly and efficiently evaluate the integrands in Equation (9) by reusing computations. In addition, variance in the Monte Carlo estimates will lead to noisy gradients that may lead to slow convergence.
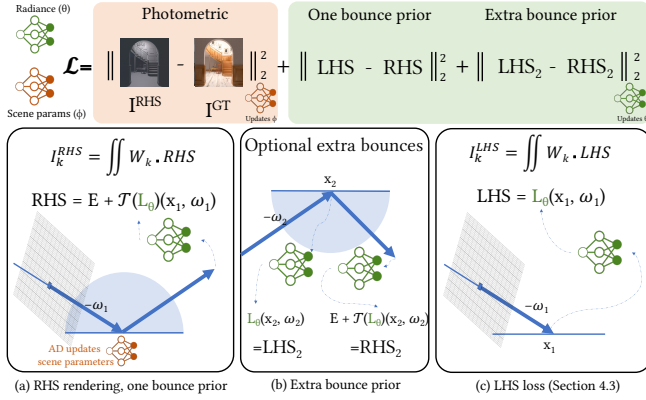
4 METHOD

We avoid the drawbacks above by completely circumventing the estimation of path integrals to solve the (differential) rendering equation. Instead, our key idea is that we use the rendering equation as a radiometric prior, which we will add as an additional loss term to the inverse rendering optimization objective in Equation (5).

To formulate the radiometric prior, we represent the radiance field in the scene as a parametric function $L_\theta(x, \omega_o)$ implemented as a neural network, where θ represents the network weights. Similarly to Hadadan et al. [2021], the radiometric prior $\mathcal{L}_{\text{prior}}$ is given by the norm of the residual of the rendering equation,

$$\mathcal{L}_{\text{prior}}(\theta) = \|L_\theta(x, \omega_o) - (E(x, \omega_o) + \mathcal{T}(L_\theta)(x, \omega_o))\|. \quad (10)$$

The residual is simply the difference between the left-hand side (LHS) and right-hand side (RHS) of Equation (1). Note that the prior depends both on the radiance field and scene parameters, θ and ϕ respectively, although we omit the dependence on ϕ in our notation for simplicity. For known scene parameters, Hadadan et al. [2021] have shown that minimizing this term on its own can be used to solve the rendering equation. In contrast, we leverage this term for inverse rendering to recover scene parameters that are unknown.

In our inverse rendering approach, we simultaneously minimize the photometric loss $\mathcal{L}(I(\phi))$ to ensure the rendering $I(\phi)$ matches the input image, while enforcing the radiometric prior $\mathcal{L}_{\text{prior}}(\theta)$ to

(a) RHS rendering, one bounce prior (b) Extra bounce prior (c) LHS loss (Section 4.3)

Figure 2: Method overview. (a) We use RHS rendering in the photometric loss (top-left) whose gradients are back-propagated using AD to the scene parameters at the primary bounce. The radiometric prior (top right) acts as a regularizer to ensure the solution satisfies the rendering equation, thereby accounting for global illumination; its gradients are back-propagated to the radiance network. (b) The prior can model global illumination using a single bounce, but it is not restricted to it. Additional radiometric terms can be defined and optimized at later bounces to impose radiometric constraints on locations not directly observed by the camera (Section 4.2). (c) Computing a photometric loss based on LHS can further improve results (Section 4.3).

ensure the radiance field satisfies the rendering equation,

$$\phi^*, \theta^* = \arg\min_{\phi,\theta} \mathcal{L}(I(\phi)) + \mathcal{L}_{\text{prior}}(\theta). \qquad (11)$$

Crucially, having the radiance field $L_\theta(x, \omega_o)$, which is constrained by the radiometric prior, allows us to formulate a simple image formation model $I(\phi) = \{I_k\}$ that correctly captures global illumination. Instead of using Equation 2, which requires estimating path integrals to calculate the radiance L, we compute pixels using the neural radiance field $L_\theta(x, \omega_o)$ as

$$I_k = \int_{\mathcal{A}} \int_{\mathcal{H}^2} W_k (E + \mathcal{T}(L_\theta)(x, \omega)) dx d\omega^\perp =: I_k^{\text{RHS}}. \qquad (12)$$

The rendering $I(\phi)$ and its pixels I_k depend on scene and radiance field parameters, ϕ and θ respectively, but we keep our notation as before for simplicity. We also call this *RHS rendering* (Figure 2), because it involves evaluating the right-hand side (RHS) of Equation 1.

As a key advantage, the RHS rendering model does not involve any path integrals or path tracing. As a consequence, we can easily compute the derivatives $\partial \mathcal{L}(I(\phi))/\partial\phi$ with standard AD and we do not need to use the adjoint approach.

4.1 Gradient Computation

As mentioned above, the radiometric prior both depends on the radiance field and scene parameters, θ and ϕ. However, the prior needs to be satisfied for any choice of scene parameters; that is, irrespective of the scene parameters, the radiance field needs to

satisfy the rendering equation. Hence we need to treat the scene parameters as given constants when minimizing the prior. We also observed experimentally that if the radiometric prior influences the scene properties ϕ, this detrimentally affects our results (Figures 7 and 8). In practice, we block backpropagating gradients from the radiometric prior term to the scene parameters, and we update scene parameters only using gradients of the photometric loss $\mathcal{L}(I(\phi))$. Similarly, the photometric loss depends both on θ and ϕ. However, we obtain better results in practice by not backpropagating gradients of $\mathcal{L}(I(\phi))$ to radiance field parameters θ.

4.2 Sampling

We numerically evaluate the integrals in Equation (10) and (12) using Monte Carlo sampling. We sample the pixel measurement and transport operator in Equation (12) using standard ray tracing with a given number of samples per pixel. We trace two consecutive ray segments, one from the camera to a primary hit point, and a second one from the primary hit point to a secondary hit point, and we use the two ray segments to evaluate the transport operator.

In theory, the radiometric prior in Equation (10) should be estimated by sampling over all surface locations and pairs of incident and outgoing directions (to evaluate the transport operator). For simplicity, however, we reuse the primary hit point and the two ray segments obtained from sampling the measurement equation to also sample the radiometric prior. We call this approach the *one bounce prior*. While this simple approach fails to minimize the radiometric prior at locations not directly visible from the camera, it often works well in practice and faithfully captures global illumination effects as our experiments show. To improve results at a small additional cost, we can also ray trace a third consecutive ray segment and sample the radiometric prior both at the first and secondary hit points, which leads to better coverage of surface locations. We call this extended approach the *extra bounce prior*.

4.3 Ground Truth Radiance Field Constraint

So far, the radiance network L_θ is only being optimized using the radiometric prior, which uses no ground truth data–implementing a *self-training* approach similar as in Neural Radiosity. In inverse rendering problems, however, we may have high dynamic range ground truth images I^{GT} that can also be used to constrain the radiance network. To leverage I^{GT} to constrain L_θ, we can directly render the radiance field into an image $I^{\text{LHS}}(\theta)$ consisting of pixels

$$I_k^{\text{LHS}} = \int_{\mathcal{A}} \int_{\mathcal{H}^2} W_k L_\theta(x, \omega) dx d\omega^\perp, \qquad (13)$$

and define and additional loss term

$$\mathcal{L}_{\text{LHS}}(\theta) = \left\| I^{LHS}(\theta) - I^{\text{GT}} \right\|^2. \qquad (14)$$

Our experiments show that adding this loss to our objective in Equation (11) produces small additional improvements (Figures 7 and 8), hence we include this term in our final results.

5 IMPLEMENTATION

5.1 Parameterizing Scene Parameters

We use the Burley BRDF [Burley 2012] in a spatially varying configuration and parameterize its albedo and roughness parameters

using separate MLPs that receive a surface location as input. When optimizing for lighting (Figure 8), we store it as a learned environment map texture.

5.2 Use of Automatic Differentiation

Having all terms of the objective computed, we rely on AD to propagate gradients of the objective to the albedo and roughness MLPs, and the gradients of the radiometric term to the radiance MLP. Technically, computing unbiased gradients using AD requires two forward passes to avoid correlation between the derivatives of the loss and the derivatives of the measurement equation. We take the same approach in this paper when using AD with our method, as well as path tracing and direct illumination. Additionally, we used detached sampling [Zeltner et al. 2021] across all AD based methods.

5.3 System & Architecture

We implement our differentiable renderer using Mitsuba 3 [Jakob et al. 2022], which provides primitives for differentiable rendering using automatic differentiation and radiative backpropagation. We implement our MLPs in PyTorch [Paszke et al. 2019] and embed them inside the Mitsuba renderer, while allowing the gradients to flow between the two frameworks. For PRB we use the official Python implementation. We use an MLP with 3 hidden layers, 256 neurons in each, to represent the radiance field. The roughness and albedo are each modeled using a dedicated MLP with a single hidden layer and 256 neurons. We equip all MLPs with a hash grid [Müller et al. 2022] to facilitate high spatial resolutions; we use the following configuration: resolutions = $[2^1 - 2^{16}]$, features per level = 2, hash table size = 2^{17}. The radiance network receives location, direction, surface normal, and albedo as input. Our code and data are publicly available online[1].

5.4 Training

We use the Adam optimizer with learning rate $5 \cdot 10^{-4}$ in all experiments. At each training step, we select a random view from the scene, and randomly crop to a certain size from the image. For NeRF scenes with transparent background, at least half of the samples in every training batch are foreground pixels. The patch is rendered to compute the loss and gradients for the parameter updates. Detailed statistics for all scenes can be found in Table 1. We use the relative L_2 loss function for the photometric and radiometric terms. The albedo-, roughness-, and radiance-predicting MLPs are trained jointly, starting with randomly initialized weights.
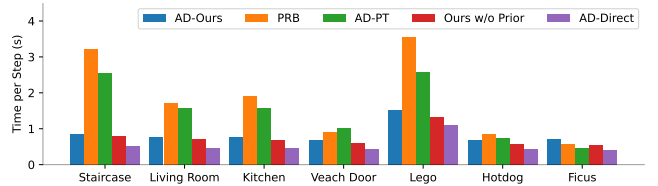
6 RESULTS & ANALYSIS

In this section, we discuss our experimental setup and compare our proposed automatic differentiation using a radiometric prior (AD-Ours) to three baselines: automatic differentiation of direct illumination (AD-Direct), path replay backpropagation (PRB) [Vicini et al. 2021], and an approach, labeled as AD-Ours w/o prior, where we only train the radiance field using the ground truth images (Section 4.3), without including the radiometric prior. This is similar in spirit to the idea proposed by Zhang et al. [2022].

[1]https://inverse-neural-radiosity.github.io

Table 1: Training statistics per scene.

	Staircase	Kitchen	Living Room	Veach Door	Lego	Hotdog	Ficus	Dragon
Mean path len.	4.74	3.67	4.10	3.33	2.32	1.46	1.43	1.26
# of views	26	62	64	19	25	25	25	25
Training spp	16	64	64	64	64	64	64	64
Batch size 2^x	14	12	12	12	12	12	12	12
Total steps	18K	32K	32K	32K	10K	10K	10K	10K

Figure 3: Total training time per step.

Table 2: *Cube* scene path lengths. We report the average and maximum path length for path tracing in the *Cube* scene depending on surface albedo.

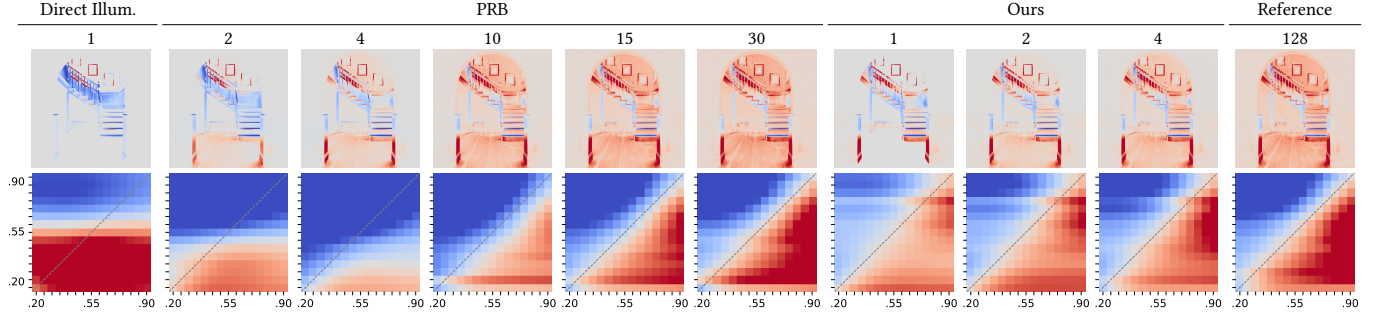| Albedo	0.3	0.5	0.7	0.9	0.95	0.97
Average path length	1.43	1.98	3.30	9.89	19.76	32.89
Limit	5	7	13	42	84	140

Multi-view datasets. We evaluated our method on a set of synthetic scenes, including indoor scenes [Bitterli 2016] and scenes from the NeRF dataset [Mildenhall et al. 2020]. All scenes were altered to use the Burley BRDF. In the STAIRCASE scene we increased the albedo to evaluate the methods in the presence of strong indirect illumination. For each indoor scene, we rendered a multi-view dataset using the Mitsuba 3 path tracer [Jakob et al. 2022], placing cameras manually to cover the entire scene. For the NeRF scenes, we sampled cameras on the hemisphere surrounding the object.

Experimental setup. All results for all methods use the same optimization hyper-parameters. The path length for PRB (and AD-PT) is limited to 15 bounces and paths are terminated using Russian Roulette with probability=0.95 [Arvo and Kirk 1990]; we report the mean path length and the number of views used for optimization for each scene in Table 1.
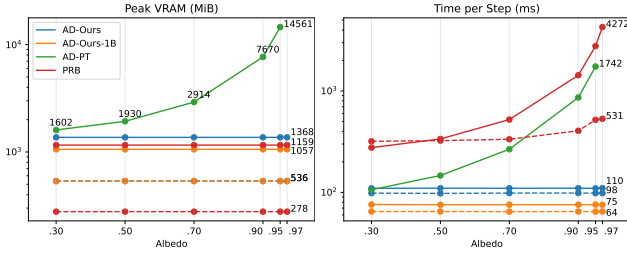
6.1 Comparison to previous techniques

We present results across all methods when reconstructing the albedo and roughness of the indoor and NeRF scenes in Figure 6 and supplementary. In most cases, our method faithfully recovers the parameters thanks to its use of a radiometric prior that correctly accounts for global illumination. Despite its simplicity, our method achieves comparable (and sometimes better) results to PRB. Compared to the premise of Zhang et. al [2022], which uses only ground truth images to train the radiance network, our method converges to a better solution. This is because we optimize the radiance network to respect the radiometric prior in additional spatio-directional locations, rather than just fitting it to the pixels in the input images.

Figure 4: Analysis of biased gradients. We visualize gradients with respect to the roughness parameter of the wooden material on the staircase and the picture frames, using different methods. We cap the maximum path length for PRB at each specified number. For our approach, we perform a certain number k of *differentiable* bounces before querying our radiance cache. The first row visualizes per-pixel gradient magnitudes with respect to roughness ($\frac{\partial I}{\partial \phi}$ in Equation (6)) at a value of 0.5. We can see that the roughness of the staircase leads to indirect gradients on pixels on the floor. These indirect gradients are missed by our method using only $k = 1$ differentiable bounce, but accounted for when tracing more differentiable bounces ($k = 2, 4$). Note that at equal number of bounces, our method provides more accurate global information than PRB because our radiance field accurately captures global illumination, irrespective of the number of differentiable bounces k. The second row visualizes the magnitude of the gradient of the L2 loss with respect to the roughness parameters over the entire image, comparing the current state with the target state ($\frac{\partial \mathcal{L}}{\partial \phi}$ in Equation (6)). For unbiased gradients, cells on the diagonal have gradients equal to zero, and non-zero values indicate bias. As shown, direct illumination, PRB with low numbers of bounces, and our method provide biased gradients. Increasing the number k of differentiable bounces reduces bias in our gradients. All results in this paper use 'Ours' with $k = 1$ differentiable bounce, but we sample the residual at an extra bounce. In all visualizations, red indicates negative and blue indicate positive values.

Figure 5: *Cube* scene measurements. We compare the time and memory consumption of each training step, with all parameters represented in dense grids. Dotted lines have Mitsuba 3 mega kernels enabled. AD-PT runs out of VRAM (24GB) at albedo 0.97. Our method uses a constant amount of VRAM and time.

6.2 Time and memory consumption

A key benefit of our approach is that the memory and time usage of each training step is independent of the light path lengths sufficient to represent global illumination, therefore our method results in significant speedups (see Figure 3) and bounds memory usage. We illustrate this on the *Cube* scene where the cameras are placed at the corners inside a closed cube, the walls have a uniform, diffuse material, and the only light source is an area light on the ceiling. By varying the albedo of the walls, we can control the average path length in a path tracer as shown in Table 2; implementation details are provided in the supplementary document. In Fig. 5 we report how time and memory consumption of all methods change as the albedo increases from 0.3 to 0.97. To avoid overhead between Mitsuba and PyTorch and to be able to use a mega-kernel, we

conduct measurements using Mitsuba dense grids representing the scene parameters and radiance (results obtained with MLPs can be found in the supplementary document). We observe that both our method and PRB use a constant amount of memory, while memory requirements for AD-PT grow with increasing albedo, i.e., increasing average path length. Our method, however, runs faster than PRB and AD-PT, and in constant time independent of albedo.

6.3 Biased gradients

Unlike path integral methods with sufficiently many bounces, our method provides biased gradients due to two reasons. First, the neural radiance field is only an approximation of the true radiance field, which in itself causes bias. Furthermore, our method introduces bias because we do not differentiate the neural radiance field itself with respect to the scene parameters. More formally, in Equation (8) the term $\partial_\phi L$ is assumed to be zero.

We illustrate the effect of this approximation in Figure 4. Note that we can easily generalize our method by replacing the term $E + \mathcal{T}(L_\theta)$ in Equation 12 with a truncated series expansion $\sum_{i=0}^{k-1} \mathcal{T}^i(E) + \mathcal{T}^k(L_\theta)$. For a small $k > 1$, we can obtain additional differentiable light bounces with standard AD, and increasing k reduces bias in the gradients. Figure 4 shows how setting $k = 2, 4$ reduces bias in our gradients. In practice, we use $k = 1$ as in Equation 12 for all results in this paper since this is sufficient to obtain good results.

6.4 Ablation Study

We now study how each component of our method contributes to the final result when optimizing material properties (Figure 7) and lighting (Figure 8). We start from a direct illumination integrator, which recovers albedo/roughness and environment maps

with baked-in artifacts that are due to ignoring indirect effects. Next, we add the radiometric prior and prevent its gradient from updating the scene parameters. The next column demonstrates the improvement due to leveraging the radiometric prior also on the second bounce. Finally, adding the LHS reconstruction loss to train the radiance network yields further improved results.

7 LIMITATIONS AND FUTURE WORK

Specular materials. We do not attempt to model outgoing radiance on specular surfaces with the radiance MLP. Instead, we keep tracing the path until it hits a non-specular surface. Our technique thus reverts to differentiating the path integral for specular chains and therefore suffers from the same limitations as AD-PT—large operation graph—if many surfaces in the scene are specular.

Joint reconstruction of parameters. In this paper, we either optimized MLP representations of BRDF parameters or an environment map. Extending our method to joint optimization of materials, geometry, and lighting would be an interesting avenue for future research.

8 CONCLUSION

In this paper, we proposed an inverse rendering method that leverages a radiometric prior—as a physical condition on the optimization—to account for global illumination. We evaluate the prior using a neural radiance field, which is trained during the optimization of scene parameters by minimizing the norm of the residual of the rendering equation. Our method is theoretically grounded yet simple and practical. We achieved the best results using neural representations, but the approach is compatible with standard data structures. In contrast to prior work that requires tracing complete path samples, our approach recovers scene parameters with comparable (and sometimes better) quality. Introducing additional physics constraints on the optimization may improve performance further; we believe this is a direction worth pursuing in the future.

ACKNOWLEDGMENTS

REFERENCES

James Arvo and David Kirk. 1990. Particle Transport and Image Synthesis. *SIGGRAPH Comput. Graph.* 24, 4 (sep 1990), 63–66. https://doi.org/10.1145/97880.97886

Sai Bangaru, Jesse Michel, Kevin Mu, Gilbert Bernstein, Tzu-Mao Li, and Jonathan Ragan-Kelley. 2021. Systematically Differentiating Parametric Discontinuities. *ACM Trans. Graph.* 40, 107 (2021), 107:1–107:17.

Benedikt Bitterli. 2016. Rendering resources. https://benedikt-bitterli.me/resources/.

Brent Burley. 2012. Physically Based Shading at Disney. In *Practical Physically-Based Shading in Film and Game Production*. ACM SIGGRAPH 2012 Courses. https://doi.org/10.1145/2343483.2343493

G. Greger, P. Shirley, P.M. Hubbard, and D.P. Greenberg. 1998. The irradiance volume. *IEEE Computer Graphics and Applications* 18, 2 (1998), 32–43. https://doi.org/10.1109/38.656788

Saeed Hadadan, Shuhong Chen, and Matthias Zwicker. 2021. Neural Radiosity. *ACM Trans. Graph.* 40, 6, Article 236 (dec 2021), 11 pages. https://doi.org/10.1145/3478513.3480569

Saeed Hadadan and Matthias Zwicker. 2022. Differentiable Neural Radiosity. *arXiv preprint arXiv:2201.13190* (2022).

Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. 2022. *Mitsuba 3 renderer.* https://mitsuba-renderer.org.

James T. Kajiya. 1986. The Rendering Equation. *SIGGRAPH Comput. Graph.* 20, 4 (aug 1986), 143–150. https://doi.org/10.1145/15886.15902

Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Neural 3D Mesh Renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

J. Krivanek, P. Gautron, S. Pattanaik, and K. Bouatouch. 2005. Radiance caching for efficient global illumination computation. *IEEE Transactions on Visualization and Computer Graphics* 11, 5 (2005), 550–561. https://doi.org/10.1109/TVCG.2005.83

Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. 2020. Modular Primitives for High-Performance Differentiable Rendering. arXiv:2011.03277 [cs.GR]

Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018. Differentiable Monte Carlo Ray Tracing through Edge Sampling. *ACM Trans. Graph.* 37, 6, Article 222 (Dec. 2018), 11 pages. https://doi.org/10.1145/3272127.3275109

Shichen Liu, Weikai Chen, Tianye Li, and Hao Li. 2019. Soft Rasterizer: Differentiable Rendering for Unsupervised Single-View Mesh Reconstruction. arXiv:1901.05567 [cs.CV]

Matthew M. Loper and Michael J. Black. 2014. OpenDR: An Approximate Differentiable Renderer. In *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 154–169.

Guillaume Loubet, Nicolas Holzschuch, and Wenzel Jakob. 2019. Reparameterizing Discontinuous Integrands for Differentiable Rendering. *ACM Trans. Graph.* 38, 6, Article 228 (nov 2019), 14 pages. https://doi.org/10.1145/3355089.3356510

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

Thomas Müller, Fabrice Rousselle, Alexander Keller, and Jan Novák. 2020. Neural Control Variates. *ACM Trans. Graph.* 39, 6, Article 243 (nov 2020), 19 pages. https://doi.org/10.1145/3414329.3417804

Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. 2021. Real-Time Neural Radiance Caching for Path Tracing. 40, 4 (2021). https://doi.org/10.1145/3450626.3459812

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. arXiv:2201.05989 [cs.CV]

Merlin Nimier-David, Sébastien Speierer, Benoît Ruiz, and Wenzel Jakob. 2020. Radiative Backpropagation: An Adjoint Method for Lightning-Fast Differentiable Rendering. *ACM Trans. Graph.* 39, 4, Article 146 (July 2020), 15 pages. https://doi.org/10.1145/3386569.3392406

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. https://doi.org/10.48550/ARXIV.1912.01703

Felix Petersen, Amit H. Bermano, Oliver Deussen, and Daniel Cohen-Or. 2019. Pix2Vex: Image-to-Geometry Reconstruction using a Smooth Differentiable Renderer. arXiv:1903.11149 [cs.CV]

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. *Physically Based Rendering: From Theory to Implementation (3rd ed.)* (3rd ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 1266 pages.

Peiran Ren, Jiaping Wang, Minmin Gong, Stephen Lin, Xin Tong, and Baining Guo. 2013. Global Illumination with Radiance Regression Functions. *ACM Trans. Graph.* 32, 4, Article 130 (jul 2013), 12 pages. https://doi.org/10.1145/2461912.2462009

Helge Rhodin, Nadia Robertini, Christian Richardt, Hans-Peter Seidel, and Christian Theobalt. 2016. A Versatile Scene Model with Differentiable Visibility Applied to Generative Pose Estimation. *CoRR* abs/1602.03725 (2016). arXiv:1602.03725 http://arxiv.org/abs/1602.03725

Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. 2020. NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis. https://doi.org/10.48550/ARXIV.2012.03927

A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner, R. Pandey, S. Fanello, G. Wetzstein, J.-Y. Zhu, C. Theobalt, M. Agrawala, E. Shechtman, D. B Goldman, and M. Zollhöfer. 2020. State of the Art on Neural Rendering. *Computer Graphics Forum* 39, 2 (2020), 701–727. https://doi.org/10.1111/cgf.14022

A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, W. Yifan, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi, T. Simon, C. Theobalt, M. Nießner, J. T. Barron, G. Wetzstein, M. Zollhöfer, and V. Golyanik. 2022. Advances in Neural Rendering. *Computer Graphics Forum* 41, 2 (2022), 703–735. https://doi.org/10.1111/cgf.14507

Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2021. Path Replay Backpropagation: Differentiating Light Paths Using Constant Memory and Linear Time. *ACM Trans. Graph.* 40, 4, Article 108 (jul 2021), 14 pages. https://doi.org/10.1145/3450626.3459804

Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. 1988. A Ray Tracing Solution for Diffuse Interreflection. *SIGGRAPH Comput. Graph.* 22, 4 (jun 1988), 85–92. https://doi.org/10.1145/378456.378490

Tizian Zeltner, Sébastien Speierer, Iliyan Georgiev, and Wenzel Jakob. 2021. Monte Carlo Estimators for Differential Light Transport. *ACM Trans. Graph.* 40, 4, Article 78 (July 2021), 16 pages. https://doi.org/10.1145/3450626.3459807

Cheng Zhang, Bailey Miller, Kai Yan, Ioannis Gkioulekas, and Shuang Zhao. 2020. Path-Space Differentiable Rendering. *ACM Trans. Graph.* 39, 4 (2020), 143:1–143:19.

Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul Debevec, William T. Freeman, and Jonathan T. Barron. 2021. NeRFactor: Neural Factorization of Shape and Reflectance under an Unknown Illumination. *ACM Trans. Graph.* 40, 6, Article 237 (dec 2021), 18 pages. https://doi.org/10.1145/3478513.3480496

Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. 2022. Modeling Indirect Illumination for Inverse Rendering. https://doi.org/10.48550/ARXIV.2204.06837

Figure 6: Main results. For each scene, we compare the rendering, recovered albedo, and recovered roughness for direct illumination, PRB, and AD-Ours. We also compare to the case where the radiance cache is trained without the prior. We show two different views of each scene, and report PSNR to ground truth (GT). For STAIRCASE, *Ours without prior* diverges at 70% progress and we report the best results. Additional scenes are in supplementary.
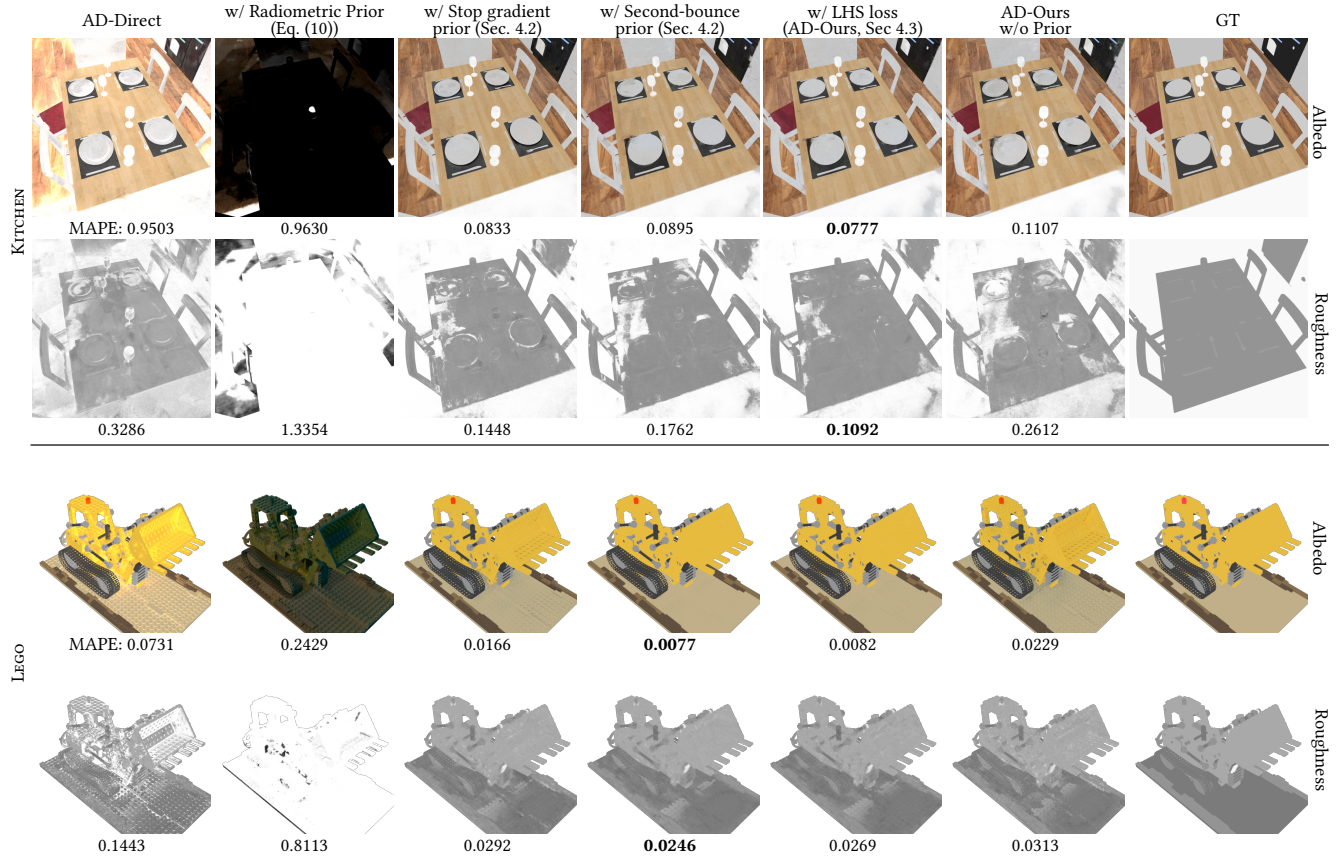
Figure 7: Ablation study of individual components and design decisions. We start with the direct illumination integrator (left), and add the radiometric prior to it. The results significantly improve when we ignore the gradients of the prior w.r.t scene parameters. Adding the prior to the second bounce better accounts for additional global illumination effects for areas unseen by the input cameras. Using ground truth data to improve the radiance field further improves the quality. Finally, the second column from the right shows our full method, except that we omit the prior.
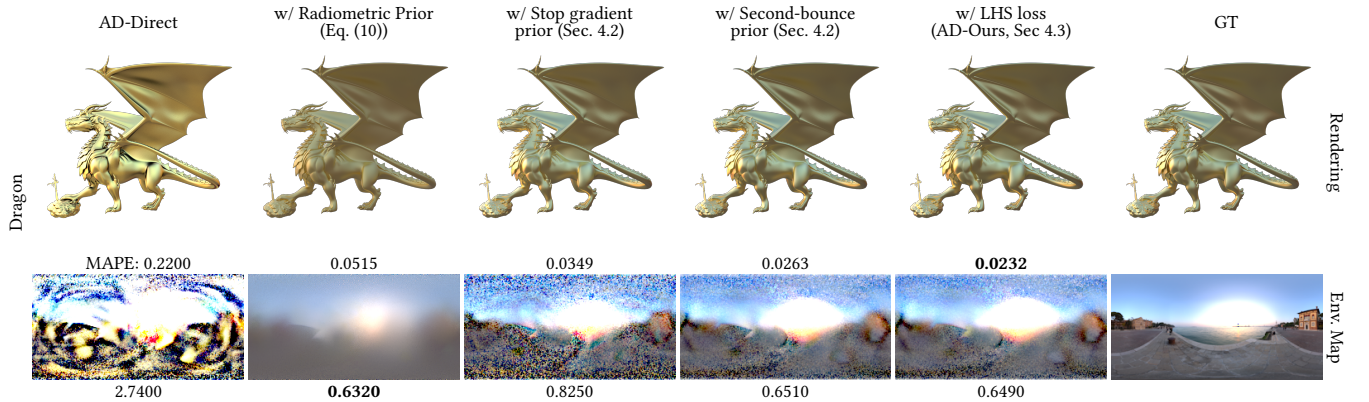

Figure 8: Ablation study for environment map optimization reveals the effectiveness of each component and agrees with the ablation study for material optimization (Figure 7). We start from the direct illumination integrator on the left, add the radiometric prior to it, stop propagating its gradient to the environment map, use the prior on the second bounce, and finally, enable the use of LHS reconstruction loss. Please note how omission of indirect effects in direct illumination causes material colors being baked into the environment map.